МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

> Государственное образовательное учреждение высшего профессионального образования "Оренбургский государственный университет"

О.В.ПРИХОДЬКО, М.А.ТОКАРЕВА

# КОМПЬЮТЕРНЫЙ ПРАКТИКУМ

## УЧЕБНОЕ ПОСОБИЕ ДЛЯ БУДУЩИХ СПЕЦИАЛИСТОВ ПО УПРАВЛЕНИЮ ПЕРСОНАЛОМ

Рекомендовано Ученым советом государственного образовательного учреждения высшего профессионального образования "Оренбургский государственный университет" в качестве учебного пособия для студентов, обучающихся по программам высшего профессионального образования управленческого профиля

Оренбург 2008

#### УДК 004.4:331.108.26(075.8) ББК 32.973-018я73 П77

Рецензент

заведующий кафедрой «Информационные технологии в экономике» ОГАУ, кандидат технических наук, доцент И.В.Матвейкин

О.В.Приходько

П77

Компьютерный практикум. Учебное пособие для будущих специалистов по управлению персоналом. /О.В.Приходько, М.А.Токарева. – Оренбург: ГОУ ОГУ, 2008. – 200 с.

Учебное пособие предназначено для методической поддержки естественно-научных дисциплин регионального вузовского компонента «Компьютерная подготовка» и «Компьютерный практикум» для студентов специальности 080505 – Управление персоналом.

 $\prod \frac{2404010000}{2}$ 

ББК 32.973-018я73

© Приходько О.В., 2008 © Токарева М.А., 2008 © ГОУ ОГУ, 2008

# Содержание

Введение	9
1 Язык разметки гипертекста HTML	.11
1.1 Введение в HTML. Создание простейших Web – документов	.11
1.1.1 Основные понятия	.11
1.1.2 История развития HTML	.12
1.1.3 Теговая модель языка	.15
1.1.4 Структура документа HTML	.16
1.1.5 Функциональные элементы HTML	.17
1.1.6 Форматирование текста	.19
1.1.7 Задания для самостоятельной работы	.20
1.1.8 Вопросы для самоконтроля	.23
1.2 Гипертекстовые ссылки	.23
1.2.1 Понятие гиперссылки	.23
1.2.2 Части URL-адреса документа	.25
1.2.3 Создание якоря на Web-странице	.27
1.2.4 Задания для самостоятельной работы	.29
1.2.5 Вопросы для самоконтроля	.30
1.3 Графика и мультимедиа на Web-страницах	.31
1.3.1 Основной цветовой набор Web-страницы. Варианты описания цвета	.31
1.3.2 Подготовка и размещение графических изображений на Web-страницах.	.32
1.3.3 Озвучивание Web-страницы, внедрение аудиофайлов	.34
1.3.4 Форматы видеофайлов, внедрение видео	.37
1.3.5 Задания для самостоятельной работы	.38
1.3.6 Вопросы для самоконтроля	.40
1.4 Таблицы на Web-страницах	.40
1.4.1 Теги и атрибуты размещения таблиц на Web-страницах	.40
1.4.2 Задания для самостоятельной работы	.45
1.4.3 Вопросы для самоконтроля	.46
1.5 Списки на Web-страницах	.47
1.5.1 Нумерованные списки на Web-страницах	.47
1.5.2Ненумерованные списки на Web-страницах	.48
1.5.3 Списки определений на Web-страницах	.49
1.5.4 Задания для самостоятельной работы	.49
1.5.5 Вопросы для самоконтроля	.51
1.6 Фреймы на Web-страницах	.51
1.6.1 Геги и атрибуты для описания фреймов	.51
1.6.2 Задания для самостоятельной работы	.55
1.6.3 Вопросы для самоконтроля	.56
1.7 Картированные изображения	.57
1.7.1 Гег для создания меню-карты	.57
1.7.2 Описание зон меню-карты, чувствительных к нажатию кнопки мыши	.57
1.7.3 Задания для самостоятельной работы	.60
1.7.4 Вопросы для самоконтроля	10.
1.6 ФОРМЫ И ИХ ООРАООТКа	.02
1.0.1 Тег и атриоуты тега для создания формы	.02
1.8.2 теги описания полеи формы	.04
1.0.3 Задания для самостоятельной работы	.70
г.о.4 вопросы для самоконтроля	.71

1.9 Этапы создания сайта и размещение его в глобальной сети	71
1.9.1 Проектирование сайта	71
1.9.2 Размещение сайта	74
1.9.3 Размещение сайта с помощью протокола ЕТР	83
1.9.4 Вопросы для самоконтроля	86
2 Интегрированная среда разработки Visual Basic for Application	87
2.1 Общие сведения	87
2.2 Интерфейс окна VBA	89
2.2.1 Способы запуска среды VBA	89
2.2.2 Основные элементы окна VBA	90
2.2.3 Вопросы для самоконтроля	97
2.3 Понятие алгоритма	97
2.3.1 Алгоритм. Способы записи алгоритмов	97
2.3.2 Структуры алгоритмов	99
2.3.3 Программный способ записи алгоритмов	105
2.3.4 Вопросы для самоконтроля	107
2.4 Описание алгоритмов на языке VBA	108
2.4.1 Представление данных в памяти	108
2.4.2 Типы данных	109
2.4.3 Идентификаторы, переменные	111
2.4.4 Операции, выражения, операторы, функции	112
2.4.6 Вопросы для самоконтроля	122
2.5 Отладка, использование среды для отладки программ	123
2.5.1 Типы ошибок	123
2.5.2 Обработка ошибок выполнения	126
2.6 Создание макросов при помощи макрорекордера	127
2.7 Реализация алгоритмов линейной структуры средствами VBA	130
2.7.1 Примеры решения задач с линейными алгоритмами	130
2.7.2 Задания для самостоятельной работы	133
2.8 Реализация алгоритмов ветвления средствами VBA	134
2.8.1 Условный оператор If	134
2.8.2 Оператор выбора Select Case	136
2.8.3 Задания для самостоятельной работы	138
2.8.4 Вопросы для самоконтроля	139
2.9 Реализация циклических алгоритмов средствами VBA	139
2.9.1 Циклические операторы	139
2.9.2 Примеры задач, содержащих циклические алгоритмы в решении	141
2.9.3 Задания для самостоятельной работы	144
2.9.4 Вопросы для самоконтроля	145
2.10 Обработка массивов в VBA	145
2.10.1 Описание и работа с массивами	145
2.10.2 Примеры решения задач	148
2.10.3 Задания для самостоятельной работы	153
2.10.4 Вопросы для самоконтроля	154
2.11 Работа со строками	154
2.11.1 Строковые выражения и строковые функции	154
2.11.2 Пример решения задачи, использующей в решении строки	157
2.11.3 Задачи для самостоятельной работы	159
2.12 Управление свойствами объектов VBA	160
2.12.1 Общие сведения	160
2.12.2 Элементы управления панели Toolbox	162

жимости	2.13 Списки в VBA. Создание программного кода для подсчета стоимости н	едви-
2.13.1 Виды списков в VBA	ЖИМОСТИ	164
2.13.2 Создание программного кода для подсчета стоимости недвижимости166 2.13.3 Задания для самостоятельной работы	2.13.1 Виды списков в VBA	164
2.13.3 Задания для самостоятельной работы	2.13.2 Создание программного кода для подсчета стоимости недвижимос	сти166
2.13.4 Вопросы для самоконтроля	2.13.3 Задания для самостоятельной работы	169
2.14 Финансово-математические функции VBA	2.13.4 Вопросы для самоконтроля	171
2.14.1 Пример решения задачи с использованием финансово-математической функции	2.14 Финансово-математические функции VBA	171
функции	2.14.1 Пример решения задачи с использованием финансово-математич	еской
2.14.2 Задания для самостоятельной работы	функции	171
Список использованных источников	2.14.2 Задания для самостоятельной работы	176
Приложение А	Список использованных источников	177
Приложение Б	Приложение А	178
Приложение В	Приложение Б	182
Приложение Г	Приложение В	196
	Приложение Г	200

## Введение

Основная цель профессионального образования – подготовка квалифицированного специалиста соответствующего уровня и профиля, конкурентоспособного на современном рынке труда, компетентного, ответственного, свободно владеющего своей профессией.

В связи с этим актуальна задача активного поиска эффективной системы формирования высокопрофессиональных специалистов, способных к быстрой адаптации на рынке труда, смене специализации в рамках определенного в вузе направления, готовых к непрерывной самообразовательной деятельности.

Понимание информационной картины мира, умение решать профессиональные задачи на основе использования информации, коммуникационных технологий, владение этими технологиями, понимание их применения, слабых и сильных сторон и способов критического суждения в отношении информации формируют ключевые компетенции будущего специалиста, необходимые для любой профессиональной деятельности.

Данное учебное пособие предназначено для студентов специальности «Управление персоналом» как методическая поддержка дисциплин по выбору информационно-компьютерного блока. Информационно-компьютерный блок для этой специальности, имеющий своей целью формирование информационной компетентности будущих управленцев, в ГОУ ОГУ представлен совокупностью нескольких дисциплин: «Информатика», «Компьютерная подготовка» и «Компьютерный практикум». «Информатика» относится к циклу общематематических и естественнонаучных дисциплин федерального компонента. Дисциплины «Компьютерная подготовка» и «Компьютерный практикум» являются дисциплинами по выбору, устанавливаемыми ВУЗом. Данный блок для вышеуказанной специальности охватывает 2 курса обучения.

Мы считаем важным вариативную составляющую дисциплин информационно-компьютерного блока представить профессионально направленной, ориентированной на освоение приемов решения прикладных задач средствами современных информационных технологий. Решение этой важной проблемы мы видим в дифференцированном подходе к проектированию содержания дисциплин, преподаванию этих дисциплин в зависимости от квалификационной характеристики выпускника, при этом полностью реализуя требования государственного образовательного стандарта. Варианты индивидуальных заданий для компьютерного практикума, предлагаемые в этом пособии, ориентированы на предметную область будущей профессиональной деятельности специалистов по управлению персоналом, интегрируют знания сразу по нескольким дидактическим единицам, что позволяет повысить мотивацию студентов к овладению соответствующими средствами современных информационных технологий.

Пособие содержит два раздела, посвященных изучению языка разметки гипертекста HTML и интегрированной среде разработки Visual Basic for Application (VBA).

Проанализировав квалификационную характеристику будущего управленца персоналом, нами выделены умения, которыми он должен обладать при работе с информацией:

1) применение современных средств сбора и переработки информации,

- 2) качественный и оперативный поиск новой информации,
- 3) возможность сделать доступной собственную информацию.

Безусловно, использование глобальной сети Internet для поиска и размещения информации – это наиболее современный и быстрый способ для выполнения таких задач. Для обеспечения доступа к собственной информации используется одна из возможностей сети Internet, а именно: размещение в ней самостоятельно созданных при помощи языка HTML Web-сайтов. Это могут быть, например, сайты предприятий или фирм, предназначенные для демонстрации своей деятельности в целях поиска персонала. Умение создавать и размещать Web-сайты повысит информационную компетентность, следовательно, и конкурентоспособность будущего менеджера по управлению персоналом.

В учебном пособии в доступной форме рассмотрены основные средства и возможности языка программирования высокого уровня VBA, который позволяет создавать профессиональные офисные проекты. Изучение основных принципов технологии объектно-ориентированного программирования ориентирует на дальнейшие возможности быстрого и квалифицированного освоения студентами профессиональных пакетов прикладных программ, позволит конструировать пользовательские функции и формы для создания приложений, повышающих гибкость и расширяющих функциональные возможности офисного приложения MS Excel, с которым менеджер в своей профессиональной деятельности сталкивается наиболее часто.

Разделы пособия разбиты на параграфы, и каждый параграф есть законченный блок по изучению определенной темы. Закрепление темы представлено в виде задач 1-го и 2-го уровней, а также интегрированных задач, при решении которых необходимы умения, знания и навыки, приобретенные при изучении естественнонаучной дисциплины федерального компонента «Информатика». Кроме задач после каждой темы имеются вопросы для самоконтроля, позволяющие оценить уровень усвоения темы и восполнить обнаруженные пробелы.

Список рекомендуемой литературы содержит источники, позволяющие получить более обширные сведения по рассмотренным вопросам и способствующие углубленному изучению данной темы.

Считаем целесообразным применять данное учебное пособие:

- будущим специалистам по управлению персоналом для формирования профессионально значимых навыков будущей профессиональной деятельности;
- преподавателю при подготовке занятий, так как оно содержит структурированный материал, составляющий базовую часть дисциплин по выбору;
- 3) студентам технических и профильных специальностей в качестве дополнительной литературы для изучения возможностей расширения офисных приложений, основанных на парадигме объектно-ориенти-

рованного программирования, и изучении возможностей языка гипертекстовой разметки HTML.

## 1 Язык разметки гипертекста HTML

## 1.1 Введение в HTML. Создание простейших Web – документов

#### 1.1.1 Основные понятия

В настоящее время разработано несколько тысяч языков программирования. Каждый язык имеет свою область использования, предназначен для решения определенных задач.

Название языка программирования HTML происходит от английских слов HyperText Markup Language — язык разметки гипертекста. Этот язык используется при создании красочно оформленных Web-страниц, которые пользователь наблюдает при работе в Internet.

"Всемирная паутина" World Wide Web на сегодня является ведущим сервисом Интернет, постепенно вытесняющим или включающим в себя большинство других сетевых служб. Документы, составляющие содержание WWW, называются Web-cmpaницами, а формат, с помощью которого подготавливаются Web-страницы, называется *HTML* (HyperText Markup Language) или "язык разметки гипертекста". Под *гипертекстом* же в простейшем случае понимается текст, позволяющий не только последовательное прочтение, то есть, указателями-ссылками связанный с другими текстами.

HTML представляет собой достаточно простой набор команд, которые описывают структуру документа. HTML позволяет выделить в тексте отдельные логические части (заголовки, абзацы, списки и т.д.), поместить на Webстраницу отдельно подготовленную фотографию или картинку, организовать на странице ссылки для связи с другими документами.

HTML не задает конкретные и точные атрибуты форматирования документа, как, например, Microsoft Word. Конкретный вид документа окончательно определяет только программа-*браузер* на Вашем компьютере. Необходимость именно такого подхода связана с разнородностью аппаратного и программного обеспечения компьютеров, подключенных к Интернет. HTML также не является языком программирования, но web-страницы могут включать в себя встроенные программы-*скрипты* на языках Javascript и Visual Basic Script и программы-*апплеты* на языке Java.

С точки зрения пользователя Windows, Web-страница – это просто файл типа \*.htm или \*.html, находящийся на сервере Интернет, в локальной сети или на жестком диске Вашей машины.

Способ представления документа - текстовый файл. Причины такого представления заключаются в следующем:

- 1) такой файл можно создать в любом текстовом редакторе на любой аппаратной платформе в среде любой операционной системы
- 2) к моменту разработки HTML существовал американский стандарт для разработки сетевых информационных систем - Z39.50, в котором в качестве единицы хранения указывался простой текстовый файл в кодировке LATIN1, что соответствует US ASCII

Основными понятиями языка HTML являются: гипертекст и гиперссыл-

Гипертекст — это текст, представленный в виде связанных блоков текста. При этом блоки текста могут находиться на просматриваемой Web-странице, на другой странице данного сервера, либо на другом сервере глобальной сети. Переход от блока к блоку осуществляется скачком с помощью гиперссылок.

Гиперссылка — специальным образом выделенный фрагмент гипертекста (или графический объект), который содержит невидимый для пользователя указатель (адрес, URL) на другой сервер, другую Web-страницу или фрагмент просматриваемого документа (блок).

При работе с гиперссылками широко используется технология "клиентсервер". Документы заранее размещаются на компьютерах, которые постоянно подключены к глобальной сети (на серверах, хостах), а с помощью браузера (программа-клиент, предназначенная для поиска информации в Internet) пользователь может выбирать и просматривать (или прослушивать) хранящиеся на сервере документы. Чтобы «добраться» до хранящегося на сервере файла, необходимо в адресной строке браузера указать путь к этому файлу, то есть адрес или, как его еще называют, ресурс URL.

# 1.1.2 История развития HTML

ка.

В 1989 году Тим Бернерс-Ли предложил руководству международного центра высоких энергий (CERN) проект распределенной гипертекстовой системы, которую он назвал World Wide Web (WWW), Всемирная паутина. Первоначально идея системы состояла в том, чтобы при помощи гипертекстовой навигационной системы объединить все множество информационных ресурсов CERN в единую информационную систему. Технология оказалась настолько удачной, что дала толчок к развитию одной из самых популярных в мире глобальных информационных систем. Практически в сознании большинства пользователей глобальной компьютерной сети Internet сама эта сеть ассоциируется с тремя основными информационными технологиями:

- электронная почта (e-mail);
- файловые архивы FTP;
- World Wide Web.

Причем последняя технология постепенно перемещается на первое ме-

Успех технологии World Wide Web определен двумя основными факторами: простотой и использованием протоколов межсетевого обмена семейства TCP/IP, (Transmission Control Protocol, протокол управления передачей/Internet Protocol, протокол Internet), которые являются основой Internet.

Практически все пользователи Сети одновременно получили возможность попробовать себя в качестве создателей и читателей информационных материалов, опубликованных во Всемирной паутине. Но и популярность самого Internet во многом вызвана появлением World Wide Web, так как это первая сетевая технология, которая предоставила пользователю простой современный интерфейс для доступа к разнообразным сетевым ресурсам. Простота и удобство применения привели к росту числа пользователей WWW и привлекли внимание коммерческих структур. Далее процесс роста числа пользователей стал лавинообразным, и так продолжается до сих пор.

При этом сама технология на начальном этапе была чрезвычайно проста. Дело в том, что при разработке различных компонентов технологии (языка гипертекстовой разметки HTML (HyperText Markup Language, язык разметки гипертекста), протокола обмена гипертекстовой информацией HTTP, спецификации разработки прикладного программного обеспечения CGI и др.) предполагалось, что квалификация авторов информационных ресурсов и их оснащенность средствами вычислительной техники будут минимальными.

Одним из компонентов технологии создания распределенной гипертекстовой системы World Wide Web стал язык гипертекстовой разметки HTML, разработанный Тимом Бернерсом-Ли на основе стандарта языка разметки печатных документов — SGML (Standard Generalised Markup Language, стандартный обобщенный язык разметки). Дэниел В. Конноли написал для него Document Type Definition — формальное описание синтаксиса HTML в терминах SGML.

Разработчики HTML смогли решить две задачи:

- 1) предоставить дизайнерам гипертекстовых баз данных простое средство создания документов;
- 2) сделать это средство достаточно мощным, чтобы отразить имевшиеся на тот момент представления об интерфейсе пользователя гипертекстовых баз данных.

Первая задача была решена за счет выбора теговой модели описания документа. Такая модель широко применяется в системах подготовки документов для печати. Примером такой системы может служить хорошо известный язык разметки научных документов TeX, предложенный Американским математическим обществом, и программы его интерпретации.

Язык HTML позволяет размечать электронный документ, который отображается на экране с полиграфическим уровнем оформления; результирующий документ может содержать самые разнообразные метки, иллюстрации, аудио- и видеофрагменты и так далее. В состав языка вошли развитые средства для со-

сто

здания различных уровней заголовков, шрифтовых выделений, различные списки, таблицы и многое другое.

Вторым важным моментом, повлиявшим на судьбу HTML, стало то, что в качестве основы был выбран обычный текстовый файл. Выбор был сделан под влиянием следующих факторов: такой файл можно создать в любом текстовом редакторе на любой аппаратной платформе в среде какой угодно операционной системы; к моменту разработки HTML существовал американский стандарт для разработки сетевых информационных систем — Z39.50, в котором в качестве единицы хранения указывался простой текстовый файл в кодировке LATIN1, что соответствует US ASCII.

Таким образом, гипертекстовая база данных в концепции WWW — это набор текстовых файлов, размеченных на языке HTML, который определяет форму представления информации (разметка) и структуру связей между этими файлами и другими информационными ресурсами (гипертекстовые ссылки). Гипертекстовые ссылки, устанавливающие связи между текстовыми документами, постепенно стали объединять самые различные информационные ресурсы, в том числе звук и видео; в результате возникло новое понятие — гипермедиа.

Такой подход предполагает наличие еще одного компонента технологии — интерпретатора языка. В World Wide Web функции интерпретатора разделены между Web-сервером гипертекстовой базы данных и интерфейсом пользователя. Сервер, кроме доступа к документам и обработки гипертекстовых ссылок, обеспечивает предпроцессорную обработку документов, в то время как интерфейс пользователя осуществляет интерпретацию конструкций языка, связанных с представлением информации.

Первая версия языка (HTML 1.0) была направлена на представление языка как такового, где описание его возможностей носило скорее рекомендательный характер. Вторая версия языка (HTML 2.0) фиксировала практику использования его конструкций. Версия ++ (HTML++) представляла новые возможности, расширяя набор тегов HTML в сторону отображения научной информации и таблиц, а также улучшения стиля компоновки изображений и текста. Версия 3.2 смогла упорядочить все нововведения и согласовать их с существующей практикой. HTML 3.2 позволяет реализовать использование таблиц, выполнение кодов языка Java, обтекание графики текстом, а также отображение верхних и нижних индексов.

Сейчас World Wide Web Consortium (W3C) — международная организация, которая занимается подготовкой и распространением документации на описание новых версий HTML — уже опубликовала материалы спецификации HTML 4.01. Кроме возможностей разметки текста, включения мультимедиа и формирования гипертекстовых связей, уже существовавших в предыдущих версиях HTML, в версию 4.01 включены дополнительные средства работы с мультимедиа, языки программирования, таблицы стилей, упрощенные средства печати изображений и документов. Для управления сценариями просмотра страниц Website (гипертекстовой базы данных, выполненной в технологии World Wide Web) можно использовать языки программирования этих сценариев, например, JavaScript, Java и VBScript.

Усложнение HTML и появление языков программирования привело к тому, что разработка Web-узлов стала делом высокопрофессиональным, требующим специализации по направлениям деятельности и постоянного изучения новых Web-технологий. Но возможности Internet позволяют пользователям, владеющим основами HTML, создавать и размещать собственные Web-узлы без больших затрат. Именно на таких пользователей и рассчитан предлагаемый курс. Документы HTML обычно создаются в текстовых редакторах, сохраняются как файлы с расширение .htm, а затем просматриваются в программе (браузере) Internet Explorer. Составление программы на HTML чем-то напоминает набор текста в текстовом редакторе MS Word.

Особенность языка HTML заключается в том, невозможно достичь абсолютной точности воспроизведения исходного документа. Т.к., документ может быть широко доступен в Интернете, то неизвестно как будет организовано его воспроизведение (на графическом экране, или в чисто текстовом виде или др.).

## 1.1.3 Теговая модель языка

Управляющие конструкции языка называются тегами. Все теги заключаются в угловые скобки <...>. Сразу после открывающей скобки помещается ключевое слово, определяющее тег, например <BODY>. Теги бывают парными и непарными. Непарные теги оказывают воздействие на весь документ. Парные теги бывают открывающими и закрывающими и воздействуют они на ту часть документа, которая заключена между ними.

Например:

# <B> - открывающий тег, </B> - закрывающий тег.

Закрытие парных тегов выполняется так, чтобы выполнялось *правило* вложения: если тег открыт внутри какого-то тега, то и закрываться он должен внутри этого же тега. Например:

<B><I>На этот текст воздействуют два тега</I></B>.

Эффект применения тега может видоизменяться путем добавления атрибутов. В парных тегах атрибуты могут добавляться только к открывающему тегу.

Атрибуты – это дополнительные ключевые слова, отделяемые от слова, определяющего тег и от других атрибутов пробелами и размещаемые до символа >. Иногда требуется указание значения атрибута. Значение атрибута отделяется от ключевого слова атрибута знаком «=» и заключается в кавычки. Например:

## <HR ALIGN="LEFT" >

Тег <HR > позволяет создать на странице горизонтальную линейку, а атрибут ALIGN со значением LEFT размещает ее в левой части экрана.

## 1.1.4 Структура документа НТМL

Все документы HTML имеют одну и ту же структуру, определяемую фиксированным набором тегов.

Документ HTML всегда должен начинаться с тега <HTML> и заканчиваться соответствующим тегом </HTML>. Внутри документа выделяются два основных раздела: раздел заголовков и тело документа – идущие именно в таком порядке. Раздел заголовков содержит информацию, описывающую документ в целом, и ограничивается заголовками <HEAD> и </HEAD>. Раздел заголовка также содержит общий заголовок документа, ограниченный парным тегом <TITLE>.

Основное содержание размещается в теле документа, которое ограничивается парным тегом <BODY>.

Простейший правильный документ HTML, содержащий все теги, выглядит следующим образом:

> <HTML> <HEAD> <TITLE>Заголовок документа</TITLE> </HEAD> <BODY>Содержание документа</BODY> </HTML>

В браузере Internet Explorer страница отобразится так, как показано рисунке 1.



Рисунок 1 – Вид простейшей Web-страницы

## 1.1.5 Функциональные элементы HTML

Элементы, заключенные между тегами <BODY> и </BODY> делят на блочные и текстовые. Блочные элементы относятся к частям текста уровня абзаца. Текстовые элементы описывают свойства отдельных фраз и еще более мелких частей текста. Создавая документ HTML нужно следовать основным **правилам вложения элементов**, а именно:

- 1) элементы не должны пересекаться, т.е. если открывающий тег располагается внутри элемента, то и соответствующий закрывающий тег должен располагаться внутри этого же элемента,
- 2) блочные элементы могут содержать вложенные блочные и текстовые элементы,
- 3) текстовые элементы могут содержать вложенные текстовые элементы,
- 4) текстовые элементы не могут содержать вложенные блочные элементы.

# В большинстве документов основными функциональными элементами являются заголовки и абзацы.

Язык HTML поддерживает шесть уровней заголовков. Они задаются при помощи парных тегов от <H1> до <H6>. При отображении Web-документов на экране компьютера эти элементы отличаются размером шрифта, и расположением относительно последующего текста. Параметры каждого уровня заголовков описаны в таблице 1.

Уровень абзаца	Параметры
H1	Толстый, очень крупный шрифт, текст центрирован. Между заго- ловком и последующим текстом вставляется одна или две пустых строки. При выводе на принтер заголовок печатается на новой странице.
H2	Толстый крупный шрифт. Без отступа. До и после заголовка поме- щаются одна или две пустых строки.
Н3	Крупный шрифт. До и после заголовка помещаются одна или две пустые строки. С небольшим отступом.
H4	Толстый нормальный шрифт. До и после заголовка помещается пустая строка.
H5	Нормальный шрифт. Пустая строка ставится перед заголовком, но не после.
H6	Толстый шрифт. Перед заголовком ставится пустая строка.

Таблица 1 – Параметры уровней заголовков

На Web-странице абзацы разных уровней выглядят так, как показано на рисунке 2.



Рисунок 2 – Шесть уровней абзацев, отображенных на странице

HTML-код такой страницы выглядит следующим образом:

```
<HTML>
<HEAD>
<TITLE>Уровни абзацев</TITLE></HEAD>
<BODY>
<H1>Первый уровень абзацев</H1>
<H2>Второй уровень абзацев</H2>
<H3>Третий уровень абзацев</H3>
<H4>Четвертый уровень абзацев</H4>
<H5>Пятый уровень абзацев</H5>
<H6>Шестой уровень абзацев</H6>
</BODY>
</HTML>
```

Каждый абзац страницы задается парным тегом <P>. Язык HTML не содержит средств для создания «красной строки», поэтому при отображении на экране абзацы разделяются пустой строкой. Закрывающийся тег </P> является необязательным. Например, если часть HTML-кода записана так:

> <H1>Заголовок</H1> <Р>Первый абзац<Р>Второй абзац <H2>Заголовок второго уровня</H2>,

то на Web-странице она отобразится так, как показано на рисунке 3.



Рисунок 3 – Страница с заголовками двух уровней и двумя абзацами, расположенными между ними

В отличие от текстового редактора ввод символа Enter не является признаком конца абзаца. Язык HTML рассматривает символы конца строки и пробелы особым образом. Любая последовательность, состоящая только из пробелов и символов конца строки, при отображении рассматривается как одиночный пробел. Enter не осуществляет также и переход на новую строку. Для того, чтобы перейти на новую строку, используют тег <BR>.

В качестве разделителя абзацев используется также текстовая линейка. Она задается непарным тегом <HR>. К этому тегу можно применять атрибуты:

ALIGN – для выравнивания линейки по центру (значение center), по правому краю страницы (значение right) и по левому краю страницы (значение left),

SIZE – высота линейки в пикселах,

WIDTH – ширина в процентах относительно ширины экрана,

COLOR – цвет линии

Например,

<HR ALIGN="RIGHT" SIZE="10" WIDTH="50%">

Этот тег создает горизонтальную линейку шириной в 10 пикселов, занимающую половину экрана и расположенную справа.

#### 1.1.6 Форматирование текста

Основным тегом, управляющим форматированием текста является тег <FONT>. <FONT> - парный тег, управляющий параметрами шрифта.

Тег FONT должен обязательно содержать хотя бы один из атрибутов: COLOR, FACE или SIZE.

Атрибут SIZE задает размер шрифта в относительных единицах от -7 до 7 (от мелкого шрифта к крупному), например,

< FONT SIZE="6"> - шрифт шестого размера.

Атрибут COLOR задает цвет текста, выраженный текстовым значением, например,

< FONT COLOR ="GREEN"> - шрифт зеленого цвета.

Атрибут FACE задает гарнитуру шрифта. Значение этого атрибута есть название шрифта, например,

< FONT FACE="Times New Roman"> шрифт Times New Roman.

Кроме тега FONT существуют и другие теги, предназначенные для форматирования текста.

<СІТЕ> - парный тег для вывода цитат (отображаются курсивом),

<EM> - парный тег выделения,

<STRONG> - парный тег сильного выделения,

<U> - парный тег подчеркнутого шрифта,

<S> - парный тег зачеркнутого шрифта,

<B> - парный тег полужирного шрифта,

<I> - парный тег курсива,

<SUB> - парный тег нижнего индекса,

<SUP> - парный тег верхнего индекса,

<MARQUEE> - парный тег бегущей строки,

<BASEFONT> - парный тег задания параметров вывода текста по умолчанию. Также должен обязательно содержать хотя бы один из атрибутов: COLOR, FACE или SIZE.

## 1.1.7 Задания для самостоятельной работы

## Задание 1

Выполните задание по предложенному ниже алгоритму.

1) В текстовом редакторе Блокнот введите следующий документ:

<HTML>

<HEAD>

<TITLE>Заголовок документа</TITLE>

</HEAD>

<BODY>Содержание документа</BODY>

</HTML>

2) Сохраните этот документ под именем first.htm.

а. Замечание: перед сохранением выполните команду Пуск – Настройка – Свойства папки – Вид и убедитесь, что сброшен флажок Не показывать расширения для зарегистрированных типов *файлов*. В противном случае Блокнот автоматически присвоит файлу расширение .txt.

- Запустите программу Internet Explorer. Дайте команду Файл Открыть – first.htm. Просмотрите как отображается содержимое файла. Отличается ли оно от обычных текстовых документов? Закройте файл.
- 4) Откройте first.htm в приложении Блокнот.
- 5) Удалите весь текст, заключенный между тегами <BODY> и </BODY>. Текст, который будет вводиться в следующих пунктах, необходимо поместить после тега <BODY>, а его конкретное содержание придумывается самостоятельно.
- 6) Введите заголовок первого уровня, заключив его между тегами <H1> и </H1>.
- 7) Введите заголовок второго уровня, заключив его между тегами <H2> и </H2>.
- 8) Введите отдельный абзац текста, начиная его с тега <P>.
- 9) Введите тег горизонтальной линейки <HR>.
- 10) Введите еще один абзац, начиная его с тега <P>.
- 11) Сохраните документ под именем paragraph.htm.
- 12) Откройте этот файл в Internet Explorer. Обратите внимание, как влияют теги HTML на отображение соответствующих частей текста.
- 13) Откройте first.htm в приложении Блокнот. Удалите весь текст, заключенный между тегами <BODY> и </BODY>. Текст, который будет вводиться в следующих пунктах, необходимо поместить после тега <BODY>, а его конкретное содержание придумывается самостоятельно.
- 14) Введите тег <BASEFONT SIZE="5" COLOR="BROWN">. Этот тег задает вывод текста по умолчанию увеличенным шрифтом и коричневым цветом.
- 15) Введите произвольный абзац текста, начиная его с тега <P>. Он будет введен с параметрами, установленными тегом **BASEFONT**.
- 16) Введите теги: <P> <FONT SIZE="-2" FACE="ARIAL "COLOR="RED">
- 17) Введите очередной абзац текста, закончив его тегом </FONT>.
- 18)В следующем абзаце используйте по своему усмотрению парные теги <B>, <I>, <U>, <S>, <SUB>, <SUP>.
- 19) В следующем абзаце используйте по своему усмотрению парные теги <EM>, <STRONG>.
- 20) Сохраните полученный документ под именем format.htm.
- 21) Откройте этот файл в Internet Explorer и изучите, как элементы форматирования языка HTML воздействуют на отображения документа в целом.
- 22) Вернитесь в программу **Блокнот** и измените документ так, чтобы элементы, задающие форматирование были вложены друг в друга. Сохраните изменения.

23) Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы.

#### Задание 2

Создайте Web-страницу по предложенному ниже образцу (рисунок 4). Цветовой набор страницы настройте самостоятельно.



## Рисунок 4 – Образец страницы

## Задание 3

Создайте Web-страницу на тему: «Методики анализа заданий менеджеров», содержащую восемь абзацев, причем каждый абзац должен содержать 3-5 строк. При формировании абзацев дайте определение таким методикам, как Наблюдение, Самоописание/дневник, Метод критических случаев, Интервью с целью анализа работы, Иерархический анализ заданий, Исследование компонентов работы, Шкалирование востребованности возможностей. Воспользуйтесь для этого знаниями, полученными на специальных дисциплинах или глобальной сетью Интернет. Отформатируйте абзацы следующим образом:

- 1) Первые три абзаца являются заголовками первого, второго и третьего уровней.
- 2) Текст четвертого абзаца зеленого цвета, размер шрифта 1.
- 3) Текст пятого абзаца синего цвета, размер шрифта –1, начертание полужирное.
- 4) Четвертый и пятый абзац разделены горизонтальной линейкой.
- 5) Текст шестого абзаца желтого цвета, размер шрифта 4, курсив.
- 6) Текст седьмого абзаца коричневого цвета, размер шрифта 5, начертание подчеркнутое.

- Шестой и седьмой абзац разделены горизонтальной линейкой, шириной в 15 пикселов, занимающей половину ширины окна и расположенной слева.
- Текст восьмого абзаца черного цвета, размер шрифта 2, начертание сильное выделение; в тексте содержатся элементы с верхними и нижними индексами.
- 9) Весь текст заканчивается горизонтальной линейкой шириной в 10 пикселов, занимающей 35% ширины окна и расположенной справа.

## 1.1.8 Вопросы для самоконтроля

- 1) Как готовятся документы, наполняющие службу WWW, и из чего она состоит?
- 2) Расшифруйте аббревиатуру HTML и расскажите о задачах этого языка.
- 3) Что такое гипертекст, гиперссылка?
- 4) Расскажите историю развития языка HTML.
- 5) Дайте определение тегу. Какие теги называются парными, непарными?
- 6) Что такое атрибут? В чем его назначение, где и как он располагается?
- 7) В чем заключается правило вложения тегов?
- 8) Опишите структуру простейшего правильного документа HTML.
- 9) Какие функциональные элемент выделяются на Web-странице? Расскажите о правилах вложения функциональных элементов.
- 10) Как описать заголовки на HTML, и какие уровни заголовков поддерживает этот язык?
- 11) Перечислите теги, предназначенные для форматирования текста.

# 1.2 Гипертекстовые ссылки

## 1.2.1 Понятие гиперссылки

Гипертекстовая ссылка (рисунок 5) является фрагментом текста и позволяет ссылаться на какой-нибудь другой файл, имеющий расширение .htm. T.o, на одной Web-странице можно связывать несколько различных документов. Именно гиперссылки совместно с линиями связи и серверами участвуют в создании «паутины» WWW.

🚰 С:\Мои документы\intuit\courses\html\hs~1.html - Micros 💶 🗙				
айлравкаид	<u>И</u> збранное С <u>е</u> рвис <u>С</u> правка	-		
🗢 🗸 🔶   Назад Вперед	- 🐼 🗊 🚮 Остановить Обновить Домой	»		
🛛 🗛 рес 餐 кументы\intuit\	courses\html\hs~1.html 🔽 🔗 Переход 🗍 Сс	ылки »		
Этот текст содержит: <u>гипертекстовую ссылку</u>				
і 🛃 Готово	🛄 Мой компьютер			

Рисунок 5 - Web-страница, содержащая гипертекстовую ссылку.

Гиперссылка это — специальным образом выделенный фрагмент гипертекста (или графический объект), который содержит невидимый для пользователя указатель (адрес, URL) на другой сервер, другую Web-страницу или фрагмент просматриваемого документа (блок).

Гиперссылка состоит из "внутренней" части, то есть, адреса документа, на который она ссылается, и "внешней" части, видимой на экране и называемой *именем* гиперссылки. Имя гиперссылки может представлять из себя слово или группу слов, картинку или часть картинки. Если подвести указатель "мыши" к ссылке, он примет форму ладони с вытянутым указательным пальцем – и это самый надежный способ ее определить. При этом в строке состояния браузера можно будет прочитать адрес, на который указывает ссылка.

Кроме того, практически всегда по умолчанию текстовые ссылки выделены другим цветом и очень часто подчеркнуты. Графические ссылки часто заключены в рамку того же цвета, которым выделяются текстовые ссылки. По умолчанию это синий цвет, но каждая Web-страница может использовать свое собственное оформление. Цвет ссылок, которые уже выбирались пользователем, обычно отличается от цвета непосещенных ссылок и выглядит более "бледным" (по умолчанию – фиолетовый).

Для создания гиперссылки служит парный тег <A>, который обязательно должен иметь атрибут HREF для указания той страница, на которую произойдет ссылка.



Текст ссылки внутри тега, т.е. ее внешняя часть, <A> может быть любым.

Рассмотрим пример создания гиперссылки на страницу, расположенную на другом сервере по отношению к серверу, на котором расположен документ:

< A HREF =" http://www.midi.ru"> Midi-музыка </A>

Действие этого фрагмента программы таково: на экране появится гиперссылка **Міdi-музыка** (рисунок 6), при щелчке по которой произойдет переход на Web-страницу, расположенную по адресу <u>http://www.midi.ru</u>. Заметим, что здесь имя тега «а», имя атрибута — «hreft», а значение аргумента — это адрес в сети (URL).



Рисунок 6 – Гиперссылка на Web-странице

## 1.2.2 Части URL-адреса документа

Для указания местоположения отдельного ресурса сети используется указатель, который называется URL (Uniform Resource Locator). URL — это универсальный указатель ресурсов, адрес, который указывает путь к конкретному документу. Он содержит информацию об используемом имени сервера, сведения о папке, в которой хранится файл, и имени файла.

Чтобы научиться создавать ссылки, нужно иметь представление о том, как адресуются документы в Сети.

Полный адрес документа в сети называется URL (Uniform Resource Locator, принято читать "урл"). URL может состоять из следующих частей:

1) *префикс протокола*, т.е., указание на используемую сетевую службу. Основные протоколы, с которыми вы столкнетесь, приведены в таблице 2.

Протокол	Пояснение
http://	НТТР – основной протокол, обеспечивающий доступ к
	Web-страницам. Используется по умолчанию, поэтому часто
	не указывается в URL.
ftp://	Протокол передачи файлов FTP, позволяющий при помощи
	программы FTP-клиента обмениваться файлами с удален-
	ным компьютером.
mailto:	Доступ к электронной почте. Вслед за mailto: указывается
	адрес электронной почты, имеющий общий вид имя_пользо-
	вателя@адрес_почтового_домена.
file:///	Указывается вместо имени протокола при обращении к
	Web-странице, расположенной на локальной машине.

Таблица 2 - Основные префиксы протоколов в URL

- 2) доменное имя компьютера или его IP-адрес вместо доменного имени. Например, сервер НГАСУ имеет доменное имя <u>http://www.sibstrin.ru/</u> и IP-адрес 62.76.97.33;
- 3) номер порта, через который происходит взаимодействие с сервером. Перед номером порта ставится двоеточие. С точки зрения пользователя указание порта бывает полезно, например, для "принудительной" перекодировки документа. Так, адреса <u>http://www.newmail.ru:8100/</u> и <u>http://www.newmail.ru:8100/</u> адресуют один и тот же сервер, но в первом случае документ читается в кодировке KOI-8, а во втором – в кодировке Windows. Вообще же, номер порта включается в URL только при нестандартных настройках сервера;
- 4) имя файла на сервере, которое может включать и путь от корневого каталога сервера. В записи пути по дереву каталогов сервера используется символ '/', а не '\', как принято в Dos и Windows. Корневой каталог на сервере – совсем не обязательно "головной каталог диска", как на локальной машине, а при соединении с сервером мы получаем доступ не ко всем его папкам и файлам, а только к тем, которые на нем специальным образом "размещены" и открыты для просмотра через WWW.
- 5) *закладка*, позволяющая перейти в нужную часть документа. Имя закладки отделяется от имени файла символом '#'.

В целях совместимости имена файлов, размещаемых в Интернет, обычно строятся по правилам DOS, то есть состоят из латинских букв, цифр и символа подчеркивания и имеют длину не более 8 символов. С другой стороны, web-страницам принято давать раширение \*.html, a не \*.htm.

Следует также помнить, что URL чувствителен к регистру символов, то есть, <u>http://www.tht.ru/</u> и <u>http://www.tnt.ru/</u> – это не один и тот же адрес.

Например, один из файлов страницы кафедры прикладной математики НГАСУ имеет полный URL <u>http://www.sibstrin.ru/prikl/person.html</u>, из чего можно заключить, что он находится на сервере НГАСУ в папке с именем prikl.

Если в URL не указано имя html-файла, это означает, что документ имеет имя по умолчанию, которое может назначаться при администрировании сервера. Чаще всего это имя index.html, так что URL <u>http://www.host.ru/</u> может означать то же самое, что <u>http://www.host.ru/index.html</u>.

Ссылка, в которой указан полный URL документа, называется абсолютной. Абсолютные ссылки используются для связи с внешними ресурсами Интернет, URL которых известен нам и не меняется. Относительная ссылка указывает на документ, опуская общую адресную часть. Например, если из документа index.html нужно сослаться на документ test.html, находящийся в той же папке, это можно сделать ссылкой вида <a href="test.html">а href="test.html">документ test.html</a>.

При использовании относительной ссылки можно ссылаться на папки, которые являются как вложенными, так и родительскими по отношению к пап-

ке, в которой расположен исходный документ. Например, ссылка на рисунок с именем my.jpg, находящийся во вложенной папке images может иметь вид

<A HREF="images\my.jpg">посмотрите рисунок</A>.

Следует обратить внимание на то, что, если в адресе перехода не указан каталог (директория, папка), то переход будет выполнен внутри текущего каталога (папки). Если в адресе перехода не указан сервер, переход будет выполнен на страницу, расположенную на данном сервере, и указанном в адресе каталоге.

## 1.2.3 Создание якоря на Web-странице

С помощью гиперссылки можно также ссылаться и на любое место внутри самой страницы. Это делается при помощи так называемого якоря. Якорь задается парным тегом <A>, использующим обязательный атрибут NAME=. Значением этого атрибута является имя якоря - любая последовательность латинских букв и цифр. Таким образом, якорь указывает на то место страницы, на которое произойдет ссылка. Например, в самом начале Web-страницы отметим какое-то место якорем Jakor:

<A NAME="Jakor"> на это место страницы указывает якорь Jakor</A>.

*Замечание:* Название якоря может быть абсолютно любым, пользователь придумывает его сам.

Для ссылки на якорь в конце адреса гиперссылки через знак # указывается имя якоря.

< A HREF="C:\Mou документы\link.htm # Jakor">.

В рассматриваемом примере произойдет быстрое «листание» документа и пользователь мгновенно окажется в начале просматриваемого документа. Браузер будет искать указанную метку на данном сервере и на текущей странице.

Такие метки удобно использовать при создании больших документов (книг, отчетов, энциклопедий). В начале документа можно поместить оглавление, состоящее из ссылок на метки, расположенные в заголовках разделов

При разработке собственного web-сайта для перехода между его страницами используются, как правило, относительные ссылки, что позволяет просматривать сайт на локальной машине, не внося в него изменений, а также при необходимости легко переместить сайт на другой сервер.

Примеры различных видов ссылок приведены в таблице 3.

Обычная ссылка         Это обычная гипертекстовая ссылка. Целчок по ней вызовет переход к но- вому документу (в данном случае - к главной странице нашего сайта).           Почтовая ссылка         Это ссылка на почтовый адрист           mailto:pers@mail.ru         Это ссылка на почтовый адриску окна со- здания сообщения, которое можно от- править по указанному в ссылке адре- су. Если же ни одна почтовая програм- ма на машине не установая програм- ии файловые архивы. При щелчке по ция target при этом не указывается.           FTP-ссылка         Это ссылка на FTP-сервер, содержа- щий файловые архивы. При щелчке по ней браузер перейдет в режим FTP- клиента и покажет содержимое голов- ного каталога архива. При зтом, пане- ли инструментов браузера изменятся. Если ке на машине установлена внеш- няя программа поддержки FTP, она бу- дет запущена.           Ccылка на файл http://www.sibstrin.ru/prikl/archives/ inet2000,zip         Это ссылка на архивный файл типа * zip, находящийся на сервере. После щелчка по ней браузер обречки спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем полжении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe           Ссылка на картинку тttp://www.sibstrin.ru/prikl/ngasu.gif Нажиите CTRL и щелкните ссылку         Это ссылке на картинку типа *.gif, ко- торая откроется в новом окне.	Вид гиперссылки	Действие
http://www.sibstrin.ru/prikl         Щелчок по ней вызовет переход к но- вому документу (в данном случае - к главной странице нашего сайта).           Почтовая ссылка         Это ссылка на почтовый адрес. Щел- чок по ней приведет к запуску окна со- здания сообщения, которое можно от- править по указанному в ссылке адре- су. Если же ни одна почтовая програм- ма на машине не установая програм- пото скаталота архива. При этом, пане- ли инструментов браузера изменятся. Если же на машине установлена внеш- няя программа поддержки FTP, она бу- дет запущена.           Ссылка на файл         Это ссылка на архивый файл типа *.zip, находящийся на сервере. После шелчка по ней браузер обычно спра- пивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe           Ссылка на картинку тир://www.sibstrin.ru/prikl/ngasu.gif Нажиите СТRL и щелкните ссылку         Это ссылке откроет в но- торая откроется в новом окне.	Обычная ссылка	Это обычная гипертекстовая ссылка.
http://www.sibstrin.ru/prikl         вому документу (в данном случае - к главной странице нашего сайта).           Почтовая ссылка         Это ссылка на почтовый адрес. Щел- чок по ней приведет к запуску окна со- здания сообщения, которое можно от- править по указанному в ссылке адре- су. Если же ни одна почтовая програм- ма на машине не установлена, браузер сообщит об этом в окне диалога. Оп- ция target при этом не указывается.           FTP-ссылка         Это ссылка на FTP-сервер, содержа- щий файловые архивы. При щелчке по ней браузер перейдет в режим FTP- клиента и покажет содержимое голов- ного каталога архивы. При щелчке по ней браузер перейдет в режим FTP, она бу- дет запущена.           Cсылка на файл         Это ссылка на архивный файл типа *.zip, нахолящийся на сервере. После щелчка по ней браузер обычно спра- пивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о мнотих других типах файлов, в том числе об исполняемых программа *.exe           Ссылка на картинку тttp://www.sibstrin.ru/prikl/ngasu.gif Нажимите СТКL и щелкните ссылку         Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.		Щелчок по ней вызовет переход к но-
Почтовая ссылка       главной странице нашего сайта).         Почтовая ссылка       Это ссылка на почтовый адрес. Щелчок по ней приведет к запуску окна создания сообщения, которое можно отправить по указанному в ссылке адресу. Если же ни одна почтовая программа на машине не установлена, браузер сообщит об этом в окне диалога. Опщия target при этом не указывается.         FTP-ссылка       Это ссылка на FTP-сервер, содержащий файловые архивы. При щелчке по ней браузер перейдет в режим FTP-клиента и покажет содержимое головного каталога архива. При этом, панели инструментов браузера изменятся. Если же на машине установлена внешняя программа поддержки FTP, она будет запущена.         Ccылка на файл       Это ссылка на архивы При этом, панели инструментов браузера изменятся. Если же на машине установлена внешняя программа поддержки FTP, она будет запущена.         Ccылка на файл       Это ссылка на архивы при этом, панели инструментов браузера изменятся. Если же на машине установлена внешняя порграмма поддержки FTP, она будет запущена.         Ccылка на файл       Это ссылка на архивы файл типа         http://www.sibstrin.ru/prikl/archives/ inet2000.zip       Это ссылка на архивный файл типа         *.zip, находящийся на сервере. После щелчка по ней браузер обычно спращивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Это ссылка на картинку типа *.gif, которая откроется в новом окне.       Это ссылке откроет в но-торая откроется в новом окне.	http://www.sibstrin.ru/prikl	вому документу (в данном случае - к
Почтовая ссылка         Это ссылка на почтовый адрес. Щелчок по ней приведет к запуску окна создания сообщения, которое можно отправить по указанному в ссылке адресу. Если же ни одна почтовая программа на машине не установлена, браузер сообщит об этом в окне диалога. Опщия target при этом не указывается.           FTP-ссылка         Это ссылка на FTP-сервер, содержащий файловые архивы. При щелчке по ней браузер перейдет в режим FTP-клиента и покажет содержимое головного каталога архива. При этом, панели инструментов браузер изменятся. Если же на машине установлена внешния программа поддержки FTP, она булает и покажет содержимое головного каталога архива. При этом, панели инструментов браузер изменятся. Если же на машине установлена внешния программа поддержки FTP, она булает запущена.           Ссылка на файл         Это ссылка на архивный файл типа *.zip, находящийся на сервере. Послещина но сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe           Ссылка на картинку         Это ссылка на картинку типа *.gif, которая откроется в новом окне.		главной странице нашего сайта).
чок по ней приведет к запуску окна создания сообщения, которое можно отправить по указанному в ссылке адресу. Если же ни одна почтовая программа на машине не установлена, браузер сообщит об этом в окне диалога. Опция target при этом не указывается.         FTP-ссылка       Это ссылка на FTP-сервер, содержащий файловые архивы. При щелчке поней браузер перейдет в режим FTP-клиента и покажет содержимое головного каталога архива. При этом, панели инструментов браузера изменятся. Если же на машине установлена внешняя программа поддержки FTP, она будет запущена.         Cсылка на файл       Это ссылка на архивый файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спрашивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку тири//www.sibstrin.ru/prikl/ngasu.glf       Это ссылка на картинку типа *.gif, которая откроется в новом окне.         Покальная ссылка       Целчок по этой ссылке откроет в но-	Почтовая ссылка	Это ссылка на почтовый адрес. Щел-
mailto:pers@mail.ru       здания сообщения, которое можно отправить по указанному в ссылке адресу. Если же ни одна почтовая программа на машине не установлена, браузер сообщит об этом в окне диалога. Опция target при этом не указывается.         FTP-ссылка       Это ссылка на FTP-сервер, содержащий файловые архивы. При щелчке по ней браузер перейдет в режим FTP-клиента и покажет содержимое головного каталога архива. При этом, панели инструментов браузера изменятся. Если же на машине установлена внешняя программа поддержки FTP, она будет запущена.         Cсылка на файл       Это ссылка на архивый файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спрашивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Cсылка на картинку титр!//www.sibstrin.ru/prikl/ngasu.gif       Это ссылка на картинку типа *.gif, которая откроется в новом окне.         Покальная ссылка       Щелчок по этой ссылке откроет в но-		чок по ней приведет к запуску окна со-
править по указанному в ссылке адре- су. Если же ни одна почтовая програм- ма на машине не установлена, браузер сообщит об этом в окне диалога. Оп- ция target при этом не указывается.         FTP-ссылка       Это ссылка на FTP-сервер, содержа- щий файловые архивы. При шелчке по ней браузер перейдет в режим FTP- клиента и покажет содержимое голов- ного каталога архива. При этом, пане- ли инструментов браузера изменятся. Если же на машине установлена внеш- няя программа поддержки FTP, она бу- дет запущена.         Cсылка на файл       Это ссылка на архивный файл типа *.zip, находящийся на сервере. После шелчка по ней браузер обычно спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку тиtp://www.sibstrin.ru/prikl/ngasu.gif Нажмите СТRL и щелкните ссылку       Это ссылке откроет в но- торая откроется в новом окне.	<mark> mailto:pers@mail.ru</mark>	здания сообщения, которое можно от-
су. Если же ни одна почтовая программа на машине не установлена, браузер собщит об этом в окне диалога. Оп- ция target при этом не указывается.         FTP-ссылка       Это ссылка на FTP-сервер, содержа- щий файловые архивы. При щелчке по ней браузер перейдет в режим FTP- клиента и покажет содержимое голов- ного каталога архива. При этом, пане- ли инструментов браузера изменятся. Если же на машине установлена внеш- няя программа поддержки FTP, она бу- дет запущена.         Ссылка на файл       Это ссылка на архивый файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спра- пивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку тиtp://www.sibstrin.ru/prikl/ngasu.gif Нажмите СТRL и щелкните ссылку       Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.		править по указанному в ссылке адре-
Ма на машине не установлена, браузер сообщит об этом в окне диалога. Оп- ция target при этом не указывается.           FTP-ссылка         Это ссылка на FTP-сервер, содержа- щий файловые архивы. При щелчке по ней браузер перейдет в режим FTP- клиента и покажет содержимое голов- ного каталога архива. При этом, пане- ли инструментов браузера изменятся. Если же на машине установлена внеш- няя программа поддержки FTP, она бу- дет запущена.           Ссылка на файл         Это ссылка на архивный файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe           Ссылка на картинку тиtp://www.sibstrin.ru/prikl/ngasu.gif Нажмите СTRL и щелкните ссылку         Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.		су. Если же ни одна почтовая програм-
Ссобщит об этом в окне диалога. Оп- ция target при этом не указывается.           FTP-ссылка         Это ссылка на FTP-сервер, содержа- щий файловые архивы. При щелчке по ней браузер перейдет в режим FTP- клиента и покажет содержимое голов- ного каталога архива. При этом, пане- ли инструментов браузера изменятся. Если же на машине установлена внеш- няя программа поддержки FTP, она бу- дет запущена.           Ссылка на файл         Это ссылка на архивный файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спра- пивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe           Ссылка на картинку тttp://www.sibstrin.ru/prikl/ngasu.gif Нажмите CTRL и щелкните ссылку         Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.		ма на машине не установлена, браузер
ция target при этом не указывается.           FTP-ссылка         Это ссылка на FTP-сервер, содержа- щий файловые архивы. При щелчке по ней браузер перейдет в режим FTP- клиента и покажет содержимое голов- ного каталога архива. При этом, пане- ли инструментов браузера изменятся. Если же на машине установлена внеш- няя программа поддержки FTP, она бу- дет запущена.           Ссылка на файл         Это ссылка на архивный файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe           Ссылка на картинку тttp://www.sibstrin.ru/prikl/ngasu.gif Нажмите CTRL и щелкните ссылку         Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.           Докальная ссылка         Щелчок по этой ссылке откроет в но- ста		сообщит об этом в окне диалога. Оп-
FTP-ссылка       Это ссылка на FTP-сервер, содержащий файловые архивы. При щелчке по ней браузер перейдет в режим FTP-клиента и покажет содержимое головного каталога архива. При этом, панели инструментов браузера изменятся. Если же на машине установлена внешняя программа поддержки FTP, она будет запущена.         Ссылка на файл       Это ссылка на архивный файл типа         http://www.sibstrin.ru/prikl/archives/ inet2000.zip       Это ссылка на архивный файл типа         *.zip, находящийся на сервере. После щелчка по ней браузер обычно спрашивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку       Это ссылка на картинку типа *.gif, которая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-		ция target при этом не указывается.
ftp://ftp.nsk.su/       щий файловые архивы. При щелчке по ней браузер перейдет в режим FTP-клиента и покажет содержимое головного каталога архива. При этом, панели инструментов браузера изменятся. Если же на машине установлена внешняя программа поддержки FTP, она будет запущена.         Cсылка на файл       Это ссылка на архивный файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спрашивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Cсылка на картинку       Это ссылка на картинку типа *.gif, которая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-	<u> FTP-ссылка</u>	Это ссылка на FTP-сервер, содержа-
ftp://ftp.nsk.su/       ней браузер перейдет в режим FTP-клиента и покажет содержимое головного каталога архива. При этом, панели инструментов браузера изменятся. Если же на машине установлена внешняя программа поддержки FTP, она будет запущена.         Cсылка на файл       Это ссылка на архивный файл типа *.zip, находящийся на сервере. Послещелчка по ней браузер обычно спрашивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Cсылка на картинку       Это ссылка на картинку типа *.gif, которая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-		щий файловые архивы. При щелчке по
Клиента и покажет содержимое головного каталога архива. При этом, панели инструментов браузера изменятся. Если же на машине установлена внешняя программа поддержки FTP, она будет запущена.Ссылка на файл http://www.sibstrin.ru/prikl/archives/ inet2000.zipЭто ссылка на архивный файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exeСсылка на картинку http://www.sibstrin.ru/prikl/ngasu.gif Нажмите СТRL и щелкните ссылкаЭто ссылка на картинку типа *.gif, ко- торая откроется в новом окне.Локальная ссылкаЩелчок по этой ссылке откроет в но-	<mark>ftp://ftp.nsk.su/</mark>	ней браузер перейдет в режим FTP-
Пого каталога архива. При этом, пане- ли инструментов браузера изменятся. Если же на машине установлена внеш- няя программа поддержки FTP, она бу- дет запущена.         Ссылка на файл       Это ссылка на архивный файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку http://www.sibstrin.ru/prikl/ngasu.gif Haжмите CTRL и щелкните ссылку       Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-		клиента и покажет содержимое голов-
ли инструментов браузера изменятся. Если же на машине установлена внеш- няя программа поддержки FTP, она бу- дет запущена.Ссылка на файлЭто ссылка на архивный файл типа *.zip, находящийся на сервере. После щелчка по ней браузер обычно спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exeСсылка на картинку http://www.sibstrin.ru/prikl/ngasu.gif Нажмите CTRL и щелкните ссылкуЭто ссылка на картинку типа *.gif, ко- торая откроется в новом окне.Локальная ссылкаЩелчок по этой ссылке откроет в но-		ного каталога архива. При этом, пане-
Ссылка на файл       Это ссылка на архивный файл типа         http://www.sibstrin.ru/prikl/archives/ inet2000.zip       Это ссылка на архивный файл типа         *.zip, находящийся на сервере. После щелчка по ней браузер обычно спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку nttp://www.sibstrin.ru/prikl/ngasu.gif Нажмите СТRL и щелкните ссылку       Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-		ли инструментов браузера изменятся.
Ссылка на файл       Это ссылка на архивный файл типа         http://www.sibstrin.ru/prikl/archives/ inet2000.zip       Это ссылка на архивный файл типа         http://www.sibstrin.ru/prikl/archives/ inet2000.zip       щелчка по ней браузер обычно спра- шивает в окне диалога, что следует         сделать с архивом - открыть в текущем положении или сохранить на диске.       Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку nttp://www.sibstrin.ru/prikl/ngasu.gif       Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.         Докальная ссылка       Щелчок по этой ссылке откроет в но-		Если же на машине установлена внеш-
Ссылка на файл       Это ссылка на архивный файл типа         http://www.sibstrin.ru/prikl/archives/ inet2000.zip       Это ссылка на архивный файл типа         http://www.sibstrin.ru/prikl/archives/ inet2000.zip       щелчка по ней браузер обычно спра- шивает в окне диалога, что следует         сделать с архивом - открыть в текущем положении или сохранить на диске.       Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку nttp://www.sibstrin.ru/prikl/ngasu.gif       Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-		няя программа поддержки FTP, она бу-
Ссылка на файл       Это ссылка на архивный файл типа         http://www.sibstrin.ru/prikl/archives/ inet2000.zip       Это ссылка на архивный файл типа         http://www.sibstrin.ru/prikl/archives/ inet2000.zip       шелчка по ней браузер обычно спра- шивает в окне диалога, что следует         сделать с архивом - открыть в текущем положении или сохранить на диске.       такие же вопросы задаются и о многих         других типах файлов, в том числе об исполняемых программах *.exe       Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-		дет запущена.
<ul> <li>*.21р, находящиися на сервере. После щелчка по ней браузер обычно спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe</li> <li>Ссылка на картинку http://www.sibstrin.ru/prikl/ngasu.gif Нажмите CTRL и щелкните ссылку</li> <li>Локальная ссылка</li> <li>Щелчок по этой ссылке откроет в но-</li> </ul>	Ссылка на файл	Это ссылка на архивный файл типа
http://www.sibstrin.ru/prikl/archives/ inet2000.zip       щелчка по неи ораузер ооычно спра- шивает в окне диалога, что следует сделать с архивом - открыть в текущем положении или сохранить на диске. Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку nttp://www.sibstrin.ru/prikl/ngasu.gif       Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-		*. z1p, находящиися на сервере. После
ППес 2000.210       шивает в окне диалога, что следует         сделать с архивом - открыть в текущем       положении или сохранить на диске.         Такие же вопросы задаются и о многих       других типах файлов, в том числе об         исполняемых программах *.exe       Это ссылка на картинку         http://www.sibstrin.ru/prikl/ngasu.gif       Это ссылка на картинку типа *.gif, ко-         Покальная ссылка       Щелчок по этой ссылке откроет в но-	http://www.sibstrin.ru/priki/archives/	щелчка по неи ораузер обычно спра-
Сселать с архивом - открыть в текущем положении или сохранить на диске.         Такие же вопросы задаются и о многих других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку http://www.sibstrin.ru/prikl/ngasu.gif         Нажмите СТRL и щелкните ссылку         Докальная ссылка         Щелчок по этой ссылке откроет в но-	InetZUUU.ZID	шивает в окне диалога, что следует
Положении или сохранить на диске.         Такие же вопросы задаются и о многих         других типах файлов, в том числе об         исполняемых программах *.exe         Ссылка на картинку         http://www.sibstrin.ru/prikl/ngasu.gif         Нажмите СТRL и щелкните ссылку         Докальная ссылка         Щелчок по этой ссылке откроет в но-		сделать с архивом - открыть в текущем
Такие же вопросы задаются и о многих         других типах файлов, в том числе об         исполняемых программах *.exe         Осылка на картинку         это ссылка на картинку типа *.gif, ко-         торая откроется в новом окне.         Локальная ссылка         Целчок по этой ссылке откроет в но-		Положении или сохранить на диске.
Других типах файлов, в том числе об исполняемых программах *.exe         Ссылка на картинку http://www.sibstrin.ru/prikl/ngasu.gif         Нажмите CTRL и щелкните ссылку         Локальная ссылка         Щелчок по этой ссылке откроет в но-		такие же вопросы задаются и о многих
Ссылка на картинку       Это ссылка на картинку типа *.gif, ко- торая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-		$\mu_{\text{CHO}}$
этср://www.sibstrin.ru/prikl/ngasu.gif       Это ссылка на картинку типа .gn, ко-         Нажмите CTRL и щелкните ссылку       торая откроется в новом окне.         Локальная ссылка       Щелчок по этой ссылке откроет в но-		Это ссилка на картинки типа * gif ko
Нажмите CTRL и щелкните ссылку Локальная ссылка	otto://www.sibstrin.ru/prikl/pgasu.gif	TODAS OTKDOETCS & HOROM OKHE
<u>Локальная ссылка</u> Щелчок по этой ссылке откроет в но-	Нажмите CTRL и щелкните ссылку	
<u>Локальная ссылка</u> Щелчок по этой ссылке откроет в но-	-	
	Локальная ссылка	Щелчок по этой ссылке откроет в но-
вом окне содержимое диска С.	file:///c:\	вом окне содержимое диска С.

## 1.2.4 Задания для самостоятельной работы

#### Задание 1

Выполните задание по предложенному ниже алгоритму.

- 1) Откройте документ first.htm в программе Блокнот.
- 2) Удалите весь текст, находящийся между тегами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY>.
- 3) Введите фразу: «Текст до ссылки».
- 4) Введите тег: <A HREF="first.htm">.
- 5) Введите фразу: «Ссылка».
- 6) Введите закрывающий тег </А>.
- 7) Введите фразу: «Текст после ссылки».
- 8) Сохраните документ под именем link.htm.
- 9) Запустите приложение Internet Explorer.
- 10) Откройте файл link.htm.
- 11) Убедитесь в том, что текст между тегами <A> и </A> выделен как ссылка (цветом и подчеркиванием)
- 12) Щелкните по ссылке и убедитесь, что при этом загружается документ, на который указывает ссылка.
- 13) Щелкните на кнопке **Назад** и убедитесь, что Вы вернулись к предыдущей странице, но ссылка теперь отображается другим цветом и считается "просмотренной".
- 14) Вернитесь в приложение **Блокнот** и откройте в нем файл (если он закрыт link.htm).
- 15) Удалите весь текст, находящийся между тегами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY>.
- 16) Введите такой текст, чтобы его величина превышала длину окна браузера Internet Explorer, т.е., чтобы документ в этом окне можно было просматривать с помощью полос прокрутки.
- 17) Два или три слова первой строки текста заключите в тег <A NAME="Jakor"> </A>.
- 18) Последнее слово документа заключите в тег <A HREF="C:\Mou документы\link.htm # Jakor">.
- 19) Сохраните изменения в файле.
- 20) Откройте приложение Internet Explorer.
- 21) Откройте файл link.htm.
- 22) Щелкните по последнему слову Вашего текста, который является гиперссылкой.
- 23) Убедитесь, что после этого Вы видите начало, а не конец страницы, то есть то место, которое выделено якорем.

#### Задание 2

Создать Web-сайт, главной страницей которого будет Web-страница, предложенная ниже (рисунок 7). Расшифровку каждого понятия, входящего в список гиперссылок, найти в глобальной сети Интернет, и расположить на отдельной Web-странице, применив к тексту известные вам виды и способы форматирования.

одя из видения будущего кадровых служб и опираясь на опыт успешных корпо-раций, можно опреде - Microsoft Internet Explorer 📕			
Правка Вид Избранное Сервис Справка			
азад и 📀 и 😰 🚱 🌈 Гонск 👷 Избранное 🤣 😥 е 🍃 💹 и 🖵 🎎 💈 🖄			
🔄 D:(Исходя из видения будущего кадровых служб и опираясь на опыт успешных корпо.htm 💽 🎅 Пере			
<u>Исходя из видения будущего кадровых служб и опираясь на опыт успешных корпораций, можно определить ключевые роли для профессии менеджера по</u> <u>персоналу.</u>			
<u>1. Кадровый стратег</u> 2. Гуководитель службы управления персоналом 3. Кадровый техновог			

Рисунок 7 – Образец страницы

#### Задача З

<u>5. Исполнитель</u>
 Кадровый консультант

Найдите в глобальной сети Интернет информацию по теме «История развития кадрового менеджмента» и расположите ее на одной Web-странице, отформатировав текст по своему усмотрению. Выделите в тексте разделы и назовите их, расположив эти названия в виде списка гиперссылок в самом начале Web-страницы. Гиперссылки должны являться якорями на соответствующие разделы текста.

#### 1.2.5 Вопросы для самоконтроля

- 1) Как определить гиперссылку на странице?
- 2) Из каких частей состоит гиперссылка?
- 3) Из каких частей может состоять внутренняя часть гиперссылки?
- 4) Какой тег и какие атрибуты используются для создания гиперссылки на странице?
- 5) Какая гиперссылка является абсолютной? Относительной?
- 6) Что такое якорь? Каким способом можно создать якорь?
- 7) Какие виды гиперссылок вы знаете?

## 1.3 Графика и мультимедиа на Web-страницах

# 1.3.1 Основной цветовой набор Web-страницы. Варианты описания цвета

Выбор различных цветов для оформления Web-страницы может значительно улучшить внешний вид и "читабельность" документа.

Основной цветовой набор HTML-документа указывается через атрибуты тега <BODY>. Это такие атрибуты, как:

BGCOLOR="цвет"	цвет фона документа,
ТЕХТ="цвет"	цвет текста,
LINK="цвет"	цвет непосещенных ссылок,
ALINK="цвет"	цвет активной (нажатой в данный момент) ссылки,
VLINK="цвет"	цвет посещенных пользователем ссылок

Цвета можно указывать через их имена, либо через шестнадцатиричные коды. В первом случае имена цветов такие же, как в большинстве языков программирования. Все браузеры поддерживают стандартный цветовой набор Windows. На рисунке 8 приведены примеры некоторых цветов, поддерживаемых языком HTML.

Рисунок 8 - Некоторые цвета Windows, поддерживаемые браузерами

Aqua	Black	Blue	Fuchsia	Gray	Green	Lime
Navy	Olive	Purple	Red	Silver	Teal	Yellow

С полным вариантом таблицы цветов, вы можете ознакомиться в Приложении А.

Кроме того, цвета можно вводить, указывая информацию о яркости, красной, зеленой и синей компонент в шестнадцатиричном коде. В этом случае перед кодом цвета ставится символ #. Коды компонент могут принимать значения от 00 до FF включительно. Так, белому цвету соответствует код #FFFFFF, черному - #000000, красному - #FF0000, зеленому - #00FF00, синему - #0000FF и т.д. Для того, чтобы цвета документа корректно отображались различными браузерами и при 256-цветном режиме монитора, следует использовать только интенсивности компонент, равные 00, 33, 66, 99, СС и FF. Всего насчитывается 216 таких цветов. Впрочем, 256-цветных мониторов остается все меньше, и это требование постепенно теряет актуальность.

Пример указания цветов в теге <BODY>:

< BODY BGCOLOR="white" TEXT="black" LINK="#009900" ALINK="#006600" VLINK="#003300"> Указывая код цвета любым из перечисленных способов, можно изменять цвет любой части текста, заключая ее в тег вида <FONT COLOR="цвет">часть текста</FONT>.

# 1.3.2 Подготовка и размещение графических изображений на Web-страницах

Web-страницы становятся привлекательными, благодаря размещениям на них разнообразных графических объектов. Графические иллюстрации в большинстве случаев являются неотъемлемой частью Web- документов. Чтобы встроить изображение в HTML-документ, нужно заранее подготовить рисунок или фотографию в файлах с расширениями - .GIF, .JPG или PNG.

Формат .GIF - Graphics Interchange Format – графический формат обмена -обычно используется для хранения рисунков с четкими деталями, небольшим набором цветов (до 256) и возможностью анимации ("мультипликация" на Web-страницах, графические кнопки, "украшения"). Кроме того, формат .GIF поддерживает эффект прозрачности, то есть, точки определенного цвета можно сделать того же цвета, что и фон страницы. Второй полезный эффект - возможность черезстрочной загрузки изображения GIF - то есть, изображение может постепенно "проявляться" по мере загрузки страницы, что даст пользователю возможность уже на ранней стадии загрузки получить представление о содержании картинки.

Формат .JPG - Joint Photographic Experts Group – объединенная группа экспертов фотографии - используется для хранения полноцветной графики и фотоизображений. Этот формат поддерживает 24-битовую графику, то есть, 16.7 миллиона цветов. Благодаря компрессии изображений, столь многоцветные файлы .JPG имеют приемлемые размеры, но сжатие производится за счет некоторой потери качества. Поэтому при подготовке иллюстраций для Webстраницы всю предварительную обработку следует производить с несжатыми изображениями, например, в формате .BMP, а в .GIF или .JPG конвертировать только окончательный вариант картинки.

Формат .**PNG** - Portable Network Graphics – переносимая сетевая графика – имеет все преимущества формата GIF, делает файлы компактными. Но пока еще поддерживается не всеми браузерами.

Из доступных приложений эффективно управлять файлами в перечисленных форматах позволяет **Microsoft Photo Editor**, входящий в пакет программ Microsoft Office. Профессионалы используют более мощные графические пакеты, такие как **Adobe PhotoShop** и **Corel Draw**. Для подготовки изображений можно использовать любой графический редактор, позволяющий сохранять файлы в этих форматах (например, графический редактор Paint).

Рисунки хранятся на Web - узлах в отдельных файлах, но отображаются как элементы Web- страниц. Для вставки рисунка используется текстовый элемент, задаваемый тегом <IMG>, который обязательно должен содержать атрибут SRC, задающий адрес файла, содержащего рисунок.

Например,

<IMG SRC="picture1.gif">.

При отображении рисунка браузер использует его реальные размеры. Если же рисунок нужно отмасштабировать, применяют атрибуты WIDTH= и HEIGHT=, задающие ширину и высоту рисунка в пикселах.

*Примечание:* Задавать эти атрибуты лучше всегда, так как в этом случае еще до загрузки рисунка, браузер выделяет для его отображения нужное место, что ускоряет процесс загрузки.

Например,

<IMG SRC="picture1.gif" WIDTH="120" HEIGHT="50">.

Внешний вид Web - страницы зависит от того, как именно располагается рисунок на странице. Так как рисунок задается как текстовый элемент, находящийся внутри какого-то абзаца, по умолчанию он рассматривается как встроенное изображение, включаемое в строку текста. Чтобы изображение задавалось автономно, его включают в отдельный абзац. Для изображения рисунка используются атрибуты, названия и назначения которых собраны в таблице 4.

Название атрибута	Значения атрибута	Назначение атрибута
1	2	3
ALIGN=	BOTTOM	Нижняя граница изображения сов-
(атрибут взаимо-		мещается с основанием строки.
действия рисунка с	MIDDLE	Середина изображения совмещает-
текстом)		ся с серединой текстовой строки.
	ТОР	Верхняя граница изображения сов-
		мещается с верхним краем строки.
	LEFT	Изображение размещается у левого
		края страницы, а последующий
		текст размещается справа от него.
	RIGHT	Изображение размещается у право-
		го края страницы, а последующий
		текст размещается слева от него.
HSPACE=	Любое число	Величина промежутка между изоб-
		ражением и текстом (в пикселах)
		по горизонтали.
VSPACE=	Любое число	Величина промежутка между изоб-
		ражением и текстом (в пикселах)
		по вертикали.

Таблица 4 - Атрибуты изображения рисунка

Продолжение таблицы 4

1	2	3
ALT=	Нужный текст	Отображение альтернативного тек-
		ста для изображения. Атрибут ис-
		пользуется для тех случаев, если
		картинка не может быть загружена,
		т.е. нет средства для ее просмотра.
		Альтернативный текст отображает-
		ся в виде всплывающей подсказки,
		появляющейся при наведении на
		картинку указателя мыши.

Еще один способ применения изображений на Web-страницах состоит в использовании их в качестве фонового рисунка. Фоновый рисунок задается с помощью атрибута BACKGRAUND= в теге <BODY>. Значением этого атрибута является адрес файла с изображением.

Например, если вы используете в теге <BODY> следующий атрибут

<BODY BACKGRAUND= "aaa.gif" TEXT="YELLOW">,

то фоном Web-страницы будет являться файл aaa.gif, а текст на этом фоне выводится желтым цветом.

Рисунок может также являться и гиперссылкой. В этом случае он будет заключен в синюю рамку, и курсор мыши, попадая на него, изменит свою форму со стрелки на ладонь с вытянутым указательным пальцем, а при нажатии на любую часть рисунка произойдет загрузка той страницы, адрес которой прописан во внутренней ей части. Таким образом, рисунок будет являться внешней частью гиперссылки. Для того, чтобы сделать рисунок гиперссылкой, необходимо тег вставки рисунка вставить в качестве внешней части гиперссылки. Например,

<A HREF=" www.rambler.ru"><IMG SRC="picture1.gif"> </A>

## 1.3.3 Озвучивание Web-страницы, внедрение аудиофайлов

Звук на странице играет достаточно существенную роль, он создает эффект присутствия и не занимает места на экране. Внедрение звука делает страницу «живой», интересной. Однако, при размещении аудиофайлов следует учитывать, что они имеют большой объем и длительное время загрузки.

Существуют несколько распространенных форматов звуковых файлов, используемых в WWW:

**WAV** (Waveform Audio – «звук в форме волны», расширение .wav) – стандартный, популярный формат цифрового звука.

**MIDI** (Musical Instrument Digital Interface – «цифровой интерфейс музыкальных инструментов», расширение .midi) – формат не содержит никакой музыкальной обработки и представляет собой просто запись набора нот.

**MP3** (MPEG Audio Layer 3, расширение .mp3) – формат отличается высокой степенью сжатия данных за счет некоторого ухудшения качества звука.

При создании страниц можно использовать либо звуковой фон, либо внедрение звукового файла. Применение звукового фона самый простой путь озвучивания страницы. Для этого используется тег <BGSOUND>, который имеет перечень атрибутов, приведенных в таблице 5.

<BGSOUND SRC="URL-адрес">.

Однако звуковой фон лишает возможности пользователя управлением звуком. Управлять звуком можно, если встроить на страницу аудиоплейер. Это осуществляется с помощью тега <EMBED>, который размещается внутри тега <BODY>.

<ЕМВЕD SRC="URL-адрес ">.

Тег <EMBED> также имеет собственные атрибуты, характеристика которых отображена в таблице 5.

Название тега	Название	Значения атрибу-	Назначение тега или
	атрибута	та	атрибута
1	2	3	4
<bgsound/>			Тег внедрения звуково- го файла
	LOOP	целое число; "infinite"- повторе- ние, пока пользо- ватель находится на странице;	Определяет количество воспроизведений
	VOLUME	число в диапазоне (-10000;0)	Назначает уровень вос- произведения звука
	BALANCE	число в диапазоне (-10000;10000)	Настраивает стереоба- ланс
<embed/>			Тег встраивания в страницу аудиоплеера
	WIDTH	число; процент	устанавливает ширину панели управления проигрывателя
	HEIGHT	число; процент	устанавливает высоту панели управления проигрывателя

Таблица 5 – Атрибуты и их значения для тегов звуковых эффектов

#### Продолжение таблицы 5

1	2	3	4
	AUTOSTART	false – включение	назначает способ
		звука пользова-	включения звука
		телем;	
		true – автоматиче-	
		ское включение;	
	HIDDEN	false – отображает	управляет отображени-
		панель;	ем панели проигрыва-
		true – не отобража-	теля
		ет панель;	

Пример HTML-кода, внедряющего аудиофайл:

<HTML>

<HEAD>

<TITLE>Использование звука на Web-страницах</TITLE>

</ HEAD>

<BODY bgcolor="red" topmargin=14 leftmargin=34 bottommargin=14 rightmagin=14>

<FONT face="Arial, Arial Cyr" color="white"><H2 align="center">Пример внедрения аудиофайла</H2>

<HR color="lime" widht ="66%" size=10>

<EMBED src="C:\WINDOWS\Media\flourish.mid" autostart=false align=right> </EMBED> <!---внедрение аудиофайла при возможности включения звука пользователем, выравнивание панели проигрывателя по правому краю страницы-->

</BODY>

</HTML>

Вид полученной страницы представлен на рисунке 9.



Рисунок 9 - Вид Web-страницы с внедренным аудиофайлом

## 1.3.4 Форматы видеофайлов, внедрение видео

При создании Web-страниц используются следующие форматы видеофайлов:

**MPEG** (Moving Picture Expert Group – «группа экспертов по движущимся изображениям», расширение .mpg или .mpeg) – формат предназначен для сжатия как аудио, так и видеофайлов. Обеспечивает высокое качество и является стандартом в цифровом видео. Недостаток – большой размер.

AVI (Audio Video Interleaved, расширение .avi) – применяется для хранения и воспроизведения аудио- и видеоданных. Наиболее распространенный формат.

**QuickTime** (расширение .mov) – популярный формат, не требует длительной загрузки по сети.

Для внедрения видео в Web-страницу используется тег <EMBED>. Формат записи:

< EMBED SRC="URL-адрес">

Атрибуты и их значения для рассматриваемого тега такие же как и в случае внедрения аудиофайлов (см. п. 1.3.3).

Примером внедрения на страницу так называемого секундомера может служить следующая HTML-программа:

<HTML>

<HEAD>

<TITLE>Использование видео на Web-страницах</TITLE>

</ HEAD>

<BODY bgcolor="silver" topmargin=14 leftmargin=14 bottommargin=14 right-magin=14>

<FONT color="maroon"> <H1 align="center">Пример внедрения видеофайла</H1>

<EMBED src="C:\WINDOWS\clock.avi"> </EMBED>

</BODY>

</HTML>

Вид полученной страницы представлен на рисунке 10.



Рисунок 10 – Вид Web-страницы с внедренным видеофайлом

## 1.3.5 Задания для самостоятельной работы

## Задача 1

Выполните задание по предложенному ниже алгоритму.

- 1) Откройте документ first.htm в программе Блокнот.
- 2) Удалите весь текст, находящийся между тегами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY>.
- 3) В графическом редакторе **Paint** подготовьте рисунок, размером 150х150 пикселей.
- 4) Используя пункт меню Рисунок, снимите флажок с команды Непрозрачный фон.
- 5) Сохраните файл под именем picture1.gif
- 6) В документе first.htm введите тег: <IMG SRC="picture1.gif">
- 7) Запустите приложение Internet Explorer.
- 8) Откройте файл first.htm
- 9) Убедитесь в том, что созданный рисунок отобразился на странице.
- 10) Откройте файл first.htm в Блокноте и отредактируйте вашу программу, добавив в нее 4-5 строк, расположенных после тега вставки рисунка,

- 11) Примените к тегу <IMG> атрибут ALIGN, проверив все его возможные значения, обратив особое внимание на расположение рисунка относительно текста.
- 12) Вернитесь в программу Блокнот и добавьте в тег <IMG> атрибуты HSPACE и VSPACE, соответственно со значениями 100 и 100.
- 13) Вернитесь в браузер и обновите содержимое окна, нажав на пиктограмму **Обновить** на панели инструментов. Убедитесь, что размер рисунка изменился.
- 14) Вернитесь в Блокнот и добавьте в тегу <IMG> атрибут ALT, сделав его значением произвольный текст.
- 15) В браузере просмотрите изменения, совершив операцию зависания курсора мыши над рисунком.
- 16) Вернитесь в приложение **Блокнот** и заключите тег <IMG> в тег <A> со значением атрибута HREF="link.htm ",
- 17) В браузере убедитесь в том, что рисунок заключился в синюю рамку и является гиперссылкой на файл link.htm,
- 18) В графическом редакторе создайте рисунок image.jpg.
- 19) В программу HTML-кода внесите изменения, добавив к тегу <BODY> атрибут BACKGRAUND= " image.jpg".
- 20) Откройте браузер Internet Explorer и убедитесь в том, что рисунок image.jpg стал фоном страницы.

## Задача 2

Создайте Web- сайт, состоящий из трех страниц, посвященных одной теме.

Первая страница должна быть главной и содержать общее название; фоном ее является самостоятельно созданный рисунок.

На второй странице необходимо расположить два рисунка: слева и справа. Фоном страницы является серый цвет, а текст страницы имеет красный цвет. Текст должен содержать не меньше 10 строк.

Третья страница должна содержать, по крайней мере, один рисунок в любом месте страницы, причем он должен располагаться в 20 пикселах от основного текста, как по вертикали, так и по горизонтали.

Все три страницы должны быть связаны гиперссылками, а именно: на первой странице должны располагаться ссылки на вторую и третью страницы, а последняя страница должна содержать ссылку на первую.

Цвет текста и фона для первой и второй страницы выбрать самостоятельно.

Для каждого рисунка сайта создайте альтернативный текст.
#### Задача З

Создать Web-страницу, в которую внедрена схема должностной структуры некоторого предприятия с именами возможных специалистов. Схема должна быть создана самостоятельно в графическом редакторе. Озвучьте страницу.

#### 1.3.6 Вопросы для самоконтроля

- 1) Какие атрибуты устанавливают основной цветовой набор Web-страницы?
- 2) Перечислите известные Вам форматы графических изображений. Дайте им полную характеристику.
- 3) В какой форме возможна запись цвета для атрибутов, определяющих основной цветовой набор страницы?
- 4) Какие форматы рисунков поддерживает HTML?
- 5) Какие графические редакторы позволяют создавать рисунки, встраиваемые в Web-страницу?
- 6) При помощи какого тега можно разместить изображение на Webстранице?
- 7) Какие атрибуты расширяют возможности по нужному размещению рисунка на странице?
- 8) Как сделать рисунок гиперссылкой?
- 9) В чем преимущество внедрения звуковых файлов перед созданием звукового фона?
- 10) Каким образом осуществить внедрение аудиофайла. Назовите тег и укажите формат его описания.
- 11) Какую роль играет использование атрибутов width и height в процессе внедрения звука?
- 12) Каким образом можно управлять способом включения/отключения звука на странице?
- 13) Перечислите известные Вам форматы видеофайлов. Какой из них характеризуется наиболее высоким качеством воспроизведения?
- 14) Существует ли отличие между внедрением аудио- и видеоинформации на Web-страницу? Поясните Ваш ответ.

## 1.4 Таблицы на Web-страницах

## 1.4.1 Теги и атрибуты размещения таблиц на Web-страницах

Таблицы удобны для представления больших объемов данных, а также для точного размещения элементов Web-страниц.

Таблица в языке HTML задается при помощи парного тега <TABLE>. Заголовок таблицы определяется парным тегом <CAPTION>.

Строки таблицы задаются парными тегами <TR>.

Каждая строка таблицы содержит ячейки, которые могут относиться к двум разным типам: ячейки в заголовке таблицы и ячейки обычной строки.

Ячейки заголовка таблицы задаются парным тегом <TH>, а ячейки обычной строки парным тегом <TD>.

Замечание: закрывающие теги </TR>,</TD> можно опускать.

Каждая ячейка может содержать произвольный текст, а также любые теги HTML.

При отображении таблицы на экране компьютера происходит ее автоматическое форматирование с подбором размеров ячеек в соответствии с объемом размещаемой информации и заданными атрибутами. Атрибуты элементов позволяют оформить таблицу по своему усмотрению.

В таблице 6 приведена сводка основных атрибутов элементов таблицы.

Атрибут	Элемент	Назначение
1	2	3
COLSPAN="число"	Ячейка	Протяженность ячейки в несколь-
		ко столбцов, в количестве столб-
		цов
ROWSPAN="число"	Ячейка	Протяженность ячейки в несколь-
		ко строк, в количестве срок
BGCOLOR="цвет"	Таблица, стро-	Цвет фона
	ка	
	ячейка	
CELLSPACING="число"	Таблица	Зазор между ячейками в пикселах
CELLPADDING="число"	Таблица	Зазор между содержимым ячейки и
		ее границей в пикселах
BORDER="число"	Таблица	Отображение границ ячеек и внеш-
		ней рамки таблицы в пикселах
BORDERCOLOR="цвет".	Таблица, ячей-	Цвет границы таблицы
	ка	
ALIGN="выравнивание"	Таблица, стро-	Выравнивание текста относитель-
	ка, ячейка	но таблицы или ячейки, в которых
		он находится (left - выравнивает
		текст по левому краю ячейки,
		center - выравнивает текст по цен-
		тру, right - выравнивает текст по
		правому краю ячейки)

Таблица 6 – Основные атрибуты тегов таблиц

продолжение таоли	цы с	
1	2	3
VALIGN="выравнивание"	Ячейка	Выравнивание текста и графики внутри ячейки по вертикали (top - выравнивает содержимое ячейки по ее верхней границе, middle - центрирует содержимое ячейки по вертикали, bottom - выравнивает содержимое ячейки по ее нижней границе)
HSPACE="число"	Таблица	Размер свободного пространства слева и справа от таблицы, в пик- селах
VSPACE="число"	Таблица	Размер свободного пространства сверху и снизу от таблицы, в пик- селах
WIDTH="ширина"	Таблица	Требуемая ширина таблицы в пик- селах или в процентах от ширины окна бруазера

Продолжение таблицы 5

Рассмотрим примеры создания таблиц на HTML.

Пример 1. Создание таблицы с объединенными ячейками – рисунок 9.

	Соот	ветствие веса и р	оста	
		Характеристики		
		Средний рост, см	Средний вес, кг	
Пол	Мужской	177	73	
	Женский	166	65	

Рисунок 9 - Таблица с объединенными ячейками

Фрагмент HTML-кода, позволяющий описать такую таблицу.

```
<TABLE BORDER="1" CELLPADDING="5" CELSPACING="0">
<CAPTION> Соответствие веса и роста </CAPTION>
<TR>
<TD COLSPAN="2" ROWSPAN="2"></TD>
```

```
<TD COLSPAN ="2">Характеристики</TD>
</ TR >
< TR >
 <TD>Средний рост, см</TD>
 <TD>Средний вес, кг</TD>
</TR>
<TR ALIGN="center">
 <TD ROWSPAN="2">Пол</TD>
 <TD>Мужской</TD>
 <TD>177</TD>
 <TD>73</TD>
</TR>
<TR ALIGN="center">
 <TD>Женский</TD>
 <TD>166</TD>
 <TD>65</TD>
</TR>
</TABLE>
```

**Пример 2.** Следующая таблица использует 90% ширины окна браузера, рамку в 5 пикселов и некоторые способы выравнивания данных в ячейках (рисунок10).

Способы выравнивания данных		
Заголовок		
Данные, выровнен- ные по левому краю ячей- ки	Данные, выровненные по правому краю ячейки	
25% ширины та- блицы		

Рисунок 10 – Таблица, использующая 90% ширины окна браузера, рамку в 5 пикселов и некоторые способы выравнивания данных в ячейках

Фрагмент HTML-кода, позволяющий описать такую таблицу.

<TABLE WIDTH="90%" ALIGN="center" BORDER="5" CELL-PADDING="2" CELLSPASING="0">

<CAPTION> Способы выравнивания данных </CAPTION> <TR>

<TD ALIGN="center" COLSPAN="2">Заголовок</TD>

```
</TR>
<TR>
<TR>
<TD ALIGN="left">Данные, выровненные по <BR>
левому краю ячейки</TD>
<TD ALIGN="right">Данные, выровненные по <BR>
правому краю ячейки</TD>
</TR>
</TR>
</TR>
<TD WIDTH="25%">25% ширины таблицы</TD>
</TN>
</TN WIDTH="75%">75% ширины таблицы</TD>
</TR>
</TR>
```

**Пример 3**. Представленная на рисунке 11 таблица, содержит различные способы выравнивания с использованием атрибута VALIGN, а также заливку ячеек.



Рисунок 11 – Таблица, содержащая различные способы выравнивания с использованием атрибута VALIGN, а также заливку ячеек

Фрагмент HTML-кода, позволяющий описать такую таблицу. <TABLE BORDER="2" WIDTH="60%"> <CAPTION> Способы выравнивания данных </CAPTION> <TR BGCOLOR="red"> <TH colspan="4"> Ячейки заголовка <TR BGCOLOR="yellow"> <TD WIDTH="100">Атрибут VALIGN ocyществляет выравнивание текста и графики внутри ячейки по вертикали</TD> <TD VALIGN="top">верх</TD> <TD VALIGN="middle">середина</TD> <TD VALIGN="middle">середина</TD> </TD VALIGN="bottom">низ</TD> </TR>

<TR VALIGN="top" BGCOLOR="red">

<TD> VALIGN= top выравнивает

<BR> содержимое ячейки <BR>по ее верхней границе.</TD>

<TD>верх</TD>

<TD>верх</TD>

<TD>Bepx</TD>

</TR>

<TR VALIGN="middle" BGCOLOR="yellow">

<TD>VALIGN= middle центрирует <br>cодержимое ячейки <br>no вертикали.</TD>

<TD>cepeдина</TD>

<TD>середина</TD>

<TD>cepeдина</TD>

</TR>

<TR VALIGN="bottom" BGCOLOR="red">

<TD>VALIGN= bottom выравнивает <br>coдержимое ячейки <br>no ее нижней грани-це.</TD>

<TD>низ</TD> <TD>низ</TD> <TD>низ</TD> </TR> </TABLE>

*Примечание*. Обратите внимание на оформление таблиц - как и при написании программ, при "ручном" создании Web-страниц удобнее всего набирать операторы одного уровня с одинаковым сдвигом вправо.

Сложные и красивые эффекты могут быть достигнуты вложением таблиц друг в друга - внутренняя таблица при этом должна быть целиком вложена в тег <TD> внешней таблицы.

## 1.4.2 Задания для самостоятельной работы

#### Задание 1

Выполните задание по предложенному ниже алгоритму.

- 1) Создайте в Блокноте документ с именем tablic.htm,
- 2) Внесите в него стандартную структуру для создания страницы
- 3) <HTML>
- 4) <HEAD>
- 5) <TITLE>Таблица</TITLE>
- 6) </HEAD>
- 7) <BODY> </BODY>
- 8) </HTML>
- 9) Между тегами <BODY> и </BODY> введите тег <TABLE BORDER="3" WIDTH="80%">

- 10) Введите строку <CAPTION> Список сотрудников </CAPTION>
- 11) Первая строка таблицы будет содержать заголовки столбцов, определите ее следующим образом: <TR BGCOLOR="green">
- 12) <TH>ФИО <TH>Должность
- 13) Создайте последующие ячейки таблицы, открывая каждую строку тегом <TR>, а ячейку тегом <TD>. Строк в таблице должно быть 7,
- 14) Завершите таблицу тегом </TABLE>
- 15) Сохраните изменения в файле,
- 16) Запустите браузер и посмотрите, как отобразилась таблица в его окне,
- 17) Примените к содержимому таблицы все атрибуты по выравниванию текста, изменения границ, зазоров между ячейками и между границей ячейки и ее содержимым, а также атрибуты цвета, опираясь на таблицу 6.

#### Задание 2

Создайте таблицу, показанную на рисунке 12.

#### ФИЗИЧЕСКИЕ НАУКИ

Наука	Область изучения		
Акустика	Распространение звука		
Астрономия	Свойства мирового пространства; происхождение и развитие галактик, звезд и планетных систем; происхождение и развитие Вселенной. Включает астрофизику и космологию		
Атомная физика	Структура и свойсва атомов		
Геофизика	Применение физики для изучения Земли. Включает физику атмосферы, метеорологию, гидрологию,океанографию, изучение магнитного поля Земли, сейсмологию и вулканологию		
Гидродинамика	свойсива и поведение движущихся жидкостей и газов		
Квантовая физика	изучение квантовой природыф материи и энергии		
Математическая физика	Применение математических методов для решения задач физики		
Механика	Силы, взаимодействие и движение материальных объектов		
Молекулярная физика	Структура и свойства молекул		

Рисунок 12 – Таблица для задания 2

#### Задача З

Создать файл pooszhrenie.html, в котором в виде таблицы описать систему поощрений для работников торговой фирмы «Лотос». Применить к таблице различные виды форматирования.

#### 1.4.3 Вопросы для самоконтроля

1) Перечислите основные теги для создания таблицы на Web-странице.

- 2) Для чего предназначены атрибуты COLSPAN и ROWSPAN.
- 3) Как изменить цвет ячейки таблицы?
- 4) Как изменить внешний вид рамки таблицы?
- 5) Какие атрибуты предназначены для выравнивания данных в таблице?
- 6) Для чего предназначены атрибуты CELLSPACING и CELLPADDING?
- 7) Для чего предназначены атрибуты HSPACE и VSPACE?
- 8) Как изменить ширину таблицы относительно ширины экрана?

## 1.5 Списки на Web-страницах

## 1.5.1 Нумерованные списки на Web-страницах

Язык HTML поддерживает три вида списков: упорядоченные (нумерованные), неупорядоченные (ненумерованные) и списки определений.

Возможно создание вложенных списков, причем вложенный список может отличаться по типу от основного.

*Нумерованные списки* (рисунок 13) задаются с помощью парного тега <OL>. Список содержит, естественно, элементы списка, которые определяются парным тегом <LI>. Закрывающий тег </LI> можно опускать.



Рисунок 13 – Web-страница с информацией, расположенной в виде нумерованного списка

> <HTML> <HEAD> <TITLE>Нумерованные списки</TITLE></head> <BODY> <H1>Нумерованный список</H1> <OL> <LI>первый элемент <LI>второй элемент

```
<LI>третий элемент
</OL>
</BODY>
</HTML>
```

## 1.5.2 Ненумерованные списки на Web-страницах

*Ненумерованные списки* (рисунок 14) задаются парным тегом <UL>. Элементы списка задаются также тегом <LI>. С помощью атрибута TYPE тега <UL> можно задать вид маркера списка:

"disc" – круг, "square" – квадрат, "circle"- окружность.

Замечание: по умолчанию маркеры имеют атрибут "circle".



Рисунок 14 – Web-страница с информацией, расположенной в виде ненумерованного списка

```
<HTML>
<HEAD>
<TITLE> Ненумерованные списки</TITLE>
</HEAD>
<BODY>
<H1> Ненумерованный список</H1>
<UL TYPE="square">
<LI> первый элемент
<LI> второй элемент
<LI> третий элемент
<BODY>
</HTML>
```

## 1.5.3 Списки определений на Web-страницах

С помощью языка HTML можно создавать списки определений, т.е. определения понятий оформляются также как в словаре (рисунок 15). Списки определений задаются парным тегом <DL>. Внутри него текст оформляется в виде *термина*, который выделятся непарным тегом <DT> и определения, которое следует за непарным тегом <DD>.



Рисунок 15 – Web-страница с информацией, расположенной в виде списка определений

```
<HTML>
<HTML>
<HEAD>
<TITLE>Cписок определений</TITLE>
</HEAD>
<BODY>
<DL> <FONT SIZE="4" COLOR="red"> Список определений </FONT>
<DL> <FONT SIZE="4" COLOR="red"> Список определений </FONT>
<DL> <FONT SIZE="4" соцоя термина </DL>
<DT> Термин <DD> расшифровка термина </DL>
<DT> HTML <DD> Язык разметки гипертекста </DL>
<BODY>
</HTML>
```

#### 1.5.4 Задания для самостоятельной работы

#### Задание 1

Выполните задание по предложенному ниже алгоритму. 1) Создайте в Блокноте документ с именем spisok.htm, 2) Внесите в него стандартную структуру для создания страницы <HTML> <HEAD> </HEAD> <BODY> </BODY>

</HTML>

3) Между тегами <br/> <br/>BODY> и </br/>BODY> введите тег, открывающий нумерованный список <br/>  $<\!OL\!>$ 

4) Вставьте в документы элементы списка, начиная каждый из них тегом <LI>,

5) Сохраните изменения,

6) Открой те данный файл в браузере, и просмотрите, как отображается нумерованный список в нем,

7) Вернитесь в **Блокнот** и добавьте после нумерованного списка следующий фрагмент программы <UL TYPE="square">

<LI> первый элемент

<LI> второй элемент

<LI> третий элемент </UL>,

8) Сохраните изменения и просмотрите этот файл в браузере, обращая особое внимание на вид маркера,

9) Открыв данный файл в **Блокноте**, измените значения атрибута ТҮРЕ сначала на "disc", а затем, после просмотра файла в браузере, на "circle",

10) Добавьте в ваш НТМL-код фрагмент:

<DL> Список определений

<DT> Гипертекст <DD> - это текст, представленный в виде связанных блоков текста. При этом блоки текста могут находиться на просматриваемой Web-странице, на другой странице данного сервера, либо на другом сервере глобальной сети. Переход от блока к блоку осуществляется скачком с помощью гиперссылок.

<DT> Гиперссылка <DD> - специальным образом выделенный фрагмент гипертекста (или графический объект), который содержит невидимый для пользователя указатель (адрес, URL) на другой сервер, другую Web-страницу или фрагмент просматриваемого документа (блок) </DL>

#### Задание 2

Создайте Web-страницу, расположив на ней нижеприведенный текст в виде трех различных списков: нумерованного, маркированного (отмеченного маркером – диском) и списка определений).

"Рассматривая развитие социальной структуры современного общества, аналитики указывают на особое место в нём класса управленцев. Например, М. Кастельс даёт следующую упрощённую стратификационную модель, включающую четыре социальные группы:

- высший класс (менеджеры и профессионалы);
- средний класс (техники и квалифицированные рабочие);
- низший средний класс (работники торговли, конторские служащие и операторы);

низший класс (сфера обслуживания и сельскохозяйственные рабочие)".

## 1.5.5 Вопросы для самоконтроля

- 1) Какие виды списков поддерживает язык HTML?
- 2) Каким тегом задаются нумерованные списки?
- 3) Каким тегом задаются ненумерованные списки?
- 4) Какие атрибуты используются для изменения формы маркера. Отмечающего каждый элемент ненумерованного списка?
- 5) Какими тегами задаются списки определений и его элементы?

## 1.6 Фреймы на Web-страницах

## 1.6.1 Теги и атрибуты для описания фреймов

Фреймами или кадрами называют независимые окна внутри окна браузера, в которых могут одновременно отображаться разные документы.

Фреймы удобны при создании страниц, которые должны иметь как динамическое, так статическое содержимое. Например, узкий левый фрейм может содержать оглавление сайта, а широкий правый будет предназначен для вывода информации. Возможны также любые другие конфигурации (рисунок 16)



Рисунок 16 - Web-страница, разбитая на два фрейма

Документ, использующий фреймы, содержит только описание внешнего вида фреймов, которое выполняется в теге <FRAMESET>...</FRAMESET>, используемого вместо тега <BODY>. Это означает, что необходимо заранее подготовить файлы, которые затем отобразятся в кадрах, для того, чтобы точно описать структуру и вид фреймов.

У тега <FRAMESET> есть два главных атрибута - ROWS="список величин" и COLS="список величин", описывающие, соответственно, строки и столбцы таблицы фреймов.

"Список величин" представляет собой разделенный запятыми перечень значений в пикселах, процентах или относительных величинах. Например, тег <FRAMESET ROWS ="100,240,140"> создает сетку из 3 кадров, высота которых равна 100, 240 и 140 пикселов соответственно. Тег <FRAMESET COLS ="25%,75%"> создает 2 кадра с шириной 25 и 75 процентов от ширины окна браузера, а < FRAMESET COLS ="\*,2\*"> - 2 кадра с шириной 1/3 и 2/3 ширины окна браузера.

Эти способы можно использовать и совместно - например, <FRAME-SET COLS ="128,\*"> указывает оставить 128 пикселов слева под первый фрейм, а все остальное пространство - под второй. При определении обоих атрибутов ROWS и COLS, например, <FRAMESET ROWS ="\*,2\*" COLS ="\*,  $2^*$ "> получается сетка кадров.

Кроме указанных выше атрибутов, тег <FRAMESET> может содержать другие атрибуты. Перечислим их в таблице 7.

Название атрибу-	Возможное значе-	Назначение атрибута
та	ние атрибута	
BORDER	Число пикселов	указывает ширину обрамления всех
		рамок для всех кадров,
FRAMEBORDER	no, yes	включает или выключает отображе-
		ние обрамления кадров. В случае
		yes рамка имеет трехмерную фор-
		му, иначе она невидима, то есть
		имеет цвет фона окна по умолча-
		нию. Некоторые браузеры "понима-
		ют" эту опцию только в виде
		frameborder=1 или frameborder=0,
		поэтому обычно указывают оба
		способа
FRAMESPACING	Число пикселов	указывает ширину промежутка
		между смежными кадрами в писке-
		лах. Не действует в Netscape
		Navigator.

Таблица	7 – Атрибуты тега	FRAMESET
---------	-------------------	----------

Внутри тега <FRAMESET> находятся описания отдельных кадров, каждое в собственном теге <FRAME>, закрывать который не нужно. Число тегов <FRAME> должно быть равно числу кадров, определенных в теге <FRAME-SET>, при этом считается, что кадры описываются слева направо и сверху вниз. Перечислим основные атрибуты тега <FRAME> таблице 8.

Название атрибу-	Возможное значе-	Назначение атрибута
та	ние атрибута	
SRC	URL-адрес	URL исходного документа для
		данного кадра
NAME	ИМЯ	указывает имя кадра
SCROLLING	yes, no, auto	управляет линейками прокрутки
		кадр; значение может быть задано
		в виде yes (линейки есть всегда),
		no (линеек нет никогда) или auto
		(линейки появляются, если это
		необходимо)
NORESIZE		запрещает изменять размеры кад-
		ра; данный атрибут, указанный для
		конкретного кадра, влияет также и
		на все кадры, смежные с ним
FRAMEBORDER	no, yes	включает или выключает отобра-
		жение обрамления кадров. Указа-
		ние этого атрибута в теге
		<frame/> отменяет указание, сде-
		ланное в тоге <frameset> для</frameset>
		данного кадра и всех, смежных с
		НИМ
MAGREENHEIGHT	ширина в пикселах	задаёт размещение по верхней и
		нижней и сторонам кадра областей
		свободного пространства
MAGREENWIDTH	ширина в пикселах	задаёт размещение по боковым
		сторонам кадра областей свобод-
		ного пространства

Таблица 8 – Атрибуты тега FRAME

После того, как тег <FRAMESET> закрыт, можно использовать тег <NOFRAMES>...</NOFRAMES>, определяющий содержимое, которое будет выводиться браузерами, не поддерживающими кадры.

Для создания более сложных конфигураций кадров теги <FRAMESET> могут вкладываться друг в друга - внутренний тег <FRAMESET> может быть вложен вместо любого из тегов <FRAME>.

На рисунке 17 показан пример документа с фреймами:

🚰 D:\Новая папка (4)\FRAME.htm - Microsoft Internet Explorer			
Файл Правка Вид	Избранное Сервис Справка		
🕞 Назад 👻 🌍 🝷	💌 🖻 🏠 🔎 Поиск 🦟 Избранное 🚱 😒 - چ 📨 - 🛄 🏭 💈 🖄		
Адрес: 👰 D:\Новая па	anka (4)\FRAME.htm		
Здесь может быт	ь релама		
В этом кадре может находиться меню	В этом кадре может находиться основная информация		

Рисунок 17 – Web-страница, разбитая на три кадра

<FRAMESET ROWS="100,\*">

<FRAME SRC="kadr1.htm" NAME="top" SCROLLING="no" NORE-SIZE>

<FRAMESET COLS="128,\*">

<FRAME SRC="kadr2.htm" NAME="menu" SROLLING="auto"> <FRAME SRC="kadr3.htm" NAME="main" SCROLING="yes"> </FRAMESET>

Здесь верхний кадр высотой 100 пикселов может служить, например, для вывода рекламы, и связан с файлом kadr1.htm. Остальная часть окна разбита на 2 колонки. Левая колонка имеет ширину 128 пикселов и может быть предназначена для вывода меню (документ kadr2.htm). Третий кадр занимает основную часть окна и предназначен для вывода информации (файл kadr3.htm).

Для создания ссылки из одного кадра в другой достаточно указать в теге ссылки атрибут вида TARGET="имя кадра". Например, для ссылки из кадра menu в кадр main достаточно написать в документе menu.html ссылку вида <A HREF="URL нового документа" TARGET="main">...</A>. При щелчке по этой ссылке содержимое правого кадра, то есть документ index.html, будет заменено на новый документ.

При работе с кадрами можно также указывать атрибут TARGET с одним из значений:

- 1) \_*self* (загрузить в тот же кадр, откуда делается ссылка),
- 2) \_*parent* (загрузить в родительский для данного кадр; если такого нет результат действия аналогичен \_self)
- 3) \_top (загрузить в полное окно, разрушая все кадры).

## 1.6.2 Задания для самостоятельной работы

#### Задание 1

Выполните задание по предложенному ниже алгоритму.

- 1) В текстовом редакторе Блокнот наберите:
- <HTML>
  <HEAD>
  <TITLE>Фреймы</TITLE>
  </HEAD>
  <FRAMESET ROWS="50%,\*">
  <FRAMESET ROWS="50%,\*">
  <FRAMESET ROWS="40%,\*">
  <FRAMESET COLS="40%,\*">
  <FRAMESET COLS="40%,\*" NORESIZE>
  <FRAME SRC="spisok.htm">
  <FRAME SRC="link.htm"">
  </FRAMESET COLS="40%,\*"</pre>

</FRAMESET>

</HTML>,

- 2) Сохраните этот файл под именем frames.htm.
- 3) Откройте файл frames.htm через браузер Internet Explorer и изучите представление ранее созданных Web-страниц в отдельных фреймах.
- 4) Проверьте, в каком кадре можно изменить положение границы.
- 5) Щелкните по ссылке, отображенной в третьем кадре, и посмотрите, куда будет загружен документ, на который она ссылается.
- 6) Вернитесь в программу **Блокнот** и проверьте возможные значения всех атрибутов, перечисленных в таблицах 8 и 9. Каждое новое значение атрибутов просматривайте в браузере, изучая его действие.

## Задание 2

Создайте документ, содержащий 2 вертикально расположенных фрейма. Левый фрейм (узкий) предназначен для вывода меню, правый (широкий) - для вывода содержимого документов, на которые Вы ссылаетесь из меню. Размеры фреймов должны быть изменяемы. Оставьте небольшое свободное пространство между фреймами.

Создайте в левом фрейме меню для навигации по всем страницам Вашего сайта. Проверьте работу документа с фреймами.

## Задание 3

Дан текст, представленный на рисунке 18. Проанализировать этот текст и создать два списка – нумерованный, содержащий только называния характеристик ПК и список определений, расшифровывающий характеристики. Поместить эти списки на страницу, которая должна содержать три области, две из которых расположены по вертикали (столбцы), второй столбец, в свою очередь, разбит на две строки. В кадре 1 должен быть размещен нумерованный список, содержащий перечисление основных характеристик ПК, без расшифровки. Расшифровки должны быть загружены во второй кадр в виде списка определений.

Третий кадр должен выглядеть точно так же, как и страница на рисунке 15. Размеры кадров отрегулировать самостоятельно.



1.	Быстродействие, производительность, тактовая частота.
2.	Разрядность нашины
	Разрядность
	Это максимальное количество разрядов двончного числа, над которым одновременно может выполняться
	машинная операция, в том числе и операция передачи информации.
3.	Типы системного и локального интерфейсов
	$\circ$ Разные типы интерфейсов обеспечивают разные скорости передачи информации между узлами машины,
	позволяют подключать разное количество внешних устройств и различные их виды.
4.	Емкость оперативной памяти
	Опреативная память
	Это энергозависимая память, в которой хранятся программы и данные во время работы компьютера.
	<ul> <li>Емкость оперативной памяти измеряется в мегабайтах (1 Мбайт = 1024 Кбайта).</li> </ul>
5.	Емкость накопителя на жестких дисках (винчестера).
	Винчестер
	Это несъемный магнитный диск, находящийся внутри системного блока и предназначенный для записи и
	чтения информации.
6.	Тип видеомонитора и видеоадаптера
7.	Тип принтера.
	• Матричный
	• Струйный
	• Лазерный
8.	Имеющееся программное обеспечение и вид опреационной системы

Рисунок 18 - Текст задания 3

## 1.6.3 Вопросы для самоконтроля

- 1) Дайте определение фрейму. Для чего предназначаются фреймы?
- 2) Что собой представляет структура документа, описывающего страницу, разбитую на фреймы?
- 3) Каким тегом задается разбиение страницы на фреймы?
- 4) Каким может быть значение атрибутов, влияющих на размер и количество фреймов на Web-странице?
- 5) Каково назначение и возможные значения атрибута BORDER?
- 6) Каково назначение и возможные значения атрибута FRAMEBOR-DER?
- 7) Каково назначение и возможные значения атрибута FRAMESPAC-ING?

## 1.7 Картированные изображения

#### 1.7.1 Тег для создания меню-карты

Сегодня многие Web-страницы располагают интересной разновидностью меню - *картированными изображениями*, то есть, картинками, отдельные части которых чувствительны к нажатию кнопки "мыши". Обычно чувствительные части изображения связаны с HTML-документами, то есть, являются ссылками.

В качестве графического формата для изображения-карты, как правило, выбирается формат .GIF с чересстрочной загрузкой, который поддерживается практически всеми браузерами и позволяет выводить изображение, постепенно уточняя детали. Следует позаботиться о том, чтобы размер файла-изображения не был слишком большим - как и всегда при работе с графикой.

Использование изображения-карты вместо обычных гиперссылок предпочтительнее, если этого требует графический дизайн страницы. Однако не следует забывать о пользователях, браузер которых не поддерживает вывод графики или этот вывод отключен. Если на странице, помимо картированного изображения, не предусмотрено других средств навигации, они не смогут воспользоваться Вашей страницей. Поэтому наряду с картированным изображением Web-мастер обычно разрабатывает обычный текстовый вариант меню.

Изображение-карта может обслуживаться как сервером, так и на стороне клиента. Мы рассмотрим только последний вариант, так как для его реализации не нужно иных средств, кроме тегов HTML.

Для определения областей картинки, чувствительных к нажатию "мыши", их следует описать в специальном теге <MAP>, имеющем вид

<MAP NAME="имя\_карты">...</МАР>

Располагать тег <MAP> можно в любом месте документа, но для удобства это обычно делают в начале документа после тега <BODY>.

# 1.7.2 Описание зон меню-карты, чувствительных к нажатию кнопки мыши

Внутри тега <MAP> каждая чувствительная область изображения описывается в непарном теге <AREA>. Его общий вид следующий:

<AREA SHAPE="форма" COORDS="список координат" HREF="URL">

Рассмотрим возможные значения атрибутов тега AREA.

1) Атрибут **SHAPE** - определяет форму чувствительной области Форма может принимать одно из значений

- rect (прямоугольник),
- poly (многоугольник),
- circle (круг).

Замечание: Если эта опция не указана, подразумевается значение rect. При наложении двух зон действует значение, определенное первым тегом <AREA>.

2) Атрибут **COORDS** – определяет расположение чувствительной зоны на карте

Список координат зависит от выбранной формы и включает в себя координаты в пикселах, перечисленные через запятую. Как обычно, сначала указывается Х-, а затем Y-координата и при этом считается, что ось Y расположена сверху вниз от верхнего левого угла рисунка. Левый верхний угол имеет координаты (0,0). Для прямоугольника указываются координаты левого верхнего и правого нижнего углов, например,

<AREA SHAPE="rect" COORDS ="0,0,200,100">.

Для круга указываются X- и Y-координаты центра и радиус в пикселах, например,

< AREA SHAPE ="circle" COORDS ="263,200,50" HREF="1.html">.

Для многоугольника перечисляются пары X- и Y-координат вершин. Для надежности многоугольник должен быть замкнутым, то есть первая пара вершин совпадает с последней.

3) Атрибут **HREF** – определяет ссылку на файл для данной чувствительной зоны.

В атрибуте HREF ="URL" как обычно указывается URL-адрес документа, адресуемого данной областью. Если чувствительная область определена, но не связывается ни с одним документом, вместо данного атрибута указывается значение nohref.

Для связывания изображения с созданной картой достаточно в теге <IMG>, с помощью которого вставлено изображение, указать атрибут вида USEMAP="#имя карты".

Например,

<IMG SRC="1.jpg" WIDTH="526" HEIGHT="400" USEMAP="#a1">.

Здесь использована карта с именем a1, находящаяся в текущем документе.

Рассмотрим пример рисунка, являющего картой-меню. На рисунке 16, рисунок, находящейся на Web-странице, является картированным изображением и на нем определены три области, чувствительные к нажатию кнопки мыши: многоугольник (рисунок 19), круг (рисунок 20) и прямоугольник (рисунок 21).



Рисунок 19 – Зона «многоугольник» на картированном изображении Для создания такой карты используется следующий HTML-код. В нем рисунок, используемый для карты, находится в файле 29SC091.JPG, а карта, с которой он связан, называется abc.

```
<HTML>
  <HEAD>
  <TITLE>Картированное изображение</TITLE>
  </HEAD>
<BODY BGCOLOR="skyblue">
<IMG SRC="29SC091.JPG" WIDTH="700" HEIGHT="500"
USEMAP="#abc">
  <MAP NAME="abc">
  <AREA SHAPE="circle" COORDS="350,350,50" HREF="numer.htm"
  ALT="круг">
  <AREA SHAPE="poly" COORDS="2,250,100,80,350,400,10,350"
  HREF="nenumerov.htm" ALT="многоугольник">
  <AREA SHAPE="rect" COORDS="450,5, 650,250" HREF="oprede-
  lenii.htm" ALT="прямоугольник">
  </MAP>
<BODY>
```

#### </HTML>



Рисунок 20 – Зона «круг» на картированном изображении



Рисунок 21 – Зона «прямоугольник» на картированном изображении

## 1.7.3 Задания для самостоятельной работы

## Задание 1

Выполните задание по предложенному ниже алгоритму.

- 1) В графическом редакторе **Paint** создайте рисунок, содержащий круг прямоугольник и пятиугольник. Установите для него размер 200х200 пикселов и сохраните его под именем imag.gif.
- 2) Откройте Блокнот и запишите в файле
- <HTML>
- <HEAD>
- <TITLE>Meню-карта</TITLE>
- </HEAD>
- <BODY> <MAP NAME="risunok-karta">...</MAP>
- </BODY>
- </HTML>

Сохраните его под именем karta.htm.

- 3) Внутри тега МАР создайте три области, реагирующие на нажатие кнопки мыши: круг, прямоугольник и пятиугольник. Размеры, а также их расположение подберите в соответствии с размерами этих фигур на основном рисунке.
- 4) После определения областей запишите тег

<IMG SRC=" imag.gif " WIDTH="200" HEIGHT="200" USEMAP="# risunok-karta ">, сохраните изменения в файле.

5) Откройте файл karta.htm в браузере и посмотрите, как определены области, чувствительные к нажатию мыши по отношению к вашему рисунку.

#### Задание 2

Создайте в графическом редакторе изображение, которое будет являться основой для карты-меню на первой странице сайта некоторой туристической фирмы. Сайт, кроме главной страницы, должен содержать еще 5 страниц, и карта-меню должна давать возможность переместиться с разных частей рисунка на эти страницы.

#### Задание 3

Проанализировать качество результатов работы сотрудников некоторого отдела предприятия за последние два квартала рабочего года и составить столбчатую диаграмму, отражающую динамику процесса развития каждого работника отдела. Диаграмму создать в графическом редакторе или электронных таблицах Excel. Поместить ее на главную страницу Web-сайта. Сделать диаграмму меню-картой так, чтоб с каждого столбца диаграммы можно было загрузить страницу, расшифровывающую, в каких именно направлениях данный работник повысил свою квалификацию.

#### 1.7.4 Вопросы для самоконтроля

1) В чем заключается смысл использования картированного изображения на Web-странице?

- 2) Какой формат рисунка обычно выбирается для связывания рисунка с меню-картой?
- 3) Какой тег используется для создания карты?
- 4) В какой форме могут создаваться области, чувствительные для курсора мыши?
- 5) Какой атрибут предназначен для описания формы такой области?
- 6) Как сформировать список координат, необходимый для определения месторасположения зоны на карте?
- 7) Какой атрибут позволяет создать ссылку с зоны на необходимую Web-страницу?
- 8) Как связать созданную карту с рисунком?

#### 1.8 Формы и их обработка

#### 1.8.1 Тег и атрибуты тега для создания формы

Формы являются наиболее популярным способом "обратной связи" с пользователем. С помощью HTML можно создавать как простые формы, предполагающие выбор одного из нескольких ответов, так и сложные формы для заказов или для того, чтобы получить от пользователей страницы какие-либо комментарии и пожелания.

Форма представляет собой несколько полей, где пользователь может ввести некоторую информацию, либо выбрать какую-то опцию (рисунок 22). После того, как пользователь отправит информацию, она обрабатывается программой (скриптом), размещенной на сервере. Существует также возможность обрабатывать формы "на стороне клиента", встраивая в свои страницы скрипты, написанные на языках JavaScript и Visual Basic Script.

🚰 Формы - Microsoft Internet Explorer
Файл Правка Вид Избранное Сервис Справка
🔆 Назад 🗸 🕗 – 💌 🖉 🏠 🔎 Поиск 🤺 Избранное 🍪 🗟 • چ 🗷 • 🗔 鑬 😵
Адрес: 🖉 D:\Новая папка (4)\FORM.НТМ
АНКЕТА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ
Пожалуйста, введите ваше имя:
E-mail:
Укажите, к какой группе пользователей вы себя относите: Учащийся студент преподаватель
Какие из средств Интернет вы используете наиболее часто:
□ www
🗖 e-mail
🗖 ftp
Каким из браузеров Вы предпочитаете пользоваться для работы в Интернет: Internet Explorer 💌
Какую еще информацию вы хотели бы видеть на нашем сайте? Netscape Navigator
Neo Planet
Отправить анкету Очистить

Рисунок 22 – Web-страница с формой

Для создания формы на Web-странице используется специальный тег <FORM>. Тег <FORM> является парным тегом и имеет несколько атрибутов, перечисленных в таблице 9.

Название атрибу-	Возможные значе-	Назначение атрибута
та	ния атрибута	
ACTION	URL-адрес	Указывает URL-адрес, который примет и обработает данные фор- мы. Если эта опция не указана, данные отправляются по адресу страницы, на которой размещена форма.
METHOD	post, get	Указывает метод передачи данных программе-обработчику формы. "Стиль" может принимать одно из двух значений. Значение get, ис- пользуемое по умолчанию, пред- писывает посылать информацию формы вместе с URL, а значение post предписывает посылать ин- формацию формы отдельно от URL. Значение post используется обычно в случае отправки данных формы по электронной почте или при необходимости передавать значительный объем информации.
NAME	ИМЯ	Указывает имя формы. Это необ- ходимо, если требуется доступ к данным формы с помощью встроенного скрипта на JavaScript или Visual Basic Script, а также в том случае, если данные формы предназначены для отправки по электронной почте
ENCTYPE	Способ кодирования данных	Задает способ кодирования дан- ных формы. В случае отправки данных как текста указывается в виде enctype="text/plain".

Таблица 9 – Атрибуты тега FORM

Примеры:

Заголовок формы, используемой на поисковой машине Яndex, имеет вид:

<FORM NAME="web" method="get" action="/yandsearch">

Заголовок формы, отправляющей информацию на адрес E-mail, может иметь вид:

<FORM ACTION = "<u>mailto:vita@lvs.ru?subject=Internet-Test</u>" METHOD = "post" ENCTYPE="text/plain" NAME="Q" ONSUBMIT="return Validate()">

Здесь форма имеет имя Q, в качестве action указана отправка сообщения по E-mail на адрес vita@lvs.ru с темой "Internet-Test", атрибут enctype предписывает отправлять данные формы как текст, а атрибут onSubmit связывает отправку формы с функцией Validate(), написанной на Javascript. Этот прием широко используется для проверки корректности заполнения формы.

## 1.8.2 Теги описания полей формы

Внутри тега <FORM> находятся поля формы. К полям относят:

- многострочное поле ввода (рисунок 23),
- список с полосой прокрутки или раскрывающийся список (рисунок 24),
- текстовое поле (рисунок 25),
- поле для ввода пароля (рисунок 26),
- радиокнопки (рисунок 27),
- *флажки* (рисунок 28),
- кнопка для очистки формы (рисунок 29),
- кнопка для отправки данных (рисунок 30).

Какую еще информацию вы хотели бы видеть на нашем сай	re?

Рисунок 23 – Многострочное поле ввода

🛛 Каким из браузеров Вы предпочитаете пользоваться для работы в Интер	Het: Internet Explorer 🗾
	Internet Explorer
	Netscape Navigator
	Opera
	Neo Planet

Рисунок 24 – Раскрывающийся список



Рисунок 25 – Текстовое поле

Рисунок 26 – Поле ввода пароля



Рисунок 27 – Радиокнопки



Рисунок 28 – Флажки

Очистить

Рисунок 29 - Кнопка для очистки формы



Рисунок 30 - Кнопка для отправки данных

Для создания **многострочного поля ввода** используется парный тег <TEXTAREA>. Для данного тега могут использоваться следующие атрибуты:

a) COLS и ROWS, которые соответственно указывают число столбцов и строк в поле;

б) WRAP указывает на режим автоматического распределения текста в ячейке. Она может принимать одно из значений off (выключен), virtual (распределять текст по всей ячейке, но на сервер передавать как одну строку) или physical (распределять текст по всей ячейке и передавать на сервер так, как он отображается);

в) NAME обозначает имя поля и предполагается обязательной.

Текст, написанный в теге <TEXTAREA>, становится его значением по умолчанию.

Например, для создания многострочного поля ввода, показанного на рисунке 16, достаточно написать следующий фрагмент HTML-кода:

<B> Какую еще информацию вы хотели бы видеть на нашем сайте? </B>

<BR>

<TEXTAREA NAME="resume" ROWS="4" COLS="30" WRAP="physical"> </TEXTAREA>

Парный тег <SELECT> определяет в форме меню с одним или несколькими вариантами выбора или список с полосой прокрутки.

a) если атрибут SIZE указана в виде SIZE="1", отображение элементов будет организовано в виде ниспадающего меню, в противном случае будет использован список прокрутки, включающий указанное число элементов.

б) атрибут MULTIPLE, если она указана, разрешает выбирать из списка более одного значения. Это можно сделать, например, выбирая элементы списка при нажатой клавише Ctrl.

Внутри тега <SELECT> размещаются элементы меню или списка, каждый - в своем теге <OPTION>, имеющем общий вид

<OPTION VALUE ="значение" SELECTED> текст </OPTION>.

Опция VALUE указывает значение, возвращаемое программе обработки при выборе пользователем данной опции, атрибут SELECTED указывает на элемент списка, выбранный по умолчанию. Внутри тега < OPTION> пишется текст, видимый в меню или списке на экране.

Для того, чтобы создать радиокнопки, показанные на рисунке 20, используется следующий фрагмент программы.

<B>Каким из браузеров Вы предпочитаете пользоваться для работы в Интернет?</B>

<SELECT NAME="browsers"> <OPTION SELECTED> Internet Explorer <OPTION> Netscape Navigator <OPTION> Opera <OPTION> Neo Planet </SELECT> Для сбора информации такими способами, как текстовые поля, поля для ввода пароля, переключатели, флажки, кнопки для отправки данных и очистки формы предназначен непарный тег <INPUT>.

Для создания **поля ввода** в теге <INPUT> используется атрибут TYPE со значением text.

Например,

<INPUT TYPE="text" NAME="строка" MAXLENGTH="максимальный размер" SIZE="число символов"">

Атрибут MAXLENGTH ограничивает максимальную длину вводимого текста, а атрибут SIZE показывает максимальное количество отображаемых символов.

Например, для того, чтобы создать текстовое поле, показанное на рисунке 18, используется следующий фрагмент программы.

<B>Пожалуйста, введите ваше имя:</B> <BR> <INPUT TYPE="text" NAME="name" SIZE=30> <BR>

**Текстовое поле** с защитой **вводимых символов** (то есть с заменой их на звездочки, как принято при вводе паролей) можно создать, если заменить опцию TYPE="text" на TYPE="password". Остальные атрибуты поля ввода пароля те же самые, что и для текстового поля.

Например, для того, чтобы создать поле с защитой символов, показанное на рисунке 19, используется следующий фрагмент программы.

> <B>E-mail</B>: <BR> <INPUT TYPE="password" NAME="e-mail" SIZE=30>

Поле **Флажок** создается при помощи значения checkbox атрибута ТҮРЕ в теге INPUT.

Например,

```
<INPUT TYPE ="checkbox" NAME="строка" VALUE="строка"
CHECKED>
```

Атрибут VALUE здесь определяет значение, возвращаемое программе обработки при выборе пользователем флажка, атрибут CHECKED, если он указан, делает флажок выбранным по умолчанию.

Для того, чтобы создать флажки, показанные на рисунке 21, используется следующий фрагмент программы:

<В> Какие из средств Интернет вы используете наиболее часто: </B><BR>

<INPUT TYPE="checkbox" NAME="group" VALUE="www"> WWW <BR> <INPUT TYPE="checkbox" NAME="group" VALUE="e-mail"> e-mail <BR>

<INPUT TYPE="checkbox" NAME="group" VALUE="ftp"> ftp <BR>

Поле Радиокнопка создается при помощи значения radio атрибута ТУРЕ в теге INPUT.

Например,

<INPUT TYPE ="checkbox" NAME="строка" VALUE="строка" CHECKED>

Радиокнопки можно группировать, задавая одно и то же значение атрибута NAME. Атрибуты VALUE и CHECKED имеют те же значения, что и у флажка.

Для того, чтобы создать радиокнопки, показанные на рисунке 20, используется следующий фрагмент программы:

<В>Укажите, к какой группе пользователей вы себя относите:</В> <BR>

<INPUT TYPE="radio" NAME="group" VALUE="schoolboy"> учащийся <BR>

<INPUT TYPE="radio" NAME="group" VALUE="student"> студент <BR> <INPUT TYPE="radio" NAME="group" VALUE="teacher"> преподаватель<BR>

Кнопка для очистки формы создается при помощи значения reset атрибута ТҮРЕ в теге INPUT.

Например,

<INPUT TYPE="reset" VALUE="Очистить">

Значение атрибута VALUE позволяет указать надпись для кнопки. После очистки все элементы формы принимают значения, которые они имели по умолчанию.

Кнопка для отправки данных формы приложению-обработчику создается при помощи значения reset атрибута ТҮРЕ в теге INPUT.

Например,

<INPUT TYPE="submit" VALUE="Отправить анкету">

Атрибут VALUE позволяет указать надпись на кнопке.

Для создания произвольной кнопки вместо reset или submit указывается значение button. В основном это требуется при написании скриптов, работающих на стороне клиента. Например, кнопка, созданная тагом <INPUT TYPE="button" VALUE="Bывести" onClick="Look()"> подписана словом "Bывести", а при нажатии на нее вызывает функцию Look(), написанную на JavaScript.

Пример разработки формы:

Ваше имя: Ваш пол: Мужской Женский На каком курсе Вы учитесь?

Какие языки программирования Вы знаете? Паскаль Си Ассемблер Напишите несколько слов о себе:

Эта форма имеет следующий HTML-код: <FORM METHOD = "post" ACTION = "mailto:pers@mail.ru? SUBJECT=Information"> <PRE>Ваше имя: <INPUT NAME="name" TYPE="text" MAXLENGTH="40" SIZE="40"> <INPUT TYPE ="radio" ="pol" Ваш пол: NAME VALUE=="male">Мужской <INPUT TYPE ="radio" NAME ="pol" VALUE ="female">Женский На каком курсе Вы учитесь? <SELECT NAME="curs" SIZE="1"> <OPTION VALUE="1" SELECTED>Первый</option> <OPTION VALUE ="2">Второй</OPTION > <OPTION VALUE ="3">Третий</OPTION > <OPTION VALUE ="4">Четвертый</OPTION > <OPTION VALUE ="5">Пятый</OPTION > </SELECT > Какие языки программирования Вы знаете? <INPUT TYPE ="checkbox" NAME ="pascal" VALUE ="yes" СНЕСКЕД>Паскаль < NPUT TYPE ="checkbox" NAME ="c" VALUE ="yes">Си <INPUT TYPE ="checkbox" NAME ="asm" VALUE ="yes">Ассемблер Напишите несколько слов о себе:</PRE> ="about" ROWS="6" COLS="60" <TEXTAREA NAME WRAP="off"></textarea> <P>< INPUT TYPE ="submit" VALUE="Отправить данные"> < INPUT TYPE ="reset" VALUE="Очистить форму"> </FORM>

Эта форма пытается отправить данные по указанному в заголовке адресу электронной почты, используя тему письма "Information". Дальнейшие события зависят от того, установлена ли на машине пользователя почтовая программа-клиент. Если да, то информация будет отправлена, причем данные из полей формы придут в виде строк текста NAME="value", например, информация о курсе в виде curs=1. Если на машине нет программы электронной почты, Windows предупредит об этом.

Обратите внимание, что для лучшего выравнивания данных часть формы заключена в тег <PRE>, а в части, находящейся вне тега <PRE>, использованы стандартные для HTML способы форматирования текста, например, тег абзаца.

## 1.8.3 Задания для самостоятельной работы

#### Задание 1

Выполните задание по предложенному ниже алгоритму.

- 1) Опишите в программе **Блокнот** форму из примера, приведенного в пункте 1.8.2, и сохраните ее в файле под именем form.htm,
- 2) Откройте данный файл в браузере и просмотрите форму.

#### Задание 2

Создайте документ HTML, содержащий форму для заполнения заказа на товар. Заказ должен быть отправлен по электронной почте.

Форма предусматривает поле ввода номера банковского счета, поле ввода личного номера пользователя (этот номер следует защитить от "подглядывания" при вводе), список выбора товаров (можно выбрать как один, так и несколько товаров), поле ввода произвольного комментария пользователя.

Также пользователь должен указать один из двух методов оплаты - наличным или безналичным расчетом.

Пользователь должен иметь возможность указать, какую дополнительную информацию он хочет получить при ответе - данные об остатках на счете и/или список последних платежей по счету. Любая из этих двух опций может быть включена или выключена.

Предусмотрите также возможность сброса введенных в форму данных.

#### Задача З

Составить список вопросов для анкетирования сотрудников предприятия с целью тестирования их способностей, возможностей, знаний и умений на предмет выяснения соответствия определенной должности. Анкету поместить на Web-страницу в виде формы. Отправить полученные результаты на сайт предприятия или электронный ящик предприятия.

## 1.8.4 Вопросы для самоконтроля

- 1) В чем заключается предназначение форм на Web-странице?
- 2) Из чего состоит форма и как информация от пользователя передается через форму?
- 3) Перечислите основные типы полей формы.
- 4) Какой тег предназначен для создания формы? Какие существуют атрибуты для него?
- 5) Приведите пример формы, отправляющей информацию пользователя по электронной почте.
- 6) Опишите все атрибуты тега, предназначенного для создания многострочного поля ввода.
- 7) Опишите все атрибуты тега, предназначенного для создания текстового поля.
- 8) Опишите все атрибуты тега, предназначенного для создания раскрывающего списка.
- 9) Опишите все атрибуты тега, предназначенного для списка с полосой прокрутки.
- 10) Опишите все атрибуты тега, предназначенного для создания полей типа флажок и радиокнопка.
- 11) Опишите все атрибуты тега, предназначенного для создания текстового поля с защитой символов.
- 12) Опишите все атрибуты тега, предназначенного для создания кнопок по очистке формы и ее отправке.

## 1.9 Этапы создания сайта и размещение его в глобальной

#### сети

## 1.9.1 Проектирование сайта

Проектирование сайта состоит из следующей совокупности этапов:

- 1) разработка структуры сайта;
- 2) выбор материала, размещаемого на сайте;
- 3) формирование содержимого папки сайта на диске;
- 4) выбор Web-сервера, на котором будет размещаться сайт.

Рассмотрим варианты представления структуры сайта:

- линейная структура;
- древовидная структура.

Линейная структура представляет собой простую цепочку страниц (рисунок 31), где к каждой следующей странице можно перейти через предыдущую и обратно. Таким образом, при такой структуре по сайту передвигаются в двух направлениях. Для этого на страницах такого сайта необходимы кнопки *Вперед* и *Назад*. Линейная структура применяется достаточно редко тогда, когда смысл документа понятен в случае прохождения всей последовательности страниц. Кроме этого, линейная структура сайта может иметь ответвления, а также альтернативные переходы (рисунок 32).



Рисунок 31 – Изображение линейной структуры сайта



Рисунок 32 - Изображение линейной структуры сайта с ответвлением

Чаще всего используется древовидная структура организации информации сайта, имеющая несколько уровней подчинения. Древовидная структура может иметь один уровень подчинения (рисунок 33) и тогда она называется плоской. Однако, если в такой структуре будет много страниц (большая ширина), то пользователю будет затруднительно удерживать в памяти множество вариантов выбора.



Рисунок 33 – Древовидная плоская структура сайта

Древовидная структура, имеющая несколько уровней вложенности и небольшую ширину, является оптимальным представлением сайта (рисунок 34), однако следует заранее определить это оптимальное соотношение между числом уровней (глубиной сайта) и количеством вариантов выбора на каждом уровне (шириной сайта). Для более удобного обращения пользователя с сайтом следует сделать ссылки на страницах верхнего уровня, по которым можно перейти не только на следующий уровень, но и более низкий и обратно. Чаще, древовидная структура используется не в чистом виде, а с использованием альтернативных переходов, создающих так называемые петли (рисунок 34).

После разработки структуры следует тщательно подобрать материал сайта, поскольку он будет размещен в Internet, и к нему будут обращаться дру-

гие люди. Информация, представленная на сайте должна быть достоверной, качественной, интересной. Весь материал необходимо разделить по группам (темам) и представить их на отдельных страницах. Информацию следует распределить по приоритету: наиболее важную поместить на верхние уровни сайта, побочную (дополнительную) - на низкие. Кроме этого, следует помнить о том, что сайт не должен быть перегружен большими объемами текста, лучше рационально сочетать помещение собственно текста на страницы и файлов с какойлибо информацией.



Рисунок 34 – Древовидная структура сайта с несколькими уровнями подчинения

Перед размещением следует правильно сформировать папку с файлами сайта:

- 1) создайте на диске папку, например my\_site;
- 2) скопируйте в папку все html-файлы, которые будут составлять Ваш сайт (при чем, среди файлов обязательно должен находиться файл index.html, представляющий начальную страницу сайта);
- 3) создайте в папке my\_site папку img, в которую скопируйте файлы, содержащие рисунки, фотографии и пр., используемые в сайте;
- 4) аналогичным образом сгруппируйте файлы с звуковыми эффектами и видеофайлы, например в папки sound и clip соответственно;
- 5) запустите начальную страницу index.html сайта из папки my\_site и убедитесь, что страницы отображаются в должном виде, а также сохранены переходы между страницами.

После того, как подготовлена папка сайта, необходимо найти Web-сервер, на котором сайт будет впоследствии размещен. Понятие Web-сервер уже достаточно известное, Web-сервер занимается хранением, поиском, обменом информацией в WWW. Работа клиента с Web-сервером осуществляется согласно протоколу HTTP (общий протокол, определяющий функционирование WWW) и ее можно описать следующим образом:

- 1) клиент Web-сервера делает запрос какого-либо файла;
- 2) программа просмотра (браузер) отправляет данный запрос на Webсервер, на котором находится требуемый файл;
- 3) сервер находит файл на своих дисках и отправляет его на компьютер клиенту.

При выборе сервера разработчики сайта учитывают его производительность, а также установленную цену за размещение сайта, последнее, как правило, играет главенствующую роль.

На сегодняшний день существуют Web-серверы, которые осуществляют бесплатное размещение сайтов, к таким можно отнести <u>www.boom.ru</u>, <u>www.chat.ru</u>, <u>www.narod.ru</u> и др.

#### 1.9.2 Размещение сайта

Рассмотрим алгоритм размещения созданного Вами сайта на Web-сервере <u>www.narod.ru</u> в Internet с помощью протокола HTTP:

- 1) осуществить подключение к Internet и загрузить начальную страницу сайта <u>www.narod.ru;</u>
- 2) в появившемся окне **Народ.Ру** (рисунок 35) щелкнуть по ссылке **Регистрация**;
- в появившемся окне страницы Регистрация: шаг 1 из 2, отображенной на рисунке 36, заполнить поля Логин, Имя, Фамилия, например как показано на рисунке 37;
- 4) щелкнуть по кнопке Дальше;



Рисунок 35 – Вид начальной страницы сайта <u>www.narod.ru</u>

🐴 Яндекс: Паспорт - Microsoft	Internet Explorer	_ 7 🗙
Файл Правка Вид Избранное	Сервис Справка	<b>1</b>
🚱 Назад 🝷 🕥 🖌 🗾	🚯 🔎 Поиск 👷 Избранное 🤣 🍙 - چ 🔳 - 📴 🏭 🦓	
Адрес: 🕘 http://passport.yandex.ru/c	gi-bin/Reg.pl?from=narod&mode=register&retpath=http://narod.yandex.ru/userarea/after_register.xhtml 🛛 💽 🎅 Переход	Ссылки »
Canon Casy-WebPrint •	😫 Print 📲 High Speed Print 🖳 Preview 🛛 🖉 Options 📄 Duplex 📲 View Print List	
AndexImage: TageSindexImage: Tage	СПОРТ Поношь сональные данные Платежные данные Настройка Яндекса	
Логин:	Регистрация: шаг 1 из 2  должен состоять из символов А-z, 0-9, -, начинаться с буквы, заканчиваться буквой или цифрой и содержать не более 20 символов	
	Существует мнение, что все хорошие логины на Яндексе уже заняты. Но это не так. Мы можем посоветовать вам интересный незанятый логин. Для этого достаточно указать имя и фамилию (они понадобятся и для дальнейшей регистрации).	
Ваше имя:		
Фамилия:		
	Яндекс охраняет персональные сведения пользователей в соответствии с <u>Соглашением о конфиденциальности</u> <u>информации</u> .	
	Дальше →	

Рисунок 36 - Окно страницы Регистрация: шаг 1 из 2 документа Пас-

Π	0	p	Т
	~	г	-

🚰 Яндекс: Паспорт - Microsoft Int	ernet Explorer	- 7 🛛
файл Правка Вид Избранное Се	арвис <u>С</u> правка	
🚱 Назад 🝷 🐑 🔺 🛃 🦿	№ Поиск 🌟 Избранное Ø 🔗 · 🌺 🕋 · 🛄 Ø 🗱 🍇	
Адрес: 🚳 http://passport.yandex.ru/cgi-b	sin/Reg.pl?from=narod&mode=register&retpath=http://narod.yandex.ru/userarea/after_register.xhtml 🛛 🚽 🗗 Переход	Ссылки »
Canon Casy-WebPrint - 🏻 🤅	🛱 Print 🕼 High Speed Print 🔩 Preview 🛛 🍸 Options 📄 Duplex 📲 View Print List	
Sindex Reprov	ПОРТ Понощь нальные данные Платежные данные Настройка Яндекса	
	Регистрация: шаг 1 из 2	
Логин:	timoffiel должен состоять из симаволов А-z, 0-9, -, неченаться с бухвы, заканчиваться бухвой или цифрой и содержать не более 20 симаолов Существует мнение, что все хороцие логины на Яндексе уже заняты. Но это не так. Мы можкем посоветовать вам интересный незанятый логин. Для этого достаточно указать имя и фамилию (они понадобятся и для дальнейшей регистрации).	
Ваше имя:	Тимофей	
Фамилия:	Петров Яндекс охраняет персональные сведения пользователей в соответствии с <u>Соглашением о конфиденциальности</u> информации.	
	Дальше →	

Рисунок 37 – Вид страницы с заполненными полями

- 5) в случае, если логин уже существует, сервер сообщит Вам об этом на дополнительной странице (рисунок 38) и предложит другие варианты;
- 6) из имеющихся вариантов списка логинов выбрать наиболее Вам подходящий и щелкнуть по кнопке **Дальше**;
|  |  | _ |
|--|--|---|
| 🚰 Яндекс: Паспорт - Microsoft Interr         | iet Explorer 📃 🖻 🛛   | J |
| Файл Правка Вид Избранное Серви              | к Справка 🦧  |   |
| 🚱 Назад 🝷 🐑 🔹 🛃 🏠                            | 🔎 Поиск 🤺 Избранное 🥝 🍛 · 🎍 🖀 · 🛄 🏭 🦓  |   |
| Адрес: 🙆 http://passport.yandex.ru/passpor   | t?mode=register&ncmd=577048 Ссылки 2   | * |
| Canon Easy-WebPrint - 🛛 🌚 P                  | rint 🕼 High Speed Print 🔍 Preview 🛛 🕄 Options 📄 Duplex 🔢 View Print List   |   |
| Andex Repconan                               | орт <u>Пенещь</u><br>ьные данные Платежные данные Настройка Яндекса  |   |
|  | Регистрация: шаг 1 из 2. Выбор логина  |   |
| Вы можете выбрать одно<br>из следующих имен: | timofej-petrov<br>timofepetro<br>timofepetro<br>petimofej<br>timofej<br>timo-petrov<br>pet-timofej<br>timofei1<br>timofei2007                      |   |
| Логин:                                       | timofej-petrov<br>должен состоять из символов А-z, 0-9, -, начинаться с буквы, заканчиваться буквой или<br>цифрой и содержать не более 20 символов |   |
| Ваше имя:                                    | Тимофей  |   |
| Фамилия:                                     | Петров<br>Яндекс охраняет персональные сведения пользователей в соответствии с <u>Соглашением о конфиденциальности</u><br>информации.              |   |
|  | Дальше→  | • |

Рисунок 38 – Страница сообщения о существовании выбранного логина

- 7) в появившемся окне **Регистрация: шаг 2 из 2** (рисунок 39) ввести пароль, соответствующий требованиям, указанным внизу данного поля, подтвердить его;
- 8) заполнить остальные поля: Контрольный вопрос, Ответ, Электронная почта, Контрольные цифры и пр. (рисунок 39);

🗿 Яндекс: Паспорт - Microsoft Inter	rnet Explorer		
<u>Ф</u> айл Правка <u>В</u> ид <u>И</u> збранное С <u>е</u> р	вис <u>С</u> правка		🥂
🚱 Назад 🔹 📀 - 💌 😰 🏠	🔎 Поиск 🤺 Избранное 🚱 🔗 -	🖕 💷 · 📙 🏭 📓 🦀	
Адрес: 🍯 http://passport.yandex.ru/passpo	ort?mode=register&ncrnd=628903		Переход Ссылки »
Canon Easy-WebPrint • 😩	Print 🚓 High Speed Print 🖳 Preview 🛛 🔀 C	Options 📄 Duplex 📋 View Print List	
	Ваш логин - timofej-petro∨		<u>^</u>
Пароль:	******	Проверить	Как правильно составлять пароль
		-	
	простой (9 символов)		
	Пароль должен содержать от 4 до 20 символо может совпадать с полином	в из списка: А-z, 0-9, ! @ # \$ % ^ & * ( ) + и не	
подтвердите пароль.			
	bollou politic		
Контрольный вопрос:	Почтовый индекс родителей	~	Если вы забудете пароль, вы сможете получить доступ, ответив на этот вопрос.
Ответ:	520023		
Электронная почта:		7	Вы сможете использовать этот адрес при
	(если есть)	_	работе со службами Яндекса. Если вы введете адрес, на него будет выслан запрос о его подтверждении.
Мобильный телефон:		7	Если вы забудете пароль вы сможете
incensionalin resteppin.	(если есть, в формате +7 999 888 77 66)		получить доступ, получив SMS сообщение с кодом на этот номер. Если вы введете номер, на него будет отправлен код подтверждения.
Контрольные цифры:	167958		Если вы не видите кнопку <ok> и картинку с</ok>
	введите, пожалуйста, число, которое вы видите справа	开口的方方	в вашем браузере отключена поддержка рафики.

Рисунок 39 – Страница Регистрация: шаг 2 из 2 документа Паспорт

- 9) щелкнуть по кнопке Ok;
- 10) в появившемся окне страницы сервер выдаст вам ваши данные и сообщение о том, что регистрация прошла успешно (рисунок 40);
- 11) при желании Вы можете в нижней серой области данной страницы зарегистрировать дополнительную информацию;
- 12) щелкнуть по кнопке Сохранить;
- 13) щелкнуть в текущем окне по ссылке Начать пользоваться сервером;
- 14) после чего появится страница Мастерская (рисунок 41); на данной странице представлены ссылки, предназначенные для формирования отдельных разделов, страниц, элементов вашего сайта: Мой дом, Общение, Редактирование и управление, Мои данные и др.; в правом верхнем углу страницы отображается адрес Вашего сайта, в данном случае Timofej-petrov.narod.ru

🗿 Яндекс: Паспорт - Місг	osoft Internet Explorer			_ @ 🗙
Файл Правка Вид Избра	анное Сервис Справка			
🕞 Назад 🔹 🕥 – 💌	💈 🏠 🔎 Поиск   לא Избран	нов 🚱 🔗 - 🌺 🔳 - 📴 🏭 🚺	3 3	
Agpec: 🙆 http://passport.yand	lex.ru/passport?mode=register&step=three	e&ncrnd=564470&idkey=5f033d750292c525c1214b61a	8а551d802 💽 🏹 Переход	Ссылки »
Canon Casy-WebPrin	t - 🛛 🚇 Print 🕮 High Speed Print	Preview Options Duplex V	iew Print List	
Under	паспорт	<u>Помощь</u>	🚺 <u>timofej-petrov</u> 🖪 <u>Выход</u>	<u>^</u>
Andex	Персональные данные Платеж	<u>ные данные Настройка Яндекса</u>		
	Готово!			
глобыльная консьютерная соль накториото	Регистрация успешно :	завершена.		
Я	Логин: timofej-petrov Зарегистрирован: 29.01.2007 Имя: Тимофей Фамилия: Петров	Контрольный вопрос: Почтовый индекс родителей Ответ: 520023 E-mail: imofepetrov@yandex.ru IIP: 89.237.5.14	Распечатать на случай, если вы забудете пароль	
	Держите ваш пароль в секрете и помн каким предлогом не спросят пароль по	ните, что сотрудники Яндекса никогда и ни под о телефону или электронной почте.		
	Начать пользоват	ъся сервисом		
	Дополнительная инфо (заполнять не обязательно)	рмация		
Обращение:	Не указано	Сфера моих занятий:	другое 💌	
Псевдоним:			Кодировка и формат писем,	
Пол:	Опредпочитаю не указывать		направляемых вам службой поддержки:	~

Рисунок 40 – Вид страницы с сообщением об удачной регистрации

🚰 Народ, Ру: Мастерская - Microsoft Internet Explorer			×
Файл Правка Вид Избранное Сервис Справка			7
🚱 Назад 🔹 🕥 - 💌 😰 🏠 🔎 Поиск 🌟 Из	бранное 🚱 🔗 - 🌺	🗹 - 📜 🇱 📓 👋	
Adpec: 🚳 http://narod.yandex.ru/userarea/after_register.xhtml?ncm	d=709211&idkey=28cb7f6343e5	3337be9e1ab4908ab08d602 🛛 🗸 Ссылки	»
Calloli Easy-WebPrint - Print Calloli Speed Print	int 🖳 Preview 🛛 🔀 Optio	ns 📄 Duplex 📔 View Print List	
Ялdex° 🛧 Народ			^
Почта	<u>Чаты</u>	Помощь Выход	
Мастерская		Проект: <u>Timofej-petrov.narod.ru</u>	
Создание сайта		Родактирование и управление	
Мой дом		<ul> <li>Статистика посещений ?</li> </ul>	
<ul> <li>Главная страница</li> <li>Про меня</li> </ul>		<ul> <li>Управление файлами и НТМL-редактор</li> </ul>	
◆ <u>Фотоаль Бом</u>		<ul> <li><u>загрузка файлов на саит</u></li> <li><u>Доступ к файлам по FTP</u></li> </ul>	
<ul> <li> <u>             Гезноме</u> <u> </u></li></ul>		<ul> <li><u>Индексирование в поисковых системах</u></li> </ul>	
<ul> <li>Для фанатов и поклонников</li> <li>Воментика</li> </ul>			
<ul> <li>◆ Объявление или приглашение</li> </ul>			
<ul> <li>Анкета</li> </ul>			
Мой бизнес			
<ul> <li>▲ <u>Деповая страница</u></li> </ul>		<ul> <li>Персональные данные (Паспорт) ?</li> </ul>	
<ul> <li><u>Праис-лист</u></li> <li><u>Загрузка прайс-листов из Excel</u></li> </ul>		◆ <u>Семейный фильтр</u>	
		<ul> <li>Уданить сайт</li> </ul>	
Общение			
<ul> <li> <u>         Персональный форум         </u> </li> </ul>			
<ul> <li>Персональный чат</li> <li>Опросы</li> </ul>			
<ul> <li>◆ <u>Сообщества</u></li> </ul>			
			~
ê		🧐 Интернет	
🯄 пуск 🕘 🔕 📓 🎽 🤎 🖬 Разработка Web-ca	🚰 Народ.Ру: Мастерск	📚 Сетевые подключе RL 🔇 🧾 🌺 22:5	3

1 – адрес размещаемого сайта

Рисунок 41 – Страница Мастерская создания сайта

- 15) щелкнуть по ссылке Главная страница для формирования ее параметров;
- 16) в появившемся окне страницы Выбор варианта шаблона «Главная страница сайта» выбрать подходящий шаблон или создать собственный с помощью соответствующего переключателя (рисунок 42);
- 17) щелкнуть по кнопке Готово, после чего появится страница с формой редактирования главной страницы сайта, на которой Вы можете управлять цветом фона, содержимым текста, его цветом, шрифтом, начертанием и пр., а также разместить необходимые ссылки их именами (рисунок 43) с помощью соответствующих элементов управления (при необходимости в любой момент редактирования вы можете сменить дизайн, щелкнув по одноименной кнопке страницы Мастерская); если требуется на главной странице разместить больше ссылок, то следует щелкнуть по кнопке Надо больше ссылок;

🚰 Народ, Ру: Выбор вариан	нта шаблона "Главная стра	ница сайта" - Microsoft Internet Explorer	
Файл Правка Вид Избран	ное Сервис Справка		A
🚱 Назад 🝷 📀 🕤 🗙	🗟 🏠 🔎 Поиск 👷 И	ізбранное 🚱 🔗 - 嫨 🔟 - 🗾 🗱 🛐	-28
Адрес: 🕘 http://narod.yandex.r	u/templ/preview.xhtml?template_id	=19&random_nocache=08523945be346775075	Переход Ссылки *
Canon Casy-WebPrint		Print 💽 Preview 🛛 🔀 Options 📄 Duplex 📔 Viev	Print List
Maananana / Parkan n		<u>Чаты</u> <u>110</u>	
<u>Мастерская</u> / Выоор в	арианта шаолона 1 лав	ная страница саита	Imotel-betrov.narod.ru
Выберите вариант шабл	тона, который наиболее г	юлно отвечает вашим требованиям	
С Станция станция с станция с с с с с с с с с с с с с с с с с с с	Сиреневый	о развития и при и при и и при и при и При и при и	
terneting	tin the second s		
систочники Состаточники Состаточники Конструктивный		о посторустичности Посторустичности Посторустичности Посторустичности Посторустичности Посторустичности Посторустичности Посторустичности Посторустичности Посторустичности Посторустичности Посторустичности Посторустичности Посторуст	

Рисунок 42 – Страница выбора шаблона дизайна главной страницы



Рисунок 43 – Страница настройки параметров дизайна главной страницы сайта

- 18) щелкнуть по кнопке **Готово**, после чего появится окно главной страницы Вашего сайта, которую можно просмотреть (рисунок 44);
- 19) щелкнуть по кнопке Готово или Еще редактировать, если это необходимо;



Рисунок 44 – Вид главной страницы сайта

- 20) после чего появится страница с сообщением о том, что страница index.html создана с указанием ее полного адреса, а также варианты действий, которые можно осуществить далее (рисунок 45);
- 21) щелкнуть по ссылке **Перейти** в раздел "Управление файлами", после чего откроется страница управления файлами сайта, представленная полями Имя файла, Описание, Размер, Дата (рисунок 46);
- щелкнуть по ссылке Загрузить файлы (загрузка будет осуществляться в корневую папку диска сервера, при необходимости предварительно можно создать свою папку с помощью ссылки Создать папку);

Файл Правка Вид Избранное Сервис Справка			
🔇 Назад 🝷 🐑 💌 😰 🏠 🔎 Поиск 🥱	🛃 Избранное 🧭 🔗 - 🍑 🗹 -	J 🛍 📓 🦓 👘 👘	
Адрес: 🕘 http://narod.yandex.ru/templ/after_create.xhtml?05	025145be36cb1f8d5&page_id=21796		💌 🋃 Переход
Calloll Casy-WebPrint - 🛛 🕮 Print 📲 High Sp	beed Print 💽 Preview 🛛 🔀 Options 📄 Du	lex 📔 View Print List	
ищех Даларод			
Почта	Чаты	<u>Помощь</u>	Выход
<u>Мастерская</u> / Страница index.html создана			Проект: <u>Timofej-petrov.n</u> a
Адрас вашей страницы <u>http://fimofei-petrov narc</u> Если писпросмотре отредактированной страницы вы н функция посволяет перезагружать странички непосред После создания этой страницы вы можете:	od ruýndex himl е видите изменений, то для обновления воспе твенно с сервера, ане из кеша.	пьзуйтесь клавищами Ctrl+Refresh	или Reload (Ctrl+R) в окне браузеј
Адрес в <u>ашей страницы http://fimofei-petrov narc</u> Если пистросмотре отредактированной страницы вы н функция позволяет перезагружать странички непосред После создания этой страницы вы можете: • <u>отредактировать ее</u> • <u>перейти к созданию других страниц</u>	od rufindex himi е видите изменений, то для обновления воспо твенно с сервера, ане из кеша.	пьзуйтесь клавишами Ctrl+Refresh	или Reload (Ctrl+R) в окне браузе
Адрес в <u>ашей страницы http://fimofei-petrov narc</u> Если пистросмотре отредактированной страницы вы н функция позволяет перезагружать странички непосред После создания этой страницы вы можете: • <u>отредактировать се</u> • <u>перейти к созданию других страниц</u> • <u>перейти в раздел "Управление файлами</u>	od rufindex himi е видите изменения то для обновления воспо твенно с сервера, ане из кеша.	пьзуйтесь клавишами Ctrl+Refresh	- или Reload (Ctrl+R) в окне браузе
Адрас вашей страницы <u>http://timofei-petrov narc</u> Если и спросмотре отредактированной страницы вы н функция позволяет перезагружать странички непосред После создания этой страницы вы можете: • <u>отредактировать се</u> • <u>перейти к созданию других страниц</u> • <u>перейти в раздел "Управление файлами</u> • <u>порекламировать свой сайт через служ</u>	ed rufindex himi е видите изменения, то для обновления воспо твенно с сервера, ане из кеша. " у "Яндекс. Директ"	пьзуйтесь клавишами Ctrl+Refresh	- или Reload (Ctrl+R) в окне браузе
Адрес в <u>ашей страницы http://timofei-petrov narc</u> Если постросмотре отредактированной страницы вы н функций позволяет перезагружать странички непосред После создания этой страницы вы можете: <u>отредактировать се</u> перейти к созданию других страниц перейти в раздел "Управление файлами порекламировать свой сайт через служб	ed rufindex himi е видите изменения то для обновления воспо твенно с сервера, ане из кеша.	пьзуйтесь клавишами Ctrl+Refresh	- или Reload (Ctrl+R) в окне браузе
Адрес в <u>ашей страницы http://fimofei-petrov narc</u> Если об спросмотре отредактированной страницы вы н функций позволяет перезагружать странички непосред После создания этой страницы вы можете: <u>отредактировать се</u> <u>перейти к созданию других страниц</u> <u>перейти в раздел "Управление файлами</u> <u>порекламировать свой сайт через служб</u> Вы также можете получить доступ к другим в	od rufindex html не видите изменения то для обновления воспо твенно с сервера, ане из кеша. " у "Яндекс. Директ" озможностям сайта, перейдя Б раздел	пьзуйтесь клавишами Ctrl+Refresh " <u>Мастерская"</u> .	- или Reload (Ctrl+R) в окне браузе
Адрес вашей страницы <u>http://fimofei-petrov narc</u> Если и и просмотре отредактированной страницы вы н функция позволяет перезагружать странички непосред После создания этой страницы вы можете: <u>отредактировать се</u> <u>перейти к созданию других страниц</u> <u>перейти в раздел "Управление файлами</u> <u>порекламировать свой сайт через служ</u> Вы также можете получить доступ к другим в	od rufindex html не видите изменения то для обновления воспо твенно с сервера, ане из кеша. " " у "Яндекс. Директ" озможностям сайта, перейдя в раздел	пьзуйтесь клавишами Ctrl+Refresh <u>"Мастерская"</u> .	- или Reload (Ctri+R) в окне браузе
Адрес вашей страницы <u>http://fimofei-petrov narc</u> Если по спросмотре отредактированной страницы вы не функций посволяет перезагружать странички непосред После создания этой страницы вы можете: <u>отредактировать се</u> <u>перейти к созданию других страниц</u> <u>перейти в раздел "Управление файлами</u> <u>порекламировать свой сайт через служб</u> Вы также можете получить доступ к другим в	od ruýndex html не видите изменения то для обновления воспо твенно с сервера, ане из кеша. " " у "Яндекс. Директ" озможностям сайта, перейдя в раздел	пьзуйтесь клавишами Ctrl+Refresh <u>"Мастерская"</u> .	или Reload (Ctri+R) в окне браузе

1- список ссылок дальнейших возможных действий пользователя

Рисунок 45 – Страница с сообщением о создании главной страницы

- 23) в появившемся окне страницы Загрузка файлов (рисунок 47) с помощью полей выбора файлов и кнопки Обзор выбрать на своем диске файлы Вашего сайта, после чего в каждом поле появится полный путь к нему;
- 24) щелкнуть по кнопке Загрузить файлы;
- 25) в появившемся окне страницы **Управление файлами** появится обновленный список файлов Вашего сайта (рисунок 48);
- 26) с помощью кнопок Копировать, Переместить, Удалить при необходимости Вы можете манипулировать файлами.

🕘 Ha	род, Ру: Управление фай.	лами - Microsoft Internet Explor	er					
Файл	Файл Правка Вид Избранное Сервис Справка 🦧							
<b>(</b> )	🄇 Назад 🔹 🐑 🔹 😰 🏠 🔎 Поиск 👷 Избранное 🤣 🎯 - چ 🔟 🔹 🛄 🍪							
Адрес:	: 🕘 http://narod.yandex.ru/file	manager/lf.xhtml?random_nocache=090	25145be36cbc0ac6			💙 芛 Переход 🛛 Ссылки 🎽		
Can	011 Easy-WebPrint -	🚇 Print 🛛 🖓 High Speed Print 🛛 🕻	Preview 🛛 🔀 Options	📄 Duplex 📗 📗 View Pr	int List			
Яnd	lex®					<u></u>		
	N	<u>Почта</u>	<u>Чаты</u>	Помоз	щь	Выход		
Ma	<u>стерская</u> / Управление	е файлами				Проект: <u>Timofej-petrov.narod.ru</u>		
			<u>Создать папку</u> <u>С</u>	оздать страницу по шаб	пону   <u>Создать html-файл</u>	<u>Загрузить файлы   Настройки   ?</u>		
	Отмеченное: Копировать	Переместить Удалить						
		Имя	Описание	<u>Размер</u>	<u>Дата</u>			
	🗐 Страницы по шабл	юну						
	🖉 index.html		Главная	1891 E	29.01.2007	Свойства		
	Отмеченное: Копировать Переместить Удалить							
	понок Найти с по Timofej-petrov.narod.ru С по Narod.ru С в интернете							

# Рисунок 46 – Вид страницы Управление файлами

📲 Народ, Ру: Загрузка файлов - Microsoft II	ternet Explorer		
Файл Правка Вид Избранное Сервис Спр	авка		//////////////////////////////////////
🚱 Назад 🝷 🕥 - 💽 🛃 🏠 🔎 Г	риск 👷 Избранное 🧭 🔗 - 🌺 🕅	- 🔜 🏭 📓 🦓	
Appec: 🗃 http://narod.yandex.ru/filemanager/fast_u	load.xhtml		💽 🔁 Переход Ссылки »
Canon Easy-WebPrint - 🛛 🕮 Print 🗃	High Speed Print 🖳 Preview 🛛 🔀 Options 🚦	Duplex View Print List	
Япdex® 🛧 Народ	Чатки	Памань	Выход
Мастерская / Управление файлами	Загрузка файлов	<u></u>	Проект: Timofej-petrov.narod.ru
	Sarpyзка в папку «Корневая пап //my_site/first.html Обзор //my_site/registration.h Обзор //my_site/registration.h Обзор //my_site/selcopr.html Обзор Объём каждого файла не долже При этом многотомный архив счи Загрузить файлы <u>Проверка свободно</u>	<ul> <li>Каж <u>Быорать папку</u></li> <li>Обзор</li>     &lt;</ul>	

Рисунок 47 – Окно страницы Загрузка файлов

Правка Вид Избранное Сервис Справка Назад • 🕥 - 💌 😰 🏠 🔎 Поиск 👷 Избранное : 💩 http://narod.yandex.ru/filemanager/lif.xhtmi?random_nocache=090 1011 Easy-WebPrint • 🔅 Print கூHigh Speed Print 🔹	52045be37608f5c4	🏭 🕅 🎕	\$	
Назад • 💿 - 💌 🗟 🏠 🔎 Поиск 👷 Избранное :: 🗿 http://narod.yandex.ru/filemanager/fi.xhtml?random_nocache=090 1011 Easu-WebPrint • 🔅 Print ::::::::::::::::::::::::::::::::::::	: 🚱 🔗 - 头 🕅 IS2045be37608f5c4 & Preview 🛛 🕄 Options	- 📙 🏭 🕅 🎕	8	
: 🕘 http://narod.yandex.ru/filemanager/lf.xhtml?random_nocache=090	52045be37608f5c4			
1011 Easy-WebPrint - 🔅 Print 🚎 High Speed Print 🛽	Preview 🛛 🔀 Options			💙 🔁 Переход Ссылки
		🖶 Duplex 📔 View Prir	nt List	
lex® A Народ	Чаты	Помоц	Б	Выхол
астерская / Управление файлами			*	Проект: <u>Timofej-petrov.narod.ru</u>
· · ·	Создать папку   С	оздать страницу по шабл	ону   Создать html-файл	Загрузить файлы   Настройки   ?
Отмеченное: Копировать Переместить Удалить				
Шмя	Описание	<u>Размер</u>	<u>Дата</u>	
🖲 Страницы по шаблону				
<u>index.html</u> ■	Главная	1891 B	29.01.2007	Свойства
🗎 Страницы				
<u> </u>		314 B	29.01.2007	Свойства
ℤ inlinef1.html		2070 B	29.01.2007	Свойства
<u>registration.html</u>		1960 B	29.01.2007	Свойотва
		1754 B	29.01.2007	Свойотва
Z table frame.html		1126 B	29.01.2007	Свойотва
Отмеченное: Копировать Переместить Удалить				

Рисунок 48 – Вид страницы **Управление файлов** с обновленным списком файлов после загрузки на сервер

## 1.9.3 Размещение сайта с помощью протокола FTP

При использовании протокола FTP компьютеры взаимодействуют между собой по технологии «клиент-сервер». Данный протокол допускает двусторонний обмен файлами между сервером и клиентом. Все файлы хранятся на FTP-сервере, к которому подключены компьютеры (FTP-клиенты). Клиент посылает на сервер запрос и получает требуемый файл. При этом допускается не только загрузка файлов на компьютер клиента, но и передача файлов с компьютера клиента на сервер. Для такого FTP-обмена на компьютере клиента должна быть установлена программа FTP-клиент. Эта программа может быть специализированной, а может просто иметь встроенный модуль для этого. К последним относятся известные Вам программные оболочки: Far-manager, Total Соттаnder и др. Рассмотрим процедуру размещения сайта с помощью программы Far-manager на FTP-сервере ftp://ftp.narod.ru.

Для этого необходимо знать адрес FTP-сервера (ftp://ftp.narod.ru.), зарегистрироваться на нем, т.е. получить имя пользователя и пароль (см. п. 1.9.2), а затем выполнить следующую последовательность действий:

- 1) запустить программу Far-manager;
- 2) известным вам способом открыть ниспадающее меню дисков на правой панели программы;
- выбрать в меню пункт FTP, после чего откроется список уже имеющихся соединений с FTP-сервером с данного компьютера (рисунок 56), либо, если их не было, то содержимое панели будет пусто;

- 4) на левой панели **Far-manager** открыть содержимое папки Вашего сайта my\_site;
- 5) нажать комбинацию клавиш Shift+F4;
- 6) в появившемся диалоговом окне Ввод адреса FTP, отображенном на рисунке 49, в верхнем текстовом поле ввести адрес FTP-сервера, имя пользователя и пароль согласно шаблону, либо, как показано на рисунке 50, ввести эти элементы отдельно: в первое поле только адрес сервера, а в многострочное текстовое поле – имя пользователя и пароль;
- 7) щелкнуть по кнопке Соединить;

💶 {FTP: } - Far				- 8 ×
и Имя D:\m	y_site <mark>Имя</mark> video.html zu1 html	и Сервер ftp.elf.stuba.sk	FTP: Home	23:58 Опис
sound first.html index.html		, oprozri sousarsh		
inlinef1.html inlinef2.html registration.html selcopr.html				
t1.html t2.html t3.html t4.html				
t5.html t6.html t7.html				
table_color3.html table_frame.html Sharewan	те версия Прави 20 м1 м7 22-22	Shares	лате версия ———	
18,520 байтов > 1Левая 2Правая 3Смо	верх 30.01.07 23:22 в в 17 файлах ————————————————————————————————————		ов в 1 файле —— Истор 9Видео 10	Іерево

Рисунок 49 – Вид содержимого панели Far-manager при выборе FTP-подключения

- 8) в заголовке правой панели появится адрес выбранного FTP-сервера, а в качестве содержимого будет выступать страница index.html;
- 9) обычным образом скопировать на сервер файлы Вашего сайта, представленные на левой панели;
- 10) после команды Копировать появится диалоговое окно Передача, в котором будет отображаться процесс передачи файлов на FTP-сервер (время, % загрузки, объем файлов и пр.) (рисунок 51);
- 11) после загрузки имена файлов появятся на сервере (рисунок 52).

🔢 Вв	од адреса FTP - Far		- 8 ×
и	D:\my_site Имя Имя	FTP: Home Home	23:58 Опис
i S f	Вво tp://пользователь:пароль@ce ftp://ftp.narod.ru	рд адреса FTP ———————————————————————————————————	₽
iir	По ьзователь:timofej-petrov lapoль: писание:		<b>↓</b>
s t t t	[] Зап ашивать пароль [] екстовый режим [] Пасси ный режим	[x] Рас иренное окно команд [x] Расширенный список	
t	I ] спользовать firewall [] Декодировать текст коман	Размер б фера : <u>512</u> IД	
tt	[ Выбрать та <mark>т</mark> лицу ] [ Сохра <mark>н</mark> и	[ Тип сервера ] [ Соединить ] пть ] [ Отменить ]	
•	Вверх 30.01.07 2 = 18 520 байтов в 17 файдах ==	13:22 Л	
) 1 <mark>9тил</mark>	ит 2 3 4HobURL 5Ko	пир 6Переим 7 8Удален 9Сохран	10Послдн

Рисунок 50 – Диалоговое окно Ввод адреса FTP

<b>1</b> {20.05	6}Передача - Far	- 8 ×
и img	D:\my_site FTP:ftp.narod.ru/ ==== MM9 Video.html 2v1.html first.html	
soun firs inde inli	inlinef2.html /inlinef2.html -Courset 2(12.6%)	
regi selc t1.h t2.h	-Current 3(17.6%) 1 011 00:00:01 1.011K 170b 1 011 00:00:00 <00:12:58 01-Feb > -[100%	
t3.h t4.h t5.h t6.h	3 709 00:00:17 218b 18 520 00:00:14 <00:13:12 01−Feb > -[ 20.0× -[ ]-[ ]-[ ]-[ ]-[ ]-	
t7.h tabl table_f 1	rame.html 8,520 байтов в 19 файлах 402 24 01 07 00:19 table frame.html 1126 29 01 07	21:04
1 D:\my_si 1_Помощь	8,520 байтов в 17 файлах — 17,509 байтов в 16 файлах te>_ 2ПользМ ЗПросм 4Редакт 5Копир 6Перен 2Папка 8Удален 9КонфМн 10	о Выход

Рисунок 51 – Диалоговое окно передачи файлов на сервер

🔢 {FTP:ftp.narod.ru/} - Far			
	u site	FTP:ftn g	arod ru/00:08
lu Mura		u Mara	Mug
и иля		и мгря	ROUN
11:1	Video.ntmi		
img	2V1.html	first.html	
sound		index.html	
first.html		inlinef1.html	
index.html		registration.html	
linlinef1.html		selcopr.html	
linlinef2.html		lt1.html	
wegistwation html		12 html	
looloopp html		1+2 html	
SCICOPT. NUMI			
LT. NUMI		IL A TUNE	
tZ.ntml		UCS.NUM1	
t3.html		lt6.html	
t4.html		t7.html	
t5.html		table_color3.html	
t6.html		table_frame.html	
t7.html		video.html	
table color3.html		zu1.html	
table frame html			
Shaneuar		Shaveway	
			Рору
$ \begin{array}{c} \bullet \\ \bullet $			
			0.0
Шевая 2 Іравая 3Сма	отр. <u>4</u> Редак. <mark>5</mark> Іечать	БСХРОКЬ РИСКАТЬ ВИСТ	гор Увидео 10Щерево

Рисунок 52 – Вид правой панели программы с отображением каталога сайта, размещенного на FTP-сервере

Замечание: при необходимости файлы сайта можно редактировать известными Вам приемами работы в **Far-manager**, так как будто Вы работаете с содержимым какого-либо диска на Вашем компьютере.

## 1.9.4 Вопросы для самоконтроля

- 1) Назовите этапы, составляющие процесс проектирования сайта.
- 2) Какие виды структур используются при создании сайтов?
- 3) В каких случаях можно использовать линейные структуры? Когда их использование нежелательно?
- 4) В чем преимущество древовидной структуры сайта? Поясните понятия глубины и ширины сайта.
- 5) Опишите алгоритм создания папки сайта.
- 6) Что такое Web-сервер7 Объясните суть использования протоколов http и ftp.
- 7) Опишите суть работы клиента с Web-сервером при использовании протокола http.
- 8) Поясните общую схему процесса размещения сайта на каком-либо сервере.
- 9) Объясните принцип взаимодействия клиента и Web-сервера с использованием протокола передачи данных ftp.
- 10) Какие программы используются для FTP-обмена? Приведите примеры.
- 11) Опишите алгоритм использования протокола ftp при размещении сайта.

# 2 Интегрированная среда разработки Visual Basic for Application

### 2.1 Общие сведения

Для создания программ с удобным и красивым интерфейсом пользователя фирма Microsoft в 1994 году разработала язык Visual Basic, имеющий возможности визуального программирования. С тех пор Visual Basic стал одним из самых популярных в мире языков программирования. Идею визуального программирования используют почти все разработчики языков программирования, например Borland Delphi, Cbuilder, и даже в 1С:Бухгалтерии можно делать свои модули и настройки примерно так же, как на Visual Basic.

Визуальное программирование означает, что разработчик создает интерфейс пользователя, помещая мышью нужные элементы управления – строки ввода, кнопки, списки выбора, картинки и другие элементы, привычные ему по работе в Windows, на форму (то есть, на окно), при этом интерфейс как бы «ри-После ЭТОГО пишутся процедуры на бейсике, которые суется». «привязываются» к некоторым событиям. Например, если «нарисовать» на форме кнопку с надписью «Выход из программы», и к событию нажатия на эту кнопку написать команду "End", то при нажатии на эту кнопку прекратится выполнение программы.

Позже Microsoft расширила возможности своего продукта и теперь он уже не только поддерживает технологию визуального программирования, но и ее развитие – технологию RAD, Rapidly Application Development, что в переводе означает «быстрая разработка приложений». RAD - это технология разработки программ, при которой программы составляются из уже готовых, отлаженных и проверенных блоков – компонент, после чего их взаимодействие и уникальная логика описывается на языке программирования.

Событийно-управляемое программирование – стиль, или технология, программирования, при котором выполнение блоков программы происходит в ответ на события, а не в заранее определенном порядке. Источником событий обычно является пользователь. То есть это набор откликов на события, происходящие с объектами. В основе визуального, событийно-управляемого, и RAD технологий программирования лежит объектно-ориентированное программирование.

**Объектно-ориентированное программирование** – программирование, основанное на моделировании предметной области при помощи объектов.

Каждый объект отражает определенную самостоятельную сущность предметной области со своими данными и своими возможностями. На него можно воздействовать вызовом принадлежащих объекту процедур (они называются *методами*) или установкой его параметров (они называются *свойствами*). Программа состоит из объектов, которые взаимодействуют друг с другом.

Класс есть некое множество объектов, имеющих общую структуру и общее поведение. Любой конкретный объект является экземпляром какого-либо

класса. Класс называют также типом объекта. Пример класса – командная кнопка. Конкретная кнопка на форме – экземпляр класса.

Визуальное, событийно-управляемое и объектно-ориентированное программирования обычно взаимосвязаны. Основными терминами в них являются «объект», «событие», «свойства».

Событие – это результат одного или нескольких внешних воздействий. К простейшим событиям можно отнести события связанные с мышью: щелчок, двойной щелчок, перетаскивание, нажатие на определенную кнопку и т.д. Создание приложения состоит из настройки свойств используемых объектов и создания обработчиков событий. Обработкой события является процедура, которая откликается на него. Смысл процедуры – выполнение запрограммированных действий: изменение свойств объекта, вызов методов, выполнение вычислений с выводом результата в определенные места приложения и т.д.

**Объекты** – это блоки или фрагменты, из которых собирается, проектируется приложение. Среда визуального программирования позволяет увидеть объекты на экране на этапе проектирования приложения, дает возможность настроить объекты, изменяя их расположение, размеры и другие свойства. Объекты располагаются на форме, которая в свою очередь тоже является объектом.

Свойства – это совокупность характеристик объекта. Объекты могут быть визуальными и невизуальными. Все визуальные объекты могут иметь такие свойства как цвет, шрифт, размеры и положение на экране. Невизуальные объекты становятся невидимыми во время запуска приложения и в определенный момент, после выполнения какого-то события могут стать визуальными, т.е. видимыми.

Каждый объект имеет:

- свойства
  - методы
  - события. Например,

Объект "телефон" 🐇



#### Свойства

цвет размер расположение вес, объем Методы снять трубку набрать номер передвинуть телефон События звонок длинный гудок короткий гудок

VBA (Visual Basic for Applications) – версия Visual Basic, которая встраивается в другие программы и позволяет расширять их функции. VBA встроен в пакет приложений Microsoft Office: Word, Excel, Access, Power Point, а также в почтовую программу Outlook Express. VBA – это среда программирования, предназначенная специально для создании макросов в перечисленных выше приложениях. Макрос – это список команд, которые запоминаются для повторного выполнения. Например, в Word можно написать программу на VBA, которая при нажатии на какую-нибудь кнопку сумму денег цифрами заменит автоматически на сумму словами, или создать в **Excel** свою собственную функцию, отсутствующую в стандартной библиотеке функций. Таким образом, VBA расширяет как вычислительные возможности офисных приложений, так и возможности пользователя при решении им конкретных задач.

# 2.2 Интерфейс окна VBA

# 2.2.1 Способы запуска среды VBA

Запустить VBA можно одним из нескольких способов.

*1 способ:* Нажать сочетание клавиш Alt+F11.

*2 способ:* Выполнить команду Сервис – Макрос – Редактор Visual Basic.

### *3 способ*:Подключить панель Visual Basic и нажать на пиктограмму **Ре**дактор Visual Basic.

После выполнения одного из вышеперечисленных способов откроется окно редактора **Редактор Visual Basic**, показанное на рисунке 53.



Строка заголовков, 2 - Горизонтальное меню, 3 - Панель инструментов Стандартная, 4 - Рабочая область, 5 - Контекстное меню вставки формы, модуля или классов модулей, 6 - Окно свойств объектов, 7 - Окно проекта Рисунок 53 – Вид окна VBA

# 2.2.2 Основные элементы окна VBA

*Горизонтальное меню* VBA состоит из пунктов, имеющих следующее назначение.

*File* - команды сохранения изменений в проекте и вывода на экран и печать исходного кода макросов.

*Edit* - команды управления исходным кодом в окне **Code**, а также объектами в формах.

*View* - команды, позволяющие выводить или убирать с экрана различные окна самого редактора VBA.

*Insert* - команды вставки в проект различных объектов: процедур, модулей, форм, классов и пр.

*Format* – команды, используемые при создании пользовательских диалоговых окон. Они позволяют выравнивать объекты в форме по отношению друг к другу, настраивать размеры и внешний вид элементов управления и другие операции.

*Debug* - команды тестирования и отладки кода. Позволяют запускать код с любой точки, отслеживать ход выполнения по шагам, видеть значения, прерывать программу в нужном месте.

*Run* - команды запуска программного кода на выполнение, прерывания, возобновление работы, а также возврата прерванной программы в исходное состояние.

*Tools* - команды, позволяющие выбрать макрос для выполнения или получения доступа к внешним библиотекам макросов. Доступ к диалоговому окну **Option** (параметры) редактора и окну свойств VBA.

*Add-Ins* - одна команда **Add-in Manager** для вывода диалогового окна, в котором можно загружать, выгружать, регистрировать или определять поведение программ - дополнений (надстроек).

*Панель инструментов* Стандартная дублирует основные команды горизонтально меню.

**Окно проектов** отображает иерархическую структуру файлов, форм и модулей текущего проекта (рисунок 54). Программы на VBA хранятся в проектах. *Проект* содержит модули различных типов, а модули включают различные процедуры. Проект может содержать несколько модулей. Имеется три типа модулей:

- 1) стандартные модули это модули, в которых можно описать доступные во всем проекте процедуры;
- модули класса содержат описание объекта, который является членом класса. Процедуры, написанные в модуле класса, используются только в этом модуле.

3) модули форм и отчетов, которые связаны с конкретной формой или отчетом. Модули форм и отчетов часто содержит процедуры обработки событий, которые срабатывают в ответ на событие в форме или отчете. Процедуры обработки событий используются для управления поведением форм и отчетов и их реакцией на действия пользователя типа щелчка мыши на кнопке.



Рисунок 54 – Окно проектов

### Окно модулей

Модули содержат описания и процедуры - наборы описаний и инструкций, сгруппированных для выполнения. Существует три типа процедур:

- 1) процедура **Sub** набор команд, с помощью которого можно решить определенную задачу. При ее запуске выполняются команды процедуры, а затем управление передается в приложение пакета MS Office или процедуру, которая вызвала данную процедуру.
- процедура Function (функция) также представляет собой набор команд, который решает определенную задачу. Различие заключается в том, что такие процедуры обязательно возвращают значение, тип которого можно описать при создании функции.
- 3) процедура **Property** используется для ссылки на свойство объекта. Данный тип процедур применяется для установки или получения значения пользовательских свойств форм и модулей.

Процедуры модуля записываются в окне модулей. Окно модулей располагается в рабочей области и может быть добавлено через контекстное меню объектов (рисунок 55). В верхней части окна имеются два списка. Первый спи-

сок облегчает выбор управляющего объекта, а второй – событий, совершаемых над объектом.

4	🦑 Книга3.xls - Module2 (Code)			:	
(6	eneral) 🔽		maximum	•	
	Function maximum(a, b) If a > b Then maximum = a Else maximum = b End Function		(Declarations) maximum		
					1

Рисунок 55 – Окно модулей или класса модулей

Написание программ существенно облегчается за счет способности редактора кода автоматически завершать написание операторов, свойств и параметров. При вводе кода редактор сам предлагает пользователю список компонентов, логически завершающих вводимую пользователем инструкцию.

Например, при наборе кода **Range** ("A1") после ввода точки на экране отобразится список элементов (рисунок 56), которые логически завершают данную инструкцию. Двойной щелчок на выбранном элементе из этого списка или нажатие клавиши <Tab> вставляет выбранное имя в код программы.



Рисунок 56 - Список элементов

Автоматическое отображение списка компонентов происходит, если установлен флажок Auto List Members вкладки Editor диалогового окна Options, отображаемого на экране выбором команды Tools—Options.

К интеллектуальным ресурсам редактора кода относится также автоматическое отображение на экране справочных сведений о процедурах, функциях, свойствах и методах после набора их имени, показанным на рисунке 57.

(	General)  Worksheet_Activate
	<pre>Private Sub Worksheet_Activate() Range("A1").Value = 1 MsgBox</pre>
	MsgBox( <b>Prompt</b> , [Buttons As VbMsgBoxStyle = vbOKOnly], [ <i>Title</i> ], [ <i>HelpFile</i> ], [ <i>Context</i> ]) As VbMsgBoxResult ]

Рисунок 57 – Справочные сведения о вводимой процедуре

Автоматическое отображение на экране сведений о процедурах, функциях, свойствах и методах после ввода их имени происходит только при установленном флажке Auto Data Tips вкладки Editor диалогового окна Options, отображаемого на экране выбором команды Tools—Options.

В процессе ввода исходного кода VBA автоматически отслеживает опечатки. Всякий раз, когда нажимается клавиша **<Enter>**, VBA просматривает введенную строку, переводит в верхний регистр начальные буквы зарезервированных слов, а остальные их буквы – в нижний регистр (если, все было введено корректно). Кроме того, зарезервированные слова редактор VBA отображает синим цветом, добавляет пробелы вокруг знаков арифметических операций, а также операций сравнения. Если после нажатия клавиши **<Enter>** VBA ничего не отредактировал, то, скорее всего, в этой строке осталось что-то не распознано, и поэтому придется искать ошибку самостоятельно.

Редактор кода также производит автоматическую проверку синтаксиса набранной строки кода сразу после нажатия клавиши **<Enter>**. Если после набора и нажатия клавиши **<Enter>** строка выделяется красным цветом, то это указывает на наличие синтаксической ошибки во введенной строке. Эту ошибку необходимо найти и исправить. Кроме того, если установлен флажок Auto Syntax Check вкладки Editor диалогового окна Options, помимо выделения красным цветом фрагмента кода с синтаксической ошибкой, на экране отображается диалоговое окно, поясняющее, какая именно ошибка возникла.

#### Окно форм

Основным инструментом визуального программирования является форма. VBA представляет разработчику проектов не только возможность использования уже существующих встроенных диалоговых окон, но и средства для разработки своих собственных форм. Пользовательские окна обеспечивают уникальный интерфейс, наилучшим образом приспособленный для решения конкретных задач, стоящих перед пользователем.

По своей сути форма (или пользовательская форма) представляет собой диалоговое окно, в котором можно размещать элементы управления (меню, кнопки, окна списков, полосы прокрутки и другие элементы). Формы позволя-

ют реализовать интерфейс, к которому пользователь привык, работая с программными продуктами операционной системы Windows. С точки зрения VBA форма представляет собой объект UserForm со стандартной сеткой (группа регулярно расположенных точек). Сетка служит для удобства размещения элементов пользовательского интерфейса. Шаг сетки можно изменить, выполнив команду Tools—Options (Сервис—Параметры) (вкладка General (Общие)). Форма является объектом и имеет свои свойства, методы и события.

Окно форм располагается в рабочей области и может быть добавлено через контекстное меню объектов (рисунок 58). Обычно вместе с окном форм подключается панель инструментов **Toolbox**, содержащая пиктограммы для помещения объектов управления на форму. Объекты управления – это списки, кнопки, флажки, радиокнопки и т.д. Описание некоторых пиктограмм данной панели приводится в таблице 9.



Рисунок 58 – Окно формы и панель объектов управления Toolbox

Таблица 9 - Назначение некоторых пиктограмм панели инструментов **Toolbox** 

Номер	Название	Назначение элемента	
	JJICMEIITA		
ЛИ			
1	2	3	
1	Указатель	изображен в левом верхнем углу панели. Назначение	
	(Pointer)	указателя - дезактивировать все кнопки панели.	
2	Label	графический элемент управления, применяемый для	
		показа текста, который пользователь не может изме-	
		нить непосредственно	

Прод	цолжение табл	ицы 9	
1	2	3	
3	TextBox	TextBox (окно с текстом), иногда называемый полем ре-	
		дактирования или элементом управления редактирова-	
		нием, показывает данные, введенные пользователем во	
		время работы, или присвоенные элементу управления из	
		программы или введенные во время «рисования» фор-	
		мы.	
4	ComboBox	сочетает возможности элементов управления TextBox и	
		ListBox. Пользователи могут вводить информацию в об-	
		ласть текста или выбирать пункты из окна со списком.	
5	ListBox	показывает список пунктов, из которого пользователь	
		может выбрать один или несколько пунктов. Если число	
		пунктов больше, чем можно показать в окне, то к окну	
		ListBox автоматически добавляется полоса прокрутки.	
6	CheckBox	Элемент управления CheckBox (флажок проверки) по-	
		казывает метку "√", если он выбран. Метка "√" исче-	
		зает, когда флажок очищается. Он может быть включен	
		(галочка видна) и выключен (не видна). Можно исполь-	
		зовать группы флажков проверки для показа множе-	
		ственного выбора, в котором пользователь может вы-	
		брать один или более флажков. Может группироваться	
		внутри элемента управления Frame	
7	OptionButt	изображает переключатель, который может быть вклю-	
	on	чен или выключен. Обычно группируется в объекте	
		Frame	
		Отличается от предыдущего тем, что может быть вы-	
		бран только один из группы.	
9	Frame	обеспечивает группировку элементов управления по	
		признакам	
10	Command	командная кнопка, ее нажатием пользователь дает ка-	
	Button	кую-либо команду программе. Часто встречаются кноп-	
		ки с надписями «ОК», «Отмена»	
13	ScrollBar	Полосы прокрутки могут дать аналоговое представле-	
		ние о текущем положении в длинном списке или	
		большой порции данных. Можно пользоваться полоса-	
		ми прокрутки в качестве устройств ввода, индикаторов	
		скорости или количества. Например, чтобы управлять	
		громкостью звука в компьютерной игре или наблюдать	
		за истекшим временем в хронометрируемом процессе.	
15	Image	используется для показа графических изображений.	
		Позволяет увеличивать/уменьшать картинки.	

Окно свойств содержит перечень тех свойств объекта, которые пользователь может изменить. Окно выполнено в виде таблицы, каждая строка которой содержит два поля. В левом поле записано название свойства, в правом значение свойства (рисунок 59). Значения свойств можно изменять непосредственно в окне **Properties**. Щелчок мышью по правому полю окна может привести к появлению в нем кнопки, от ее появления и вида зависит способ изменения свойств объекта:

- 1) если кнопка не появилась, то в поле можно ввести произвольную запись. Свойство будет изменено, если новое значение допустимо,
- если после объявленного значения в правом поле появилась кнопка с треугольником, то щелчок по этой кнопке вызовет перечень возможных значений свойства, предоставляя возможность выбрать одно из них,
- если после объявленного значения в правом поле появилась кнопка с многоточием, то щелчок по этой кнопке вызовет стандартное диалоговое окно Windows, позволяющее выбрать допустимое значение свойства.

Кроме визуальных способов изменения свойств, всегда имеется менее наглядная, но иногда необходимая возможность программного присвоения значений свойств объектов в коде (тексте) процедур Visual Basic.

Properties - Form1 🛛 🛛		
Form1 Form 💽		
Alphabetic	Categorized	
Caption	Form1	
ClipControls	True	
ControlBox	True	
DrawMode	13 - Copy F	
DrawStyle	0 - Solid	
DrawWidth	1	
Enabled	True	
FillColor	8H0000	
FillStyle	1 - Transpa	
Font	MS Sans Se	
FontTranspa	True	
ForeColor	8H8000	
HasDC	True	
Height	3600	
HelpContext10		
Caption Returns/sets the text		

Рисунок 59 - Окно свойств объекта Form1

### 2.2.3 Вопросы для самоконтроля

- 1) Дайте определение визуальному программированию.
- 2) Дайте определение событийно-управляемому программированию.
- 3) Дайте определение объектно-ориентированному программированию.
- 4) Назовите среды программирования, поддерживающие визуальное программирование, событийно-управляемое программирование и объектно-ориентированное программирование.
- 5) Что такое объект? Свойства объекта? Каковы методы объекта?
- 6) Дайте определение классу объектов, приведите примеры класса.
- 7) Дайте определение понятию «событие». Какие события в среде Windows вы знаете?
- 8) В чем назначение языка VBA?
- 9) Перечислите основные части интерфейса окна VBA.
- 10) Назовите основные пункты горизонтального меню и расскажите, какие типы команд собраны в каждом пункте.
- 11) Какая информация располагается в окне проектов?
- 12) Для чего предназначено окно модулей?
- 13) Что такое форма? Как ее создать?
- 14) Перечислите объекты управления, которые можно расположить на форме? Где находятся пиктограммы этих объектов?
- 15) Опишите назначение объектов управления панели инструментов **ToolBox**.
- 16) Для чего необходимо окно свойств? Как его активизировать?
- 17) В чем заключаются интеллектуальные способности языка VBA?

# 2.3 Понятие алгоритма

#### 2.3.1 Алгоритм. Способы записи алгоритмов

**Алгоритм** - это описание последовательности действий для решения задачи или достижения поставленной цели; правила выполнения основных операций обработки данных; описание вычислений по математическим формулам.

Перед началом разработки алгоритма необходимо четко уяснить задачу: что требуется получить в качестве результата, какие исходные данные необходимы и какие имеются в наличии, какие существуют ограничения на эти данные. Далее требуется записать, какие действия необходимо предпринять для получения из исходных данных требуемого результата.

На практике наиболее распространены следующие формы представления алгоритмов:

- 1) словесная (записи на естественном языке);
- 2) графическая (изображения из графических символов);

3) программная (тексты на языках программирования).

*Словесный* способ записи алгоритмов представляет собой описание последовательных этапов обработки данных. Алгоритм задается в произвольном изложении на естественном языке в виде слов и предложений.

*Пример*. Записать алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел.

Алгоритм может быть следующим:

- 1) задать два числа;
- если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
- 3) определить большее из чисел;
- 4) заменить большее из чисел разностью большего и меньшего из чисел;
- 5) повторить алгоритм с шага 2.

Описанный алгоритм применим к любым натуральным числам и должен приводить к решению поставленной задачи. Убедитесь в этом самостоятельно, определив с помощью этого алгоритма наибольший общий делитель чисел 125 и 75.

Словесный способ не имеет широкого распространения по следующим причинам:

- такие описания строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

*Графический* способ представления алгоритмов является более компактным и наглядным по сравнению со словесным.

При графическом представлении алгоритм изображается в виде последовательности связанных между собой геометрических фигур, каждая из которых называется блоком и соответствует выполнению одного или нескольких действий. Такое графическое представление называется схемой алгоритма или блок-схемой.

В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями или стрелками, определяющими очередность выполнения действий.

Приведем наиболее часто употребляемые символы, согласно ГОСТ 19.701-90 ЕСПД "Схемы алгоритмов, программ, данных и систем" в таблице 10.

Название символа	Обозначение и	Пояснение
	пример заполне-	
	ния	
1	2	3
Решение	da a <b rem<="" td=""><td>Проверка условий</td></b>	Проверка условий
Границы цикла	i = 1	Внутри блока реализуется команда- цикл
	i = n	Внутри блока реализуется команда- цикл
Процесс	a=a+2	Вычисления по подпрограмме, стан- дартной подпрограмме
Ввод-вывод	Ввод а,b,с	Ввод-вывод в общем виде
Пуск-останов	Начапо	Начало, конец алгоритма, вход и вы- ход в подпрограмму
Документ	Печать а,с	Вывод результатов на печать

Таблица 10 – Наиболее часто употребляемые в блок-схемах алгоритмов символы

Блок "решение" используется для обозначения переходов управления по условию. В каждом блоке "решение" должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок "граница цикла" используется для организации циклических конструкций. Состоит из двух частей, обе части должны содержать один и тот же идентификатор. Условия для инициализации, приращения, завершения и т.д. помещаются внутри символа вначале или в конце, в зависимости от расположения операции, проверяющей условие.

Блок "процесс" используется для отображения функции обработки данных любого вида: выполнение определенной операции или группы, приводящее к изменению значения, формы или размещения информации или к определению, по которому из нескольких направлений потока следует двигаться.

# 2.3.2 Структуры алгоритмов

Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных базовых (т.е. основных) элементов. Естественно, что при таком подходе к алгоритмам изучение основных принципов их конструирования должно начинаться с изучения этих базовых элементов Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур: следование, ветвление, цикл.

Характерной особенностью базовых структур является наличие в них одного входа и одного выхода.

1) Базовая структура следование (линейный алгоритм). Образуется из последовательности действий, следующих одно за другим:



2) Базовая структура ветвление (условный алгоритм). Обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Структура ветвление существует в четырех основных вариантах:

- если-то;
  - если-то-иначе;
- выбор;
- выбор-иначе.

 если-то если условие то действия конец если



2) если-то-иначе если условие то действия 1 иначе действия 2 конец если



3) выбор
выбор
при условие 1: действия 1
при условие 2: действия 2

при условие N: действия N конец выбора



4) выбор-иначе

выбор

при условие 1: действия 1 при условие 2: действия 2

. . . . . . . . . . . .

при условие N: действия N иначе действия N+1 конец выбора

> усполне 1 усполне 1 усполне 9 усполне 9

#### Пример 1. Составить блок-схему алгоритма вычисления функции





3 Базовая структура цикл (циклический алгоритм). Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла.

Структура цикл существует в трех основных вариантах:

Цикл типа для.

Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.

Цикл типа пока.

Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова пока.

Цикл типа делать - пока.

Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова пока. Условие проверяется после выполнения тела цикла.

Заметим, что циклы для и пока называют также циклами с предпроверкой условия а циклы делать - пока - циклами с постпроверкой условия. Иными словами, тела циклов для и пока могут не выполниться ни разу, если условие окончания цикла изначально не верно. Тело цикла делать - пока выполнится как минимум один раз, даже если условие окончания цикла изначально не верно.

цикл для і от і1 до і2 шаг і3

тело цикла (последовательность действий) конец цикла



цикл пока условие

тело цикла (последовательность действий) конец цикла



цикл делать

тело цикла (последовательность действий) пока условие

конец цикла



**Пример 2.** Составить блок-схему алгоритма вычисления функции yk = sin (kx) + cos (k/x), k = 1, 2, ..., 50



Итерационные циклы. Особенностью итерационного цикла является то, что число повторений операторов тела цикла заранее неизвестно. Для его организации используется цикл типа пока. Выход из итерационного цикла осуществляется в случае выполнения заданного условия.

На каждом шаге вычислений происходит последовательное приближение и проверка условия достижения искомого результата.

Пример 3. Составить алгоритм вычисления суммы ряда

S = x - 
$$\frac{x^2}{2} + \frac{x^2}{3} - \frac{x^4}{4} + \dots (-1)^{i-1} \frac{x^i}{i} + \dots$$

с заданной точностью (для данного знакочередующегося степенного ряда требуемая точность будет достигнута, когда очередное слагаемое станет по абсолютной величине меньше).

Вычисление сумм - типичная циклическая задача. Особенностью же нашей конкретной задачи является то, что число слагаемых (а, следовательно, и число повторений тела цикла) заранее неизвестно. Поэтому выполнение цикла должно завершиться в момент достижения требуемой точности.

При составлении алгоритма нужно учесть, что знаки слагаемых чередуются и степень числа х в числителях слагаемых возрастает.

Решая эту задачу "в лоб" путем вычисления на каждом i-ом шаге частичной суммы

S:=S+(-1)\*\*(i-1)\*x\*\*i/i,

мы получим очень неэффективный алгоритм, требующий выполнения большого числа операций. Гораздо лучше организовать вычисления следующим образом: если обозначить числитель какого-либо слагаемого буквой р, то у следующего слагаемого числитель будет равен -p\*x (знак минус обеспечивает чередование знаков слагаемых), а само слагаемое m будет равно p/i, где i - номер слагаемого.



Алгоритм, в состав которого входит итерационный цикл, называется итерационным алгоритмом. Итерационные алгоритмы используются при реализации итерационных численных методов. В итерационных алгоритмах необходимо обеспечить обязательное достижение условия выхода из цикла (сходимость итерационного процесса). В противном случае произойдет зацикливание алгоритма, т.е. не будет выполняться основное свойство алгоритма - результативность.

Вложенные циклы. Возможны случаи, когда внутри тела цикла необходимо повторять некоторую последовательность операторов, то есть организовать внутренний цикл. Такая структура получила название цикла в цикле или вложенных циклов. Глубина вложения циклов (то есть количество вложенных друг в друга циклов) может быть различной.

При использовании такой структуры для экономии машинного времени необходимо выносить из внутреннего цикла во внешний все операторы, которые не зависят от параметра внутреннего цикла.

## 2.3.3 Программный способ записи алгоритмов

Алгоритм, предназначенный для исполнения на компьютере, должен быть записан на "понятном" ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного их толкования. Следовательно, язык для записи алгоритмов должен быть формализован. Такой язык принято называть языком программирования, а запись алгоритма на этом языке - программой для компьютера.

В настоящее время в мире существует несколько сотен реально используемых языков программирования. Для каждого есть своя область применения.

Любой алгоритм есть последовательность предписаний, выполнив которые можно за конечное число шагов перейти от исходных данных к результату. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования - чем меньше детализация, тем выше уровень языка.

По этому критерию можно выделить следующие уровни языков программирования:

- 1) машинные;
- 2) машинно-ориентированные (ассемблеры);
- 3) машинно-независимые (языки высокого уровня).

Машинные языки и машинно-ориентированные языки - это языки низкого уровня, требующие указания мелких деталей процесса обработки данных.

Языки же высокого уровня имитируют естественные языки, используя некоторые слова разговорного языка и общепринятые математические символы. Эти языки более удобны для человека.

Языки высокого уровня делятся на:

- 1) алгоритмические (Basic, Pascal, С и др.), которые предназначены для однозначного описания алгоритмов;
- 2) логические (Prolog, Lisp и др.), которые ориентированы не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи с тем, чтобы решение следовало из составленного описания.
- 3) объектно-ориентированные (Object Pascal, C++, Java и др.), в основе которых лежит понятие объекта, сочетающего в себе данные и действия над нами. Программа на объектно-ориентированном языке, решая некоторую задачу, по сути описывает часть мира, относящуюся к этой задаче. Описание действительности в форме системы взаимодействующих объектов естественнее, чем в форме взаимодействующих процедур.

Алгоритмический язык (как и любой другой язык) образуют три его составляющие: алфавит, синтаксис и семантика.

Алфавит - это фиксированный для данного языка набор основных символов, т.е. "букв алфавита", из которых должен состоять любой текст на этом языке - никакие другие символы в тексте не допускаются. Из символов алфавита формируются лексемы языка:

- константы;
- идентификаторы;
- знаки операций;
- ключевые (служебные, иначе зарезервированные) слова;

- разделители (знаки пунктуации).

Синтаксис - это правила построения фраз, позволяющие определить, правильно или неправильно написана та или иная фраза. Точнее говоря, синтаксис языка представляет собой набор правил, устанавливающих, какие комбинации символов являются осмысленными предложениями на этом языке.

Семантика определяет смысловое значение предложений языка. Являясь системой правил истолкования отдельных языковых конструкций, семантика устанавливает, какие последовательности действий описываются теми или иными фразами языка и, в конечном итоге, какой алгоритм определен данным текстом на алгоритмическом языке.

Каждое понятие алгоритмического языка подразумевает некоторую синтаксическую единицу (конструкцию) и определяемые ею свойства программных объектов или процесса обработки данных.

Понятие языка определяется во взаимодействии синтаксических и семантических правил. Синтаксические правила показывают, как образуется данное понятие из других понятий и букв алфавита, а семантические правила определяют свойства данного понятия

Основными понятиями в алгоритмических языках обычно являются данные, имена, операции и выражения, операторы.

#### 2.3.4 Вопросы для самоконтроля

- 1) Дайте определение алгоритма.
- 2) Какие существуют способы записи алгоритмов?
- 3) Приведите примеры алгоритмов, описанных словесно.
- 4) Опишите назначение основных символов блок-схемы.
- 5) Запишите алгоритм нахождения максимального из трех целых чисел при помощи блок-схемы.
- 6) Приведите примеры задач, реализующих в своем решении линейный алгоритм.
- 7) В чем суть алгоритма ветвления? Приведите примеры таких алгоритмов.
- 8) Какие варианты ветвлений существуют?
- 9) Дайте определение структуре алгоритмов «цикл». Приведите примеры циклических алгоритмов.
- 10) Какие варианты алгоритмов структуры «цикл» вы знаете?
- 11) Дайте определение и приведите примеры итерационных циклов.
- 12) Дайте определение вложенных циклов.
- 13) Какие уровни языков программирования существуют?
- 14) На какие виды делятся языки высокого уровня? Приведите примеры.
- 15) Какие части включает в себя любой язык?
- 16) Из чего состоит алфавит любого языка программирования?
- 17) Что такое семантика языка?
- 18) Что такое синтаксис языка?

# 2.4 Описание алгоритмов на языке VBA

# 2.4.1 Представление данных в памяти

**Данные** - величины, обрабатываемые программой. Имеется три основных вида данных: константы, переменные и массивы.

**Константы** - это данные, которые зафиксированы в тексте программы и не изменяются в процессе ее выполнения. Константы представляются в виде лексем, изображающих фиксированные числовые, логические, символьные или строковые значения.

Числовые константы могут быть целыми, вещественными (с фиксированной или плавающей точкой) и перечислимыми.

Целые константы могут быть десятичными, восьмеричными и шестнадцатиричными. Десятичная целая константа определена как последовательность десятичных чисел, начинающаяся не с нуля, если это не число нуль. Восьмеричные константы в Visual Basic for Application начинаются с префикса &O и содержат числа от 0 до 7. Шестнадцатиричные числа начинаются с префикса &H и содержат числа от 0 до 9 и латинские буквы от А до F.

Примеры констант: 123, &O247, &H1F.

Вещественные константы записываются в десятичной системе счисления и в общем случае содержат целую часть (десятичная целая константа), десятичную точку, дробную часть (десятичная целая константа), признак (символ) экспоненты Е и показатель десятичной степени (десятичная целая константа, возможно со знаком).

Примеры констант: 123.456, 3.402823Е38.

**Перечислимые** константы - это набор обычных целочисленных констант. Перечисляемый набор может содержать конечный набор уникальных целых значений, каждое из которых имеет особый смысл в текущем контексте. Перечисляемые наборы являются удобным инструментом, обеспечивающим выбор из ограниченного набора параметров. Например, если пользователь должен выбрать цвет из списка, то можно установить соответствие: черный = 0, белый = 1 и т.д.

**Логические** (булевы) константы могут иметь лишь одно из двух значений: да (истина, TRUE), нет (ложь, FALSE).

Символьные и строковые константы. В отличие от большинства языков программирования, где существуют отдельно символьные (содержащие один символ алфавита) и строковые (массив символов) константы, в VBA существуют только строковые, имеющие два типа значений:

 Строки переменной длины, которые могут содержать до приблизительно 2 миллиардов (2<sup>3</sup>1) символов.  Строки постоянной длины, которые могут содержать от 1 до приблизительно 64К (2<sup>1</sup>6) символов.

Примеры строковых констант: "abcde", "информатика", "" (пустая стро-

В зависимости от значения константы по разному представляются в памяти ПК. Целые представляют последовательный набор фиксированного количества байтов, а вещественные, даже не отличаясь от целых по значению, имеют другую форму внутреннего представления, обусловленную применением арифметики с плавающей точкой при операциях с такими константами и представлением вещественного числа. В общем случае форма представления данных в памяти ПК определяется типом данных.

# 2.4.2 Типы данных

ка).

Тип данных определяет, каким образом биты данных, представляющие конкретное значение, хранятся в памяти ПК. В каждом языке программирования имеется свой фиксированный набор базовых типов данных. Некоторые языки позволяют создание дополнительных (пользовательских) типов данных. В VBA имеются следующие типы данных:

Тип данных	Размер в ра-	Диапазон значений
	мяти	
1	2	3
Byte (байт)	1 байт	От 0 до 255.
Boolean (логиче-	2 байт	True или False.
ский)		
Integer (целое)	2 байт	От -32 768 до 32 767.
Long (длинное це-	4 байт	От -2 147 483 648 до 2 147 483 647.
лое)		
Single (с плавающей	4 байт	От -3,402823Е38 до -1,401298Е-45 для
точкой обычной		отрицательных значений; от
точности)		1,401298Е-45 до 3,402823Е38 для поло-
		жительных значений.
Double (с плаваю-	8 байт	От -1,79769313486232Е308 до
щей точкой двойной		-4,94065645841247Е-324 для отрица-
точности)		тельных значений; от
		4,94065645841247Е-324 до
		1,79769313486232E308 для положи-
		тельных значений.
Currency (денеж-	8 байт	От -922 337 203 685 477,5808 до 922
ный)		337 203 685 477,5807.

Таблица 11 – Типы данных VBA

Продолжение таблицы 1	1
-----------------------	---

1	2	3
Decimal (масштаби-	14 байт	+/-79 228 162 514 264 337 593 543 950
руемое целое)		335 без дробной части;
		+/-7,9228162514264337593543950335 c
		28 знаками справа от запятой; мини-
		мальное ненулевое значение имеет вид
		+/-0,00000000000000000000000000000000000
Date (даты и время)	8 байт	От 1 января 100 г. до 31 декабря 9999 г.
Object (объект)	4 байт	Любой указатель объекта.
String (строка пере-	10 байт +	От 0 до приблизительно 2 миллиардов.
менной длины)	длина строки	
String (строка посто-	Длина строки	От 1 до приблизительно 65 400.
янной длины)		
Variant (числовые	16 байт	Любое числовое значение вплоть до
подтипы)		границ диапазона для типа Double.
Variant (строковые	22 байт +	Как для строки (String) переменной
подтипы)	длина строки	длины.
Тип данных, опреде-	Объем опре-	Диапазон каждого элемента определя-
ляемый пользова-	деляется эле-	ется его типом данных.
телем (с помощью	ментами	
ключевого слова		
Type)		

Поясним некоторые характерные для VBA типы данных.

**Byte** - Массивы данного типа служит для хранения двоичных данных, например, изображений. Использование данного типа предохраняет двоичные данные во время преобразования формата.

**Boolean** - для хранения логических (булевых) значений. По умолчанию значением булевской переменной является False- ложь.

**Currency** - для хранения чисел с дробной частью до четырех цифр и целой частью до 15 цифр, то есть данных с фиксированной десятичной точкой, удобных для денежных вычислений. Числа с плавающей десятичной точкой (Single, Double) имеют больший диапазон значений, но могут приводить к ошибкам округления.

**Decimal** - в версии 5.0 поддерживается использование типа данных Decimal только в пределах типа Variant, т.е. невозможно описать переменную с типом Decimal. Пользователь, однако, имеет возможность создать переменную типа Variant с подтипом Decimal с помощью функции CDec.

**Object** - поскольку VBA является объектно-ориентированным языком, в нем можно манипулировать различными объектами, адреса расположения которых в памяти (указатели) имеют этот тип.

String - по умолчанию данные строкового типа имеют переменную длину и могут удлиняться или укорачиваться. Однако такие строки занимают на 10 байт памяти больше, поэтому можно объявить строки фиксированной длины, явно указав количество символов. Если количество символов будет меньше объявленного, то свободные места заполняются пробелами, при попытке занесения большего количества символов лишние отбрасываются.

Variant - может быть использован для хранения данных всех базовых типов без выполнения преобразования (приведения) типов. Применение данного типа позволяет выполнять операции, не обращая внимание на тип данных, которые они содержат. Удобен для объявления переменных, тип которых заранее неизвестен. Переменные этого типа могут содержать специальные значения: Empty, Null, Error.

## 2.4.3 Идентификаторы, переменные

Имена (идентификаторы) - употребляются для обозначения объектов программы (переменных, массивов, процедур и др.). В VBA имена констант, переменных и процедур должны удовлетворять следующим требованиям:

- 1) должны начинаться с буквы;
- 2) не могут содержать точки и символов объявления типа;
- 3) не могут быть длиннее 255 символов. Длина имен объектов не должна превышать 40 символов.
- 4) не могут быть ключевыми словами (именами операций, операторов, встроенных функций).

**Переменные** представляют собой зарезервированное место в памяти ПК для хранения значения. Переменные обозначаются именами - словами, используемыми для ссылки на значение, которое содержит переменная, и характеризуются типом, определяющим вид данных, которые можно хранить в переменной. Переменные могут изменять свои значения в ходе выполнения программы. По умолчанию переменные имеют тип данных Variant. Для явного указания типа переменной можно в конце ее имени указать символ описания типа:

Currency - @; Double - #; Integer - %; Long - &; Single - !; String - \$. или объявлен

или объявления переменных. Объявить переменную - значит заранее сообщить программе о ее существовании. Объявление переменной производится специальным оператором. Одновременно с объявлением переменной после ее имени можно записать ключевое слово As, после которого задается тип переменной.

Константы - имена, идентифицирующие некоторые неизменяемые числовые значения или строки текста. В отличие от переменных их нельзя изменить или назначить им новые значения. В VBA различают внутренние или системные константы, которые имеют префикс vb, например vbOK, и символические или определяемые пользователем с помощью ключевого слова Const
константы. Имена констант имеют те же ограничения, что и имена переменных, тип данных, хранящихся в константе, задается так же, как для переменных.

## 2.4.4 Операции, выражения, операторы, функции

В VBA существуют следующие типы операций:

- арифметические операции, используемые для выполнения математических вычислений: ^, \*, /, \, Mod, +, -. Здесь \ - Возвращает результат целого деления двух чисел, Mod - Возвращает остаток при целом делении двух чисел (значение по модулю).
- 2) операции сравнения, используемые для выполнения операций сравнения
- < , > , <=, >= , = , <> ;
- 3) логические операции, используемые для выполнения логических операций

And - Возвращает результат конъюнкции (логического И) для двух выражений с операциями сравнения, либо выполняет поразрядное сравнение двух числовых выражений

Eqv - Используется для проверки логической эквивалентности двух выражений с операциями сравнения, либо выполняет поразрядное сравнение двух числовых выражений

Imp - Выполняет операцию логической импликации для двух выражений с операциями сравнения, либо выполняет поразрядное сравнение двух числовых выражений

Not - Выполняет над выражением операцию логического отрицания, а также поразрядное изменение значений каждого разряда переменной

Or - Выполняет операцию логического ИЛИ (сложения) для двух выражений

Xor - Выполняет операцию исключающего ИЛИ для двух выражений

4) операция конкатенации символьных значений друг с другом с образованием одной длинной строки - & - используется для слияния двух строковых выражений.

**Выражения** - предназначаются для выполнения необходимых вычислений, состоят из констант, переменных, функций (например, exp(x)), объединенных знаками операций.

Выражения записываются в виде линейных последовательностей символов (без подстрочных и надстрочных символов, "многоэтажных" дробей и т.д.), что позволяет вводить их в компьютер, последовательно нажимая на соответствующие клавиши клавиатуры.

Различают выражения арифметические, логические и строковые.

 Арифметические выражения служат для определения одного числового значения. Например, (1+sin(x))/2. Значение этого выражения при x=0 равно 0.5, а при x=p/2 - единице.

- 2) Логические выражения описывают некоторые условия, которые могут удовлетворяться или не удовлетворяться. Таким образом, логическое выражение может принимать только два значения - "истина" или "ложь" (да или нет). Рассмотрим в качестве примера логическое выражение x\*x + y\*y < r\*r, определяющее принадлежность точки с координатами (x,y) внутренней области круга радиусом r с центром в начале координат. При x=1, y=1, r=2 значение этого выражения - "истина", а при x=2, y=2, r=1 - "ложь".
- 3) Значения строковых выражений тексты. В них могут входить литерные константы, литерные переменные и литерные функции, разделенные знаком операции сцепки. Например, А & В означает присоединение строки В к концу строки А. Если А = "куст ", а В = "зеленый", то значение выражения А&В есть "куст зеленый".

Оператор - это наиболее крупное и содержательное понятие языка: каждый оператор представляет собой законченную фразу языка и определяет некоторый вполне законченный этап обработки данных. В состав операторов входят:

1) ключевые слова;

2) данные;

3) выражения и т.д.

Операторы подразделяются на исполняемые и неисполняемые. Неисполняемые операторы предназначены для описания данных и структуры программы, а исполняемые - для выполнения различных действий (например, оператор присваивания, функции ввода и вывода информации, условный оператор, операторы цикла, оператор процедуры и др.).

Оператор присваивания. Для того чтобы присвоить переменной нужное значение, используется операция присваивания. Операция присваивания всегда включает знак равенства «=».

Операция присваивания предписывает выполнить выражение, заданное в его правой части, и присвоить результат имени переменной, имя которой указано в левой части.

Например,

a = 3.5

a = a + 3

После выполнения данного фрагмента программы значение переменной а будет равно 6.5.

**Вывод данных.** Для отображения значений промежуточных или итоговых переменных необходимо использовать функцию **MsgBox** или метод **Range**.

Синтаксис функции: MsgBox сообщение, [, значение][, заголовок]

Аргументы:

сообщение – строковое выражение, отображаемое как сообщение в диалоговом окне.

значение – числовое выражение, представляющее сумму значений, которые указывают число и тип отображаемых кнопок, тип используемого значка, основную кнопку. Значение этого аргумента по умолчанию равняется 0.

заголовок – строковое выражение, отображающееся в заголовке окна вывода

Эта функция отображает диалоговое окно, содержащее сообщение длиной до 1024 символов, в которое с помощью операции конкатенации можно включить значение переменных, а также (необязательно) кнопки для реакции на отображения окна (по умолчанию только кнопка ОК) и заголовок окна (строковое выражение).

Значения аргумента, определяющие отображаемые кнопки

Константа	Значе- ние	Отображаемые кнопки
VbOKOnly	0	ОК
VbOKCancel	1	ОК, Отмена
VbAbortRetryIgnore	2	Стоп, Повтор, Про- пустить
VbYesNoCancel	3	Да, Нет, Отмена
VbYesNo	4	Да, Нет
VbRetryCancel	5	Повтор, Отмена

Значения аргумента, определяющие отображаемые значки

Константа	Значение	Значок сообще- ния
VbCritical	16	
VbQuestion	32	?
VbExclamation	48	Â
VbInformation	64	Ţ

Пример: Val=5 MsgBox "Значение val=" & val Результат работы фрагмента программы показан на рисунке 60:

Microsoft Excel	×
Значение val=5	
ОК	

Рисунок 60 – Результат работы функции вывода MsgBox

Для ввода и вывода значений выражений и переменных в активный документ используются также специальный метод **Range**. Например, для вывода значения в ячейку рабочего листа "Лист1" активной книги Excel можно записать в модуле и вызывать процедуру вида:

Sub Пример2() Dim name As String, val As Variant val = "Привет" Лист1.Range("B2").Value = val End Sub

Здесь name - координаты ячейки, записанные в кавычках, a val - имя выводимой переменной.

Результат работы этой программы показан на рисунке 61:

M	licrosoft l	Excel - K	нига2	:				
:	<u>Ф</u> айл (	Правка	<u>В</u> ид	Вст <u>а</u> вка	Фор <u>м</u> ат С	ервис <u>Д</u> ан	ные <u>О</u> кно	<u>С</u> правка
1	💕 🛃	🖪 🔒		🗟   💝 🛍	V 🕺 🗈	🖺 • 🝼	<b>v) -</b> (°' -	😣 Σ - 🗛
Ari	al Cyr		<b>v</b> 10	- Ж А	к ц∣≣	≣ ≣ 🖬	1 🕎 % (	00 5,0 ,00 00
	D4	•		fx				
	A		В	С	D	E	F	G
1								
2		Прив	зет					
3								
4								
5								
6								
7								
					1			

Рисунок 61 – Результат работы метода Range

**Ввод данных.** Для ввода значений переменных в программу применяют функцию **InputBox** или метод **Range.** 

## Синтаксис функции: InputBox (сообщение[, заголовок] [, значение по умолчанию] [, координата х] [, координата у]).

Эта функция отображает диалоговое окно, содержащее окно ввода, кнопки **ОК** и **Отмена**, сообщение (подсказку для ввода) и (необязательно) заголовок окна, значение, вводимое по умолчанию, координаты окна по горизонтали и вертикали в твипах. Заметим, что функция InputBox всегда (даже при нажатии кнопки **Отмена**) возвращает значение строкового типа. Присвоение переменной значения, введенного через диалоговое окно (рисунок 62) может иметь вид:

name = InputBox("Введи адрес яч	ейки", "Ввод",	"a1",	100,	200)
---------------------------------	----------------	-------	------	------

Ввод	×
Введи адрес ячейки	ОК
	Cancel
al	

Рисунок 62 – Результат работы функции InputBox

Для преобразования введенного значения к нужному типу данных используются функции явного приведения типа, такие как Var(строка), Str(число).

Для ввода значений переменных из активного документа используется метод **Range**. Например, для ввода значения из ячейки рабочего листа "Лист1" активной книги Excel необходимо записывать в модуле и вызывать процедуру вида:

Sub read Dim name As String, val As Variant val = Лист1.Range(name) End Sub

Здесь name - координаты ячейки, записанные в кавычках, a val - имя вводимой переменной. Функции. В VBA имеется большой список математических функций, позволяющих произвести любые вычисления.

Функция	Возвращаемое значение
Abs (x)	модуль числа
Atn (x)	arctg (x) – арктангенс от значения параметра, заданного в 114а- дианах
Sin (x)	sin (x) – возвращает синус угла от значения параметра, задан- ного в радианах
Cos (x)	cos (x) – косинус указанного в радианах угла
Tan (x)	tg (x) – возвращает тангенс угла от значения параметра, задан- ного в радианах
Exp (x)	е <sup>х</sup> – возвращает число е, возведенное в указанную степень, где е – основание натурального логарифма
Log (x)	ln (x) – возвращает натуральный логарифм от значения число- вого выражения
Sqr (x)	- возвращает квадратный корень числового выражения
Rnd (x)	Случайное число из интервала [0,1). Перед вызовом функции надо использовать оператор Randomize (рандомизации) – запуск генератора псевдослучайных чисел)
Sgn (x)	Возвращает +1, если значение параметра положительно, -1, если отрицательное, 0, если 0
Fix (x)	Возвращает результат округления выражения с плавающей точкой до целой части. В случае отрицательного параметра возвращает ближайшее большее отрицательное число
Int (x)	Возвращает результат округления выражения с плавающей точкой до целой части. В случае отрицательного параметра возвращает ближайшее меньшее отрицательное число
Hex(x)	Возвращает значение аргумента в шестнадцатеричной системе счисления
Oct(x)	Возвращает значение аргумента в восьмиричной системе счис- ления
Round(x, чис- ло знаков)	Число, округленное до заданного числа десятичных знаков

Таблица 12 – Математические функции VBA

Преобразование строки в число и обратно осуществляют следующими функциями.

Таблица 13 – Функции преобразования типов

Возвращаемое значение	
Возвращает числа, содержащиеся в строке, как числовое зна-	
чение соответствующего типа	
Возвращает значение типа variant (String), являющееся стро-	
ковым представлением числа	

#### 2.4.5 Разработка программного кода

Разработка программного кода представляет собой написание процедуры или функции.

При программировании с использованием процедур можно выделить два основных преимущества:

- процедуры позволяют разбивать программы на конечное число логических единиц, каждую из которых легче отладить, чем всю программу без процедур,
- 2) процедуры, разработанные для одной программы, могут выступать в качестве строительных блоков для других программ, обычно с небольшими изменениями или совсем без них.
- В VBA используется несколько видов процедур:
- 1) Процедуры Sub (не возвращают значения)
- 2) Процедуры Function (возвращают значения)

## Процедуры Sub

Процедура Sub (или подпрограмма) - это блок кода, который выполняется в ответ на событие. Разбивая код модуля на процедуры Sub, намного легче читать или модифицировать код приложения.

Синтаксис процедуры Sub таков:

#### [Private | Public][Static | Dim] Sub <Имя процедуры> (<аргументы>) <Операторы> End Sub

При вызове процедуры выполняются операторы между ключевыми словами Sub и End Sub. Процедуры Sub можно помещать в стандартные модули, модули классов и форм. По умолчанию процедуры Sub во всех модулях имеют атрибут public, который означает, что их можно вызывать из любого места приложения. Объявление переменной производится одним из операторов Dim, Static, Private, Public, за которым следует имя переменной и необязательная часть с ключевым словом As, после которого задается тип переменной, например Dim name [As integer]. Оператор Public используется только вне модуля, в его общей части и делает описываемую переменную доступной из всех процедур всех модулей проекта. Оператор Private служит для объявления переменной уровня модуля, доступной только процедурам данного модуля. Можно использовать также оператор Dim, но применение Private предпочтительнее как противоположное Public.

Переменные могут быть объявлены внутри процедуры операторами Dim или Static. Такие переменные называют также локальными, поскольку доступны только в той процедуре, в которой они объявлены. Данное свойство (область видимости) позволяет использовать одинаковые имена переменных в разных процедурах, не опасаясь конфликтов или случайных изменений значений переменных. Время жизни локальных переменных, объявленных с помощью оператора Dim равно времени работы процедуры и по ее окончании значения таких переменных теряются.

Переменные, объявленные с помощью оператора Static сохраняют свои значения в течении всего времени выполнения приложения. При повторном входе в процедуру, где описана такая переменная, ее значение сохраняется.

Операторы Public и Private можно применять при описании констант и процедур, что позволяет указать их область видимости. Для процедур возможно также применение оператора Static, что позволяет сделать все переменные в процедуре статическими:

Static Sub Total (num) as Integer

Это приводит к тому, что все локальные переменные в процедуре становятся статическими, независимо от того, как они определены; операторами Static, Dim, Private или неявным образом.

Задание параметров процедуры подобно объявлению переменных. Например:

```
Private Sub summa()
Dim a, b, c As Integer
a = 2
b = 3
c = b + a
MsgBox c, 1, "Сумма а и b"
End Sub
```

Результат работы этой процедуры показан на рисунке 63.



Рисунок 63 – Результат работы функции MsgBox

## Общие процедуры

Общая процедура указывает приложению, как выполнять конкретную задачу. Однажды определенная, она должна каждый раз специально вызываться приложением. В противоположность ей процедура обработки события после вызова остается в состоянии ожидания событий, вызванных пользователем или инициированных системой.

Зачем надо создавать общие процедуры? Одна причина заключается в том, что нескольким разным процедурам обработки событий может потребоваться выполнить одни и те же действия. Общие операторы помещают в отдельную процедуру (общую процедуру), а в процедуры обработки событий помещают вызовы этой процедуры. Это исключает дублирование кода и облегчает поддержку приложения.

## Просмотр существующих процедур

Чтобы просмотреть процедуру в текущем модуле, можно:

- существующую общую процедуру можно увидеть, выбрав элемент (General) в списке Object в окне кода и затем процедуру в списке Procedure,

- существующую процедуру обработки события можно увидеть, выбрав соответствующий объект в списке **Object** в окне кода и затем процедуру в списке **Procedure**.

Чтобы просмотреть процедуру в другом модуле, следует:

1) Выбрать команду Object Browser (Просмотр объектов) меню View (Вид).

2) Выбрать проект в списке Project/Library (Проект/Библиотека).

3) Выбрать модуль в списке Classes (Классы) и процедуру в списке Members of (Члены).

4) Выбрать опцию View Definition (Посмотреть определение).

## Вызов процедур Sub

Процедура Sub отличается от процедуры Function тем, что ее нельзя вызвать по имени в выражении. Ее вызов осуществляется в отдельном операторе. Также процедура Sub, в отличие от функции, не возвращает значения. Однако, как и функция, она может изменять значения любых переменных, переданных ей в качестве параметров.

#### Вызов процедур из других модулей

Процедуры, находящиеся в других модулях, могут быть вызваны из любого места проекта. Возможно, придется указать модуль, в котором содержится вызываемая процедура. Способы вызова открытых процедур разнообразны и зависят от того, где расположена процедура - в модуле формы, модуле класса или стандартном модуле.

#### Процедуры в формах

Для вызова процедуры, находящейся во внешнем модуле, т.е. в не в том модуле, в котором находится код, из которого производится вызов процедуры, перед именем процедуры должно быть указано имя модуля, в котором находится код процедуры. К примеру, если код процедуры с именем SomeSub находится в модуле формы, названном Form1, то вызвать процедуру можно следующим оператором: Call Form1.SomeSub(<аргументы>)

#### Процедуры в стандартных модулях

Процедура является уникальной, если она определена только в одном месте.

Если имя процедуры уникально, то включать имя модуля в вызов процедуры не обязательно. Внешние или внутренние вызовы процедуры будут ссылаться на это уникальное имя.

Если два или более модулей содержат процедуры с одинаковыми именами, при вызове таких процедур необходимо уточнять их с помощью имени соответствующего модуля. Вызов общей процедуры из модуля, ее содержащего, запускает процедуру этого модуля. Например, если процедура с именем CommonName существует и в модуле Module1, и в модуле Module2, то вызов CommonName из модуля Module2 запустит процедуру CommonName модуля Module2, а не процедуру CommonName модуля Module1.

Вызов общей процедуры из другого модуля должен уточнять модуль, в котором расположена процедура. Например, вызвать процедуру с именем CommonName модуля Module2 из модуля Module1 можно следующим образом:

Module2.CommonName(<аргументы>)

VBA содержит встроенные или стандартные функции, например, sqr, cos или chr. Кроме того, с помощью оператора Function можно писать собственные процедуры Function. Эти функции называют нестандартные или пользовательскими.

## Процедуры Function

Синтаксис процедуры Function таков:

## [Private|Public][Static]Function <имя процедуры> (<аргументы>) [As type <имя типа>] <операторы> End Function

Например: Function Dlina\_gipotenuzy (x, y) Dlina\_gipotenuzy = sqr(x^2+y^2) End function

Как и процедура Sub, процедура Function является самостоятельной и может принимать параметры, выполнять ряд операторов и изменять значения своих параметров. В отличие от процедуры Sub, имя процедуры Function может возвращать значение в вызывающую процедуру. Существуют три различия между процедурами Sub и Function:

- возвращаемое процедурой Function значение присваивается самому имени <имя процедуры> процедуры. Возвращаемое процедурой Function значение можно использовать в выражениях в программе,
- как и переменные процедуры Function имеют тип, который определяет тип возвращаемого значения. (В отсутствие ключевого слова As в операторе определения процедуры ей назначается по умолчанию тип variant),
- вызов процедуры Function, или просто функции, в основном осуществляется заданием ее имени и параметров в правой части большого оператора или в составе выражения.

Вот, например, функция, вычисляющая гипотенузу прямоугольного треугольника при заданных катетах:

В VBA процедура Function вызывается точно так же, как и любая встроенная функция:

 $strX = Dlina_gipotenuzy (x, y)$ 

#### 2.4.6 Вопросы для самоконтроля

- 1) Дайте определение данным, константам.
- 2) Какие типы данных и констант вы знаете?
- 3) Перечислите те типы данных, которые поддерживает VBA.
- 4) Назовите правила, предъявляемые VBA к формированию имен процедур, данных и констант.
- 5) Какие типы операций поддерживает VBA?

- 6) Дайте определение понятию «выражение». Какие различаются виды выражений?
- 7) Что такое оператор? Перечислите виды операторов в VBA?
- 8) Как можно ввести данные в процедуру?
- 9) Как можно вывести результат?
- 10) Перечислите математические функции языка VBA, функции преобразования типов.
- 11) Что такое процедура? Какие преимущества при использовании процедур существуют?
- 12) Какие виды процедур вы знаете?
- 13) В каких случаях используются операторы Dim, Static, Private, Public?
- 14) В чем назначение оператора Function?
- 15) Назовите различия между процедурами Sub и Function.

#### 2.5 Отладка, использование среды для отладки программ

#### 2.5.1 Типы ошибок

Поскольку идеальных программистов не существует, в большинстве систем разработки программ имеются инструменты, с помощью которых можно решить проблемы, возникающие в процессе программирования. В VBA также есть средства, которые позволяют либо исключить ошибки при разработке, либо задать обработку ошибок при выполнении программ.

Отладка программ - это проверка и внесение исправлений в программу при ее разработке. Отладка позволяет идентифицировать ошибки, допущенные при программировании (синтаксические - ошибки в выражениях и именах, и логические - в логике работы программы).

Обработка ошибок - это задание реакции на ошибки, которые возникают при выполнении программы. Их причиной могут быть как ошибки программиста, так и внешние факторы - отсутствие нужных файлов, отказы аппаратуры, неправильные действия пользователя.

Типы ошибок. Ошибки в программе делятся на три категории:

 Ошибки компиляции - возникают, когда компилятор не может интерпретировать введенный текст. Некоторые ошибки компиляции обнаруживаются при вводе, а другие - перед выполнением программы. Такие ошибки легко определить и исправить, поскольку VBA выявляет их автоматически, а сами ошибки очевидны.

*Примечание*. VBA автоматически компилирует программу каждый раз при запуске на выполнение после внесения изменений. Можно также запустить компиляцию командой **Отладка**\**Компилировать**.

2) Ошибки выполнения - возникают при выполнении программы после успешной компиляции. Их причиной обычно является отсутствие данных или неправильная информация, введенная пользователем. Такие ошибки идентифицируются VBA с указанием инструкции, при выполнении которой произошла ошибка. Для исправления таких ошибок обычно приходится выводить значения переменных или другие данные, которые влияют на успешное выполнение программы.

 Логические ошибки трудно заметить и устранить. Они не приводят к прекращению компиляции или выполнения, однако являются причиной того, что программа не выдает желаемых результатов. Выявление таких ошибок производят путем тщательной проверки с помощью средств отладки VBA.

В VBA имеется большое количество средств, предназначенных для отладки программ. К ним относятся: использование оператора Option Explicit, пошаговое выполнение программы, работа в режиме прерывания, использование точек останова, вывод значений переменных.

Использование Option Explicit. Данный оператор описания требует явного задания переменных в программах. При его использовании возникает ошибка компиляции при неправильном написании имени переменной или использовании неописанной переменной. Кроме того, явное описание переменных позволяет обойтись без использования типа данных Variant и связанных с его использованием ошибок при неявном приведении типов данных.

Пошаговое выполнение программы. Этот режим служит для локализации ошибок в теле программы. Для его запуска используются команды меню, клавиатура или панель инструментов Отладка, отображаемая командой **Вид\Панели инструментов\Отладка**. В меню и на панели инструментов имеются четыре команды (кнопки) для выполнения программы в пошаговом режиме.

- Команда Отладка\Шаг с заходом (клавиша F8) позволяет выполнить одну строку программы и перейти к следующей. Если следующая строка - вызов процедуры, то происходит переход к первому выполняемому оператору этой процедуры.
- 2) Команда Отладка\Шаг с обходом (клавиши Shift+F8) также выполняет одну строку программы, но если строкой является вызов процедуры, то она выполняется как одна инструкция. Данная команда используется, если известно, что эта процедура работает правильно.
- 3) Команда Отладка\Шаг с выходом (клавиши Ctrl+Shift+F8) заканчивает выполнение текущей процедуры и останавливается на следующей после вызова текущей процедуры инструкции в вызывающей подпрограмме.
- 4) Команда Отладка\Выполнить до текущей позиции (клавиши Ctrl +F8) выполняет программу от текущей до выбранной инструкции. Перед выбором данной команды требуется установить курсор в окне модуля на требуемую позицию.

Работа в режиме прерывания. Переход в данный режим выполняется:

- При нажатии кнопки Отладка в окне сообщения об ошибке выполнения.
- При прерывании работы программы нажатием клавиш Ctrl+Break. Текущая строка программы выделяется в окне модуля.

- По достижении точки останова.
- По достижении оператора Stop.
- При пошаговом выполнении программы.

В режиме прерывания можно:

- вывести значение переменной,
- вычислить выражение в окне отладки,
- сбросить программу,
- выполнить программу в пошаговом режиме,
- продолжить выполнение программы.

Для выхода из режима прерывания используется команда Запуск\Сброс.

Использование точек останова. Точка останова - это строка в процедуре, на которой приостанавливается выполнение программы. Все команды, находящиеся выше точки останова, выполняются с обычной скоростью, а по достижении контрольной точки программа переходит в режим прерывания. Затем можно отлаживать процедуру в пошаговом режиме, либо использовать различные способы вывода значений переменных. Кроме того, имеется возможность остановить выполнение или сбросить процедуру командами меню Запуск или кнопками панели инструментов Отладка. В одном проекте можно задать несколько точек останова, причем в различных процедурах.

Чтобы установить или снять точку останова, используется команда Отладка\Точка останова или клавиша F9, либо кнопка Точка останова панели инструментов Отладка. Можно также установить или снять точку останова, щелкнув левой кнопкой мыши на полосе индикатора против требуемой строки. Точка останова отмечается коричневой жирной точкой на полосе индикатора, а сама строка выделяется коричневым цветом.

Чтобы быстро удалить все точки останова открытого проекта, используется команда **Отладка**\**Снять** все точки останова или комбинация клавиш Ctrl +Shift+F9. Для данной команды не предусмотрена кнопка на панели инструментов Отладка.

Вывод значений переменных. При наличии тестового примера вывод значений переменных позволяет сравнить ожидаемые и полученные значения переменных. Для отображения значений переменных в режиме прерывания необходимо:

- при установленном флажке Подсказки значений переменных в окне Сервис\Параметры достаточно переместить указатель мыши на требуемую переменную для отображения имени и значения переменной во всплывающей подсказке,
- 2) выбрать команду Отладка\Контрольное значение (нажать клавиши Shift+F9) для вывода диалогового окна Контрольное значение. При этом курсор должен находиться возле переменной, значение которой надо контролировать. В окне Контрольное значение отображается контекст (имя модуля и процедуры), выделенное выражение (переменная) и кнопки Добавить и Отмена. При нажатии кнопки Доба-

вить откроется окно Контрольные значения, содержащее имена переменных (выражения), их значения, тип данных и контекст,

- 3) для добавления других контрольных значений используется команда **Отладка\Добавить** контрольное значение.
- 4) выбрать команду **Вид\Окно** локальных переменных. Откроется окно Локальные переменные, в котором в режиме прерывания отображаются имена, значения и типы всех переменных модуля,
- 5) выбрать команду **Вид\Окно** отладки. В нем немедленно выполняется введенная в него инструкция, обычно операция отображения значения выражения вида Print имя, или операция присваивания значения переменной. Выполняемая программа также может выводить информацию в это окно с помощью выражения Debug.Print имя переменной.

## 2.5.2 Обработка ошибок выполнения

Как уже отмечалось выше, одной из основных проблем являются ошибки выполнения. Эти ошибки возникают при попытке произвести недопустимую операцию, например, при открытии несуществующего файла, делении на ноль, или при обработке неправильно введенных пользователем данных.

Так или иначе, при возникновении ошибки выполнение программы прерывается и выводится сообщение. К сожалению, это сообщение несет мало полезной информации, поскольку в нем указывается только недопустимая операция, а не причина возникновения ошибки. Еще хуже то, что программа не выполняется до конца и не произведены все необходимые действия.

Пользователь может без всякого умысла произвести действия, которые нанесут непоправимый вред данным. Следовательно, необходимо каким-то образом оградить себя от возникновения подобных ситуаций.

При программировании имеются два подхода:

- предотвращение ошибочных ситуаций,
- обработка ошибки с помощью специальной процедуры.

При разработке программ рекомендуется по возможности предотвращать возникновение ошибочных ситуаций. Начиная создание алгоритма макроса, следует ответить на ряд вопросов, например:

- существует ли файл, который требуется открыть?
- находится ли курсор в требуемой для выполнения макроса позиции?

#### Перехватываемые ошибки.

Перехват используется в том случае, когда предотвратить возникновение ошибочных ситуаций невозможно. Приведем некоторые возможные перехватываемые ошибки и их коды.

КОД	СООБЩЕНИЕ
3	Инструкция Return без Gosub
6	Переполнение
7	Не хватает памяти
9	Индекс выходит за пределы допустимого диапазона
11	Деление на 0
18	Произошло прерывание, вызванное пользователем
35	Процедура Sub, Function или Property не определена
53	Файл не найден
61	Переполнение диска
71	Диск не готов
91	Не задана объектная переменная блока With
97	Невозможен вызов процедуры Friend для объекта, не являю-
	щегося экземпляром определяющего класса
335	Невозможен доступ к системному реестру
368	Истек срок данного системного файла. Программе требуется
	файл более новой версии
402	Сначала необходимо закрыть самую верхнюю модаль-
	ную форму
422	Свойство не найдено
440	Ошибка программирования объектов
448	Именованный аргумент не найден
482	Ошибка принтера
	* *

Очевидно, что ни одно из показанных сообщений не дает пользователю исчерпывающих сведений о причине возникновения ошибки. Профессионал может точно определить проблему, используя систему справочной информации, а человеку, который захочет, например, написать письмо или ввести данные в отчет с помощью разрабатываемого приложения, такие сообщения непонятны. Предотвращая возникновение ошибок или перехватывая их, можно исключить вывод стандартных сообщений. Вместо них появляется возможность либо отобразить собственную подсказку, либо попытаться исправить положение с помощью макроса. Ниже объясняется, как перехватить ошибку.

## 2.6 Создание макросов при помощи макрорекордера

Макрос – это набор инструкций, которые сообщают программе (например, Word, Excel или Access), какие действия следует выполнить, чтобы достичь определенной пользователем цели. Макрос представляет собой процедуру на VBA (но не всякая процедур является макросом!).

Макросы предназначены для автоматизации работы в приложениях. Макросы пишутся пользователем и выполняют вместо него рутинную и повторяющуюся работу. Например, в Word'е можно написать макрос для комплексного форматирования фрагмента текста (чтобы не нажимать три кнопки с пиктограммами Ж, К, Ч пишется один макрос, которые выполнит сразу три действия по форматированию), в Excel'е можно написать макрос, выделяющий не прямоугольную фигуру ячеек (рисунок 64) или макрос для вычисления функции, не существующей в Мастере функций. Обычно макросы вызываются определенными разработчиком макросов клавишами (например, Makpocyl можно назначить сочетание клавиш Ctrl+r и вызывать его ими) или могут заноситься в Мастер функций (Excel).



Рисунок 64 – Программный код макроса, записанного при помощи макрорекордера, выделяющего непрямоугольную область ячеек

Макрорекордер – средство записи макросов, обеспечивающее автоматическую запись пользователя в виде операторов VBA, выполняющих какиелибо действия с объектами приложения в котором ведется запись.

Для того, чтобы записать макрос, необходимо выполнить команду

#### Сервис - Макрос - Начать запись.

После этого появится диалоговое окно (рисунок 65) Запись макроса, в котором необходимо дать макросу имя и сопоставить его с какими-то удобными вам клавишами для последующего вызова этого макроса. Для этого необходимо нажать кнопку Клавишам в окне Запись макроса и далее вписать нужные клавиши.

Запись макроса	×
Имя макроса:	
Макрос7	
Назначить макрос	
панели	<u>клавишам</u>
Макрос доступен для:	
Bcex документов (Normal.dot)	•
<u>О</u> писание:	
Макрос записан 23.05.2006 Vp	
	ОК Отмена

## Рисунок 65 – Окно Запись макроса

После выполненных действий вы можете приступать к записи макроса, т.е. выполнять любую нужную вам последовательность команд либо с клавиатуры, либо мышью. Рядом с курсором мыши будет присутствовать значок кассеты, что будет напоминать вам о том, что идет запись каждого вашего действия, а также последовательности ваших действий. В левом верхнем углу рабочей области появится небольшая дополнительная панель макросов (рисунок 66). С ее помощью вы сможете приостановить или окончить запись макроса.



Рисунок 66 – Панель записи макроса

Закончить запись макроса можно также по команде

## Сервис – Макрос - Остановить запись

Для того, чтобы запустить записанный макрос, необходимо нажать то сочетание клавиш, которое было выбрано в окне *Запись макроса* или выполнить команду

## Сервис-Макрос - Макросы.

Появится диалоговое окно **Макрос** (рисунок 67), в котором будут перечислены все существующие в ваших приложениях макросы. перечисленные макросы можно запускать (кнопка **Выполнить**) или видоизменять (кнопка **Изменить**). При видоизменении (отладке) запускается окно VBA, где вы можете просмотреть и изменить некоторые команды макроса.

Макрос	? ×
Имя:	
1	<u>В</u> ыполнить
fi	
Normal.AutoExit.MAIN	О <u>т</u> ладка
Peshka RemoveFineYMI	Изменить
RunFineReader	
Show Makport	Созд <u>а</u> ть
Makpoc10	Удалить
Makpoc2 Makpoc3	
Макрос4	
Макросы из: Активных шаблонов	Отмена
Описание:	
Макрос записан 24.03.2006 Vp	
1	

#### Рисунок 67 – Окно Макрос.

Следует отметить, что существует огромное количество действий, которые выполняет пользователь при записи макроса и все их запомнить очень сложно. Поэтому, записывая макрос, старайтесь запоминать действия, которые вы выполняете, а, открыв макрос для редактирования в VBA, сравнивайте выполненные вами команды с тем, что записано в программе макроса. Таким образом, вы можете узнать, как описываются действия пользователя программно.

Задание. Используя Excel, напишите макрос, который позволяет выделить не прямоугольную фигуру ячеек. Для этого начните запись макроса и мышью выделите любую не прямоугольную область из ячеек. Остановите запись. Просмотрите макрос в VBA. Выполните макрос, используя клавиши клавиатуры.

## 2.7 Реализация алгоритмов линейной структуры средствами VBA

## 2.7.1 Примеры решения задач с линейными алгоритмами

Обычно в задачах, реализующих в своем решении алгоритм линейной структуры, используются операторы присваивания, а также функции ввода и вывода информации. Рассмотрим несколько примеров.

**Пример 1**. Пусть в Excel в ячейке B2 находится значение длины одного катета прямоугольного треугольника, а в ячейке D2 – второго. Найти длину гипотенузы этого треугольника и:

а) поместить ее в ячейку F2,

б) вывести в диалоговое окно.

Процедура, позволяющая решить данную задачу, выглядит так (рисунок 68).

## Пример1, а)

S Книга1 - Module1 (Code)				
(General)	🔻 gipotenuza			
Sub gipotenuza() Dim a, b As Integer Dim c As Single a = Лист1.Range("B2") b = Лист1.Range("D2") c = Sqr(a ^ 2 + b ^ 2) Лист1.Range("F2") = c				

Рисунок 68 – Программный код процедуры нахождения длины гипотенузы с выводом результата в ячейку Excel

Результат в Excel показан на рисунке 69.

icrosoft	: Excel - K	нига1							
<u>Ф</u> айл	Правка	<u>В</u> ид	Вст <u>а</u> вка	Фор <u>м</u> ат С	ервис Дан	ные <u>О</u> кно	⊆правка		
💕	l 💪 🔒	3	💁   💞 🛍	1 X 🗈	🖺 • 🝼	ii) = (2i =	$ \sum_{n \in \mathbb{Z}} \Sigma = A_n \downarrow $	A 🛄 🕴	100%
al Cyr		<b>v</b> 10	- Ж	К П  ≣	≣ ≣ ₫	9 🖓 % 000	) *,0 ,00   撞	≹≣∣⊡ -	ه 🕙
G2	-		fx .						
A	E	3	С	D	E	F	G	Н	
	Kam	em 1		Kamem 2		Гипотенуз	a		
		3		4			5		

Рисунок 69 – Результат работы процедуры

**Пример 1, б).** Процедура и результат ее работы показаны соответсвенно на рисунках 70 и 71.

4	(нига1 - Module1 (Code)		
(6	(General)		
	Sub gipotenuza() Dim a, b As Integer Dim c As Single a = Лист1.Range("B2") b = Лист1.Range("D2") c = Sqr(a ^ 2 + b ^ 2) MsgBox c, 1, "Длина гипотенузы" End Sub		

Рисунок 70 – Программный код процедуры нахождения длины гипотенузы с выводом результата в диалоговое окно

Длина гипотенузы			
5			
ОК	Отмена		

Рисунок 71 – Результат работы процедуры – диалоговое окно

**Пример 2.** Найти значение функции  $y = \frac{4x^3 + \sqrt{\cos(2x) + 3}}{x - x^6}$ 

Решим данную задачу, создав пользовательскую функцию для библиотеки функций Excel. Для этого будем создавать не процедуру, а функцию, которая будет выглядеть так, как показано на рисунке 72.



Рисунок 72 – Программный код пользовательской функции

Далее необходимо выполнить команду Вставка – Функция - Категория Определенные пользователем – функция и ввести аргументы функции в диалоговое окно Аргументы функции (рисунок 73).

Аргументы функции		X
-функция	<b>x</b> 5	<b>5</b>
Справка недоступна.		= -0,032104354
	x	
Справка по этой функции	Значение:-0,032104354	ОК Отмена

Рисунок 73 – Диалоговое окно для ввода аргументов пользовательской функции

Результат функции будет введен в активную ячейку (рисунок 74)

🔀 Microsoft Excel - Книга1								
:	Файл 🛛	равка <u>В</u> ид	Вст <u>а</u> вка	Фор <u>м</u> ат С	<u>е</u> рвис <u>Д</u> ан	іные <u>О</u> кно	⊆пр	
8 🗋	: 🗅 💕 🛃 💪 🚔 🖾 🔍 💖 🚉 🗼 🗈 🛍 • 🕩 • 🕅 • 🧶							
Ari	Arial Cyr 🔹 10 🗸 🗶 🗶 🖳 🗮 📰 🕎 % 000 %							
	B2 <b>▼ f</b> ≈ =функция(5)							
	A	В	С	D	E	F	(	
1								
2		-0,0321	1					
3								
4								
5								
6								

Рисунок 74 – Значение функции, помещенное в выделенную ячейку Ех-

## 2.7.2 Задания для самостоятельной работы

- 1) Вычислить длину окружности, площадь круга и объем шара одного и того же радиуса. Значение радиуса разместите в ячейке В1, а результаты выведите в произвольные ячейки.
- Дискета 3,5А вмещает 1,44 Мбайт. Рукопись содержит Х страниц текста. На каждой странице Y строк по Z символов в каждой. Сколько дискет потребуется для записи рукописи? X,Y,Z располагаются в разных ячейках, а количество дискет необходимо вывести через диалоговое окно.
- 3) В ячейке СЗ находится значение валютного курса евро на сегодняшний день, а в ячейке D3 – количество рублевой суммы, имеющейся у пользователя. Сколько евро он сможет купить на эту сумму?
- 4) После продажи квартиры у пользователя на руках имеется 52 000 \$, которые он помещает в банк. Банк начисляет 1% в первый месяц, а каждый следующий тоже1%, но уже с получившейся суммы. Сколько денег будет на счету у пользователя через год?
- 5) Длина отрезка задан в дюймах. Выразить это значение в метрах, сантиметрах, миллиметрах.
- 6) В банке приобретен вексель, по которому через год должно быть получено S рублей. В момент приобретения цена векселя составляла Р рублей. Определить доходность сделки, т.е. размер процентной ставки. Ставка рассчитывается по формуле I=J/(p\*n), где J=S\*P, а n – срок ссуды.
- 7) Дан список группы студентов, состоящий из 10 человек, в котором указаны ФИО каждого студента, а также размер его стипендии за октябрь. В ноябре стипендия была повышена на 17%. Написать процедуру расчета размера новой стипендии и внести изменения в список, рассчитав для каждого студента размер стипендии за ноябрь.

cel

8) Рассчитать значение следующего выражения, создав для него пользо-

$$\frac{b+\sqrt{b^2+4ac}}{2a}-a^3c+b^{-1}$$

вательскую функцию: 2а

9) Рассчитать значение следующего выражения, создав для него пользо-

$$\frac{a}{b} \cdot \frac{b}{b} - \frac{ab-c}{b}$$

вательскую функцию: c d cd

10) Рассчитать значение следующего выражения, создав для него поль-

$$\frac{\sin(x) + \cos(y)}{\cos(x) - \sin(y)} \cdot tg(xy)$$

зовательскую функцию:  $\cos(x) - \sin(y)$ 

11) Рассчитать значение следующего выражения, создав для него поль-

$$\frac{x+y}{x+1} - \frac{xy-12}{34+x}$$

зовательскую функцию x+1 34 + x

12) Рассчитать значение следующего выражения, создав для него поль-

зовательскую функцию: 
$$\frac{3 + e^{y-1}}{1 + x^2 |y - tg(x)|}$$

13) Рассчитать значение следующего выражения, создав для него поль-

$$x - \frac{x^3}{3} + \frac{x^5}{5}$$

зовательскую функцию:

14) Рассчитать значение следующего выражения, создав для него поль-

$$\ln \left| (y - \sqrt{|x|}) \left( x - \frac{y}{x + \frac{x^2}{4}} \right) \right|$$

зовательскую функцию:

15) Рассчитать значение следующего выражения, создав для него пользовательскую функцию:  $(1 - tg(x))^{ctg(x)} + cos(x - y)$ 

## 2.8 Реализация алгоритмов ветвления средствами VBA

## 2.8.1 Условный оператор If

VBA поддерживает следующие операторы, позволяющие описать программно алгоритм ветвления:

- If... Then (краткий условный оператор)
- If ... Then ... Else (полный условный оператор)

## - Select Case (оператор выбора)

Краткий условный оператор If... Then применяется, когда необходимо выполнить один или группу операторов в зависимости от некоторого условия. Синтаксис этого оператора позволяет задавать ее в одной строке или в нескольких строках программы: *If* условие *Then* выражение или *If* условие *Then* выражение *End If* 

Обычно условие является простым сравнением, но оно может быть любым выражением с вычисляемым значением. Это значение интерпретируется как False (Ложь), если оно нулевое, а любое ненулевое рассматривается как True (Истина). Если условие истинно, то выполняются все выражения, стоящие после ключевого слова Then. Для условного выполнения одного оператора можно использовать как синтаксис для одной строки, так и синтаксис для нескольких строк (блоковую конструкцию).

Заметим, что синтаксис оператора If . . . Тhen для одной строки не использует оператор End If – однострочный оператор If.

Следующие два оператора эквивалентны:

If x < 0 Then  $x = x^2$   $\mu$ If x < 0 Then  $x = x^2$ End If

Если условие ложно, то операторы, стоящие после ключевого слова Then не выполняются, а управление передается на следующую строку (или строку после оператора End If в блочной конструкции).

**Полный условный оператор** *If* . . . *Then* . . . *Else* определяет несколько блоков операторов, один из которых будет выполняться в зависимости от условия:

*If* условие 1 *Then* выражение 1 *ElseIf* условие 2 *Then* выражение 2 .... *Else* выражение п *End If* 

При выполнении данного оператора сначала проверяется условие1. Если оно ложно, VBA проверяет следующее условие2 и т. д., пока не найдет истинного условия. Найдя его, VBA выполняет соответствующий блок операторов и затем передает управление инструкции, следующей за оператором End if. В

данную конструкцию можно включить блок оператора Else, который VBA выполняет, если не выполнено ни одно из условий.

Оператор If . . . Then . . . ElseIf в действительности всего лишь специальный случай конструкции If . . . Then . . . Else. Заметим, что в данной конструкции может быть любое число блоков ElseIf, или даже ни одного. Блок Else можно включать независимо от присутствия или, наоборот, отсутствия блоков ElseIf.

Рассмотрим пример вычисления функции

$$z = \begin{cases} \sin(x), e c \pi u \ x \le a \\ \cos(x), e c \pi u \ x \ge b \\ tg(x), e o c m a \pi b + b x c \pi y u a x \end{cases}$$

```
Private Sub пример1()

Dim a As Single, b As Single, x As Single

Dim z As Double

a=Лист1.range("A1")

b=Лист1.range("B1")

x = Val(InputBox("Введите x", "Ввод данных", 0))

If x \le a Then

z = Sin(x)

ElseIf x \ge b Then

z = Tan(x)

Else: z = Cos(x)

End If

MsgBox z

End Sub
```

Заметим, что можно добавить любое число блоков Elself в конструкцию If . . . Then. Однако количество блоков Elself может стать настолько большим, что конструкция If . . . Then станет очень громоздкой и неудобной. В подобной ситуации следует применять другую конструкцию принятия решения - Select Case.

## 2.8.2 Оператор выбора Select Case

**Оператор выбора** Select Case является альтернативой конструкции If . . . Then . . . Else в случае выполнения блока, состоящего из большого набора операторов. Конструкция Select Case предоставляет возможность, похожую на возможность конструкции If . . . Then . . . Else, но в отличие от нее она делает код более читаемым при наличии нескольких вариантов выбора.

Оператор Select Case работает с единственным проверяемым выражением, которое вычисляется один раз при входе в эту конструкцию. Затем VBA сравнивает полученный результат со значениями, задаваемыми в операторах Case конструкции. Если найдено совпадение, выполняется блок операторов, ассоциированный с оператором Case:

Select Case проверяемое\_выражение Case список\_выражений1 [блок\_операторов1] Case список\_выражений2 [блок\_операторов2] .... Case Else [блок\_операторовп] End Select

Каждый список выражений является списком из одного или более значений. Если в одном списке больше одного значения, они отделяются запятыми. Каждый блок операторов содержит несколько операторов или ни одного. Если окажется, что вычисленному значению проверяемого выражения соответствуют значения из нескольких операторов Case, то выполняется блок операторов, ассоциированный с первым оператором Case из всех найденных соответствий. VBA выполняет блок операторов, ассоциированный с оператором Case Else (заметим, что он необязателен), если не найдено ни одного соответствия проверяемого значения выражения и значений из всех списков операторов Case.

Рассмотрим пример вычисления функции

 $x = \begin{cases} \sin x, & e c \pi u \ x = -\pi / 2 \\ \cos x, & e c \pi u \ x = 0 \\ t g \ x, & e c \pi u \ x = \pi / 2 \end{cases}$ 

Private Sub пример2() Const pi2 = 1.57 Dim x As Single Dim z As Double x = Val(InputBox("введи x", "Ввод данных", 0)) Select Case x Case -pi2 z = Sin(x) Case 0 z = Cos(x) Case pi2 z = Tan(x) Case Else MsgBox "Неверные исходные данные!" Exit Sub End Select MsgBox z End Sub

Заметим, что оператор Select Case вычисляет выражение только один раз при входе в нее, а в операторе If . . . Then . . . Еlse вычисляются различные выражения для каждого оператора Elself. Оператор If . . . Then . . . Else можно заменить оператором Select Case, только если оператор If и каждый оператор Elself вычисляют одно и то же выражение.

## 2.8.3 Задания для самостоятельной работы

1) Создайте процедуру, вычисляющую значение следующей функции  $v = \begin{cases} \frac{\cos x}{x}, x > 0 \end{cases}$ .

$$V = \left(\frac{x \sin x, x \le 0}{x \sin x, x \le 0}\right)$$

Значение аргумента х запросите у пользователя через диалоговое окно.

- Создайте процедуру, считывающую из трех ячеек три разных целых положительных числа и выводящую в диалоговое окно сумму двух меньших из них
- 3) Банк предлагает два вида вклада: по p1% на 3 месяца и под p2% на 6 месяцев. Какой из вкладов наиболее выгоден для вкладчика?
- 4) Создайте процедуру для расчета торговой скидки. Пусть в ячейке A1 находится сумма покупок. Через диалоговое окно необходимо выдать сумму к оплате с учетом того, что при покупке товара на сумму от 500 до 1000 рублей предоставляется скидка 5%, на сумму от 1001 до 10000 рублей 7%, на сумму свыше 10000 9%.
- 5) По введенному в ячейку названию месяца вывести название времени соответствующего сезона.
- 6) Для возраста человека, заданного в годах (от 0 до 100 лет) вывести соотвествующие фразы:
- мне 21 гоД,
- мне 32 гоДА,
- мне 15 ЛЕТ и т.д.
- 7) Даны три числа x, y, z. Найти max (x<sup>3</sup>, xyz) и min (y/z, x- y<sup>2</sup>, z<sup>3</sup>-y<sup>3</sup>).
- 8) Известно, что сторона квадрата равна a, a радиус круга r. Выведите в ячейку название фигуры с большей площадью.
- 9) Рассматриваются два варианта покупки недвижимости: заплатить сразу сумму в 70000долл или платить ежемесячно по 800 долл. в течении 12 лет при годовой ставке в 9%. Что предпочтительней?
- 10) Услуги телефонной сети оплачиваются по следующему правилу: за разговоры до А минут в месяц оплачиваются В р., а разговоры сверх установленной нормы оплачиваются из расчета С р. в минуту. Напи-

сать программу, вычисляющую плату за пользование телефоном для введенного времени разговоров за месяц.

- 11) Определить правильность даты, введенной с клавиатуры (число от 1 до 31, месяц от 1 до 12). Если введены некорректные данные, то сообщить об этом.
- 12) Даны числа х, у, z. Найти значение выражения:

 $\frac{\max^2 \{x, y, z\} - 2^x \cdot \min\{x, y, z\}}{\sin 2x + \max\{x, y, z\} / \min\{x, y, z\}}$ 

- 13) Дано число х. Напечатать в порядке возрастания числа: sinx, cosx. lnx. Если при каком-либо х некоторые из выражений не имеют смысла, вывести сообщение об этом и сравнивать значения только тех, которые имеют смысл.
- 14) В небоскребе N этажей и всего один подъезд; на каждом этаже по 3 квартиры; лифт может останавливаться только на нечетных этажах. Человек садится в лифт и набирает номер нужной ему квартиры М. На какой этаж должен доставить лифт пассажира?
- 15) Написать программу, которая по номеру дня недели (натуральному числу от 1 до 7) выдает в качестве результата количество пар в вашей группе в этот день.

16) Составить программу, которая в зависимости от выбора названия республики СНГ выводит на название ее столицы.

## 2.8.4 Вопросы для самоконтроля

- 1) Какие операторы VBA позволяют описать алгоритмы ветвления?
- 2) В каких случаях применяется оператор *If* . . . *Then*, опишите его синтаксис. Приведите примеры программ, использующих оператор *If* . . . *Then*.
- 3) В каких случаях применяется оператор *If* . . . *Then* . . . *Else*, опишите его синтаксис. Приведите примеры программ, использующих оператор *If* . . . *Then* . . . *Else*.
- 4) В каких случаях используется оператор *Select Case?* Опишите синтаксис данного оператора.
- 5) Приведите пример программы, использующей в своей работе оператор *Select Case*.

## 2.9 Реализация циклических алгоритмов средствами VBA

## 2.9.1 Циклические операторы

Циклы позволяют выполнить одну или несколько строк кода несколько раз. VBA поддерживает следующие циклы:

– цикл с предусловием,

– цикл с постусловием,

– цикл со счетчиком.

1) Цикл с предусловием (цикл-пока) — наиболее универсальная циклическая структура. Он организует выполнение операторов, составляющих тело цикла, неизвестное заранее число раз. Реализуется оператором While. Формат оператора:

*Do While* <условие> <тело цикла> *Loop* 

Здесь *Do, While, Loop* – зарезервированные слова; <условие> – выражение логического типа; <тело цикла> – операторы VBA.

Алгоритм работы оператора следующий. Вначале вычисляется значение выражения *<условие>*. Если *<условие>* имеет значение True, выполняется *<тело цикла>;* после чего вычисление значения выражения *<условие>* повторяется. Если *<условие>* имеет значение False, оператор прекращает свою работу.

Таким образом, выход из цикла осуществляется, если логическое выражение принимает значение ложь. Истинность логического выражения проверяется вначале каждого прохождения цикла, поэтому тело цикла может не выполняться ни разу.

2) Цикл с постусловием (цикл-до) позволяет организовать многократное выполнение операторов, если число повторений заранее неизвестно.

Цикл с постусловием может быть записан в одном из следующих видов:

*Do Until* <условие> <тело цикла> *Loop* или *Do* <тело цикла> *Loop Until* <условие>

Здесь *Do, Until, Loop* – зарезервированные слова; <условие> – выражение логического типа; <тело цикла> – операторы VBA.

Оператор работает по следующему алгоритму. Вначале выполняется *<тело цикла>*, после чего вычисляется значение логического выражения *<условие>*. Если его значение есть False, операторы, образующие *<тело цикла>*, повторяются. В противном случае оператор завершает свою работу.

То есть выход из цикла осуществляется, если логическое выражение принимает значение True (истина). Поскольку значение логического выражения вычисляется в конце каждого прохождения цикла, тело цикла выполнится хотя бы один раз.

3) Цикл с параметром (цикл со счетчиком, цикл – для) служит для организации циклов с заранее известным числом повторений.

Синтаксис оператора:

*For* <параметр> = <начальное значение> *To* <конечное значение> [*Step* <шаг>] <тело цикла> *Next* 

Здесь For, To, Step, Next – зарезервированные слова VBA;

<параметр> – простая переменная порядкового типа;

<начальное значение> – выражение того же типа что и <параметр>, определяющее начальное значение параметра;

<конечное значение> – выражение того же типа, определяющее конечное значение параметра;

<шаг> – некоторое значение типа <параметр>, задающее, на сколько изменяется значение параметра при каждом проходе цикла;

<тело цикла> – операторы VBA.

Этот оператор организует повторение группы инструкций, образующих *«мело цикла»*, пока *«параметр»* изменяется от *«начального значения»* до *«конечного значения»* с указанным шагом. Цикл выполняется столько раз, сколько нужно, чтобы *«параметр»* от *«начального значения»* достиг *«конечного значения»*.

При выполнении оператора вначале вычисляется значение выражения *<начальное значение>* и осуществляется присваивание *<napamemp>* = *<начальное значение>*. После этого циклически повторяется:

- проверка условия <*napamemp*> <= <*конечное значение*>, если значение выражения <*шаг*> задано и имеет положительное значение, или
   <*параметр*> >= <*конечное значение*>, если значение выражения
   <*шаг*> задано и имеет отрицательное значение;
- выполнение < тела цикла>;
- изменение значения переменной *<параметр>* на величину, указанную после слова *Step*; если *<шаг>* не указан, он полагается равным 1.

# 2.9.2 Примеры задач, содержащих циклические алгоритмы в решении

Рассмотрим некоторые примеры.

**Пример1.** Найти сумму десяти случайных чисел. Напишем программу, воспользовавшись циклами различных видов.

```
Private Sub Сумма()
Dim sum1 As Integer, sum2 As Integer, i As Integer
Randomize
*далее решим задачу, используя разные операторы циклов*
a) цикл Do While
i = 10
Do While i > 0 'цикл выполняется, пока логическое условие истинно
sum1 = sum1 + Int((10 * Rnd) + 1)
i = i - 1
Loop
MsgBox "Сумма чисел=" & sum1
б) цикл Do Until
```

```
i = 10
Do 'цикла выполненяется, пока логическое условие ложно
sum2 = sum2 + Int((10 * Rnd) + 1)
i = i - 1
Loop Until i = 0
MsgBox "Сумма чисел=" & sum2
End Sub
```

Заметим, что особенностью интерпретатора VBA является то, что значения переменных числовых типов перед выполнением процедуры полагаются равными 0. Поэтому в программе отсутствуют команды присваивания вида: sum1 = 0 и sum2=0.

Пример 2. Найти максимальное из п введенных с клавиатуры чисел.

Приведем два варианта решения задачи с использованием циклов разных видов.

а) цикл Do While

```
Private Sub Max_n_while()
Dim n As Byte, k As Single, i As Byte, Max As Single
n = Val(InputBox("Введите количество чисел"))
i = 1
Do While i <= n
k = Val(InputBox("Введите число", "Ввод чисел"))
If i = 1 Then Max = k
If k > Max Then Max = k
i = i + 1
Loop
MsgBox "Наибольшее из чисел " & Max
```

End Sub

б) цикл Do Until

```
Private Sub Max_n_until()
Dim n As Byte, k As Single, i As Byte, Max As Single
n = Val(InputBox("Введите количество чисел"))
i = 1
Do Until i > n
k = Val(InputBox("Введите число", "Ввод чисел"))
If i = 1 Then Max = k
If k > Max Then Max = k
i = i + 1
Loop
MsgBox "Наибольшее из чисел " & Max
End Sub
```

Если из текста программ удалить строку: If i = 1 Then Max = k, то программа будет работать корректно только в случае, когда хотя бы одно вводимое число неотрицательно. Это объясняется тем, что начальное значение переменной Max считается равным 0.

**Пример 3.** Найти сумму n первых членов ряда 1, 1/2, 1/3, ... 1/n,...

```
Private Sub Summ_n()
Dim n As Byte, i As Byte, sum As Single
n = Val(InputBox("Введите количество членов ряда"))
For i =1 To n
sum = sum + 1 / i
Next
MsgBox "Сумма " & sum
End Sub
```

Пример 4. Найти сумму всех четных чисел в первой десятке:

```
Private Sub Summa ()
Dim j As Integer, sum As Integer
For j = 2 To 10 Step 2
sum = sum + j
Next
MsgBox "Сумма равна " & sum
End Sub
```

## 2.9.3 Задания для самостоятельной работы

1) Имеется серия измерений элементов треугольника. Группы элементов пронумерованы. В серии в произвольном порядке могут встречаться такие группы элементов треугольника:

- основание и высота;

- две стороны и угол между ними (угол задан в радианах); - три стороны.

Разработать программу, которая запрашивает номер группы элементов, вводит соответствующие элементы и вычисляет площадь треугольника. Вычисления прекратить, если в качестве номера группы введен 0.

- 2) Составить программу, которая печатает таблицу умножения и сложения натуральных чисел в десятичной системе счисления.
- 3) Составить программу, которая печатает таблицу умножения и сложения натуральных чисел в шестнадцатеричной системе счисления.
- 4) Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?
- 5) Ежемесячная стипендия студента составляет А руб., а расходы на проживание превышают стипендию и составляют В руб. в месяц. Рост цен ежемесячно увеличивает расходы на 3%. Составьте программу расчета необходимой суммы денег, которую надо единовременно попросить у родителей или дополнительно заработать, чтобы можно было прожить учебный год (10 месяцев), используя только эти деньги и стипендию.

$$\sin x + \sin x^2 + \ldots + \sin x^n$$

6) Дано действительное х. Вычислить:

7) Даны натуральное n, действительное x. Вычислить:

 $S = 1 \cdot 2 + 2 \cdot 3 \cdot 4 + \ldots + n \cdot (n+1) \times \ldots \times 2n$ 

8) Дано натуральное п. Вычислить:

$$P = (1 - \frac{1}{2})(1 - \frac{1}{4})(1 - \frac{1}{6}) \times \dots \times (1 - \frac{1}{2n})$$

9) Дано натуральное число n. Вычислить:

$$P = (1 - \frac{1}{2^2})(1 - \frac{1}{3^2}) \times \ldots \times (1 - \frac{1}{n^2}), \text{ obe } n > 2$$

10) Создать процедуру пользователя, которая определяет, в каких двузначных числах удвоенная сумма цифр равна их произведению. Эти числа необходимо показать в окне вывода.

11) Определить, существуют ли такие четыре последовательных натуральных числа, сумма квадратов которых равна сумме квадратов трех следующих натуральных чисел.

12) Найти все трехзначные числа, средняя цифра которых равна сумме двух крайних.

13) Найти первую степень числа 3, превышающую введенное пользователем число К.

14) Пользователь в диалоговое окно вводит целые положительные числа. До тех пор, пока числа меньше 100, вычисляется квадратный корень из них, если же значение введенного числа превышает 100, то работа программы прекращается.

15) Проверить, содержит ли квадрат данного натурального числа n, меньшего 10, цифру 3 в своей записи.

#### 2.9.4 Вопросы для самоконтроля

- 1) Опишите синтаксис оператора *Do... While... Loop*. В каких случаях он используется?
- 2) Опишите синтаксис оператора *Do... Until... Loop*. В каких случаях он используется?
- 3) Опишите синтаксис оператора *For... Next.* В каких случаях он используется?
- 4) Приведите примеры программ, использующих в своем решении цикл *Do... While... Loop*.
- 5) Приведите примеры программ, использующих в своем решении цикл *Do... Until... Loop*.
- 6) Приведите примеры программ, использующих в своем решении цикл *For... Next.*

## 2.10 Обработка массивов в VBA

#### 2.10.1 Описание и работа с массивами

Массив – это упорядоченный набор данных одного типа, снабженных индексами.

Индексы предназначены для нахождения определенного элемента массива, т.е. положение каждого элемента в массиве определяется его индексом. В этом и заключается упорядоченность. Тип компонент называется базовым типом массива.

В VBA массив рассматривается как переменная структурированного типа. Массиву присваивается имя, посредством которого можно ссылаться на

него, как на единое целое, так и на любую из его компонент. Переменная с индексом — идентификатор компоненты массива. Формат записи:

*<имя массива> (<индекс>),* где индекс может быть выражением порядкового типа. Индексы принято указывать в круглых скобках после имени массива. По умолчанию нумерация элементов массива начинается с 0. Говорят, что 0 – базовый индекс.

Например, если A – массив из пяти чисел: 2, 3, 7, 9, 6, то A (0) = 2 – первый элемент массива, A (4) = 6 – последний.

Описание массива определяет имя, размер массива, базовый тип и производится в разделе переменных. В VBA имеются следующие способы описания массивов.

Первый способ:

<имя массива> (<номер последнего элемента>) [As <тип>]

Второй способ:

<имя массива> (<начальный индекс> То <конечный индекс>) [As <тип>]

Отметим, что второй способ позволяет изменить базовый индекс. Другим способом изменения базового индекса является использование оператора **Option Base**, который имеет следующий синтаксис:

*Option Base <базовый индекс>* 

*<базовый индекс>* – единица или ноль.

Этот оператор применяется перед процедурой, аналогично оператору Option Explicit. Например, для изменения базового индекса с 0 на 1 используется оператор Option Base 1.

Массив А, который был рассмотрен, требует одного индекса для указания любого элемента. Такой массив называется одномерным (или линейным) В одномерных массивах хранятся значения линейных таблиц. Примеры описания одномерных массивов:

Dim A (12) As Byte Dim A (1 To 12) As Byte Dim Bin (5) As Integer Dim Str\_mass (4) As String Dim K (7) Dim L (0 To 3)

Заполнение массива в программе производится поэлементно. Чаще всего для этого используется цикл с параметром, где в качестве параметра применяется индексная переменная. Однако возможно заполнение массива и путем простого присвоения значения элементам:

Dim B (1 To 3) As Integer B (1) = 2

B(2) = 18

B(3) = 6

Удобным способом определения одномерных массивов является функция Array, преобразующая список элементов, разделенных запятыми, в вектор из этих значений, и присваивающая их переменной типа Variant.

Dim A As Variant

A = Array (10, 20, 35, 70)

Иногда в процессе выполнения программы требуется изменять размер массива. В этом случае первоначально массив объявляют как динамический. Для этого при объявлении массива не указывают его размерность. Например,

Dim R () As Single

Пусть, например, в программе следует вычислить необходимый размер массива и связать его с некоторой переменной, например, n; затем изменить размер динамического массива можно с помощью оператора ReDim:

## ReDim [Preserve] Имя (<номер последнего элемента>) [As <тип>]

или

## ReDim [Preserve] Имя (<начальный индекс> То <конечный индекс>) [As <тип>]

Preserve – ключевое слово, используемое для сохранения данных в существующем массиве при изменении значения последней размерности.

Двумерный массив — структура данных, хранящая таблицу. В таблице каждый элемент определяется номером строки и номером столбца, на пересечении которых он расположен.

В VBA двумерный массив может быть описан одним из следующих способов:

Первый способ:

#### <имя массива> (<n1>, <n2>) [As <тип>]

где <n1>, <n2> – номер последнего элемента строки и номер последнего элемента строки соответственно;

Второй способ:

## <имя массива> (<k1> То <k2>, <m1> То <m2>) [As <тип>]

где <k1>, <k2> – начальный и конечный индексы элементов строки; <m1>, <m2> – начальный и конечный индексы элементов столбца.

Например, Dim mass (5, 5) As Integer Dim mass (1 To 5, 1 To 5) As Integer
Эти примеры описывают один и тот же массив, при условии, что был использован оператор Option Base 1.

Элементы двумерного массива идентифицируются переменными с двумя индексами. Например, М (3, 5). Обычно первый индекс связывают с номером строки, второй — с номером столбца матрицы.

#### 2.10.2 Примеры решения задач

Рассмотрим примеры.

Задача 1. Создать макрос, осуществляющей ввод десяти элементов целочисленного массива А, формирование строки элементов массива и вывод его на экран а также подсчитывающий сумму всех элементов данного массива с выводом ее в ячейку Excel.

```
🪜 Книга1 - Module1 (Code)
(General)
                                                        Mass
                                                     •
    Sub Mass()
   Dim A(1 To 10) As Integer, i As Integer, Str As String, s As Integer
   s = 0
   Str = " "
   For i = 1 To 10
   A(i) = Val(InputBox("Введите " & i & "-ый элемент массива", "Заполнение массива"))
   Str = Str & A(i) & " "
   s = s + \lambda(i)
   Next
   MsgBox Str
   Лист1.Range("a1") = "сумма"
   Лист1.Range("b1") = s
   End Sub
```

Рисунок 75 – Код макроса для задачи 1

<b>1</b>	licrosoft Exe	cel-Книга1							
:	🗐 Файл Правка Вид Вставка Формат Сервис Данные Окно Справка								
1	💕 🛃 🛛	6 6	🛕   💝 🛍	1 X 🗈	🖺 • 🝼	19 - (1 -	😫 Σ 🕶	A↓ A↓	
Ari	al Cyr	<del>+</del> 10	- ж А	″ <u>ч</u>   ≣	≣ ≣ ₫	9% 0	00, 0, 0, 00)		
		•	<b>f≈</b> 55						
	A	В	С	D	E	F	G	Н	
1	сумма	55							
2			списо	к элементо	в массива	×			
3									
4			12	3456789	10				
5									
6				<u> </u>	K				
7				1					
8									
9									

Рисунок 76 – Результат работы макроса для задачи 1

Задача 2. Написать программу, содержащую базовые элементы программ для обработки массивов: заполнение случайными числами динамического массива, поиск наибольшего и наименьшего элементов, перестановка элементов, дописывание в имеющийся массив новых элементов.

**Option Explicit** 

Option Base 1

Sub Mass()

Dim M1() As Integer, M2() As Integer, n As Integer, i As Integer, max As Integer, min As Integer, \_\_

Str1 As String, Str2 As String, Str3 As String, sum As Integer, pro As Single, buf As Integer

n = InputBox("Введите количество элементов массива", "Определение размера массива")

ReDim M1(n), M2(n)

Заполнение массива случайными числами в диапазоне от 1 до 10 'и формирование строки значений элементов массива Randomize For i = 1 To n M1(i) = Int(10 \* Rnd + 1)M2(i) = Int(10 \* Rnd + 1)Str1 = Str1 & M1(i) & ""Str3 = Str3 & M2(i) & " " Next 'Поиск максимального элемента массива M1 max = M1(1)For i = 2 To n If M1(i) > max Then max = M1(i) Next 'Поиск минимального элемента массива M1 min = M1(1)For i = 2 To n If  $M1(i) < \min$  Then  $\min = M1(i)$ Next 'Поиск суммы элементов массива M1, стоящих на четных местах sum = 0For i = 2 To n Step 2 sum = sum + M1(i)Next 'Поиск произведения ненулевых элементов массива M1 pro = 1For i = 1 To n If  $M1(i) \Leftrightarrow 0$  Then pro = pro \* M1(i)

Next 'Поменяем местами 1-ый и 2-ой элементы массива M1 If  $n \ge 2$  Then buf = M1(1) M1(1) = M1(2) M1(2) = buf End If For i = 1 To n Str2 = Str2 & M1(i) & " " Next

MsgBox "Массив: " & Str1 & Chr(13) & "Максимальный элемент: " & max & Chr(13) & \_

"Минимальный элемент: " & min & Chr(13) & "Сумма элементов массива, стоящих на четных местах: "\_

& sum & Chr(13) & "Произведение ненулевых элементов массива: " & pro & Chr(13) \_

& "Массив после обмена 1-го и 2-го элементов: " & Str2

'Допишем в массив M2 максимальный и минимальный элементы масси-

ва Ml

```
ReDim Preserve M2(n + 2)
M2(n + 1) = max
M2(n + 2) = min
Str2 = ""
For i = 1 To n + 2
Str2 = Str2 & M2(i) & " "
Next
```

МsgBox "Первый массив: " & Str1 & Chr(13) & "Второй массив: " & Str3 & Chr(13) & \_

"Второй массив с приписанным максимумом и минимумом из первого: " & Chr(13) & Str2

End Sub

Задача 3. Пример заполнения двумерного динамического массива случайными целыми числами и вывод на экран.

Option Explicit Option Base 1 Private Sub Массив\_двумерный ()

Dim mass() As Integer, i As Integer, j As Integer, n As Integer, m As Integer, str As String

n = InputBox("Введите количество строк массива", "Размер массива")

m = InputBox("Введите количество столбцов массива", "Размер массива")

'Переопределение размерности массива ReDim mass(1 To n, 1 To m) As Integer Randomize

```
'Заполнение массива случайными значениями
'и накопление элементов массива для последующего вывода значений
For i = 1 To n
For j = 1 To m
mass(i, j) = Int(100 * Rnd + 1)
str = str & mass(i, j) & " "
Next j
str = str & Chr(13)
Next i
MsgBox "Массив: " & Chr(13) & str, , "Результат"
End Sub
```

#### Задача 4. Заполнить матрицу порядка п по следующему образцу:

3 ... n-2 n-11 2 n1 2 2 ... n-3 n-2 n-13  $\dots$  n-4 n-3 n-2 2 1 1 . ·. : : n-1 n-2 n-3 ... 2 1 2  $n \quad n-1 \quad n-2 \quad \cdots \quad 3$ 2 1

Идея алгоритма основана на двух свойствах этой матрицы: она симметрична относительно главной диагонали, т.е. A (i, j) = A(j, i) и элементы верхнего треугольника матрицы вычисляются по формуле A (i, j) = i - j + 1.

```
Option Explicit
Option Base 1
Sub 3anonhenue()
Dim A() As Integer, n As Integer, i As Integer, j As Integer, Str As String
n = InputBox("Введите порядок матрицы:", "Определение размера")
ReDim A(n, n) As Integer
Str = ""
'Вложенные циклы
For i = 1 To n
For i = 1 To n
A(i, j) = i - j + 1
A(j, i) = A(i, j)
Next
Next
'Формирование строки, содержащей элементы массива
For i = 1 To n
For i = 1 To n
Str = Str \& A(i, j) \& ""
Next
Str = Str \& Chr(13)
```

Next

MsgBox "Полученный массив: " & Chr(13) & Str,, "Результат" End Sub

Задача 5. Сформировать матрицу Пифагора (таблицу умножения в матричной форме) и вывести ее на экран.

Значения элементов матрицы Пифагора вычисляются следующим образом: Р (i, j) = i\*j. Вычисления и вывод матрицы производятся в двух вложенных циклах. Вывод на экран организуем в виде прямоугольной таблицы.

```
Option Explicit
Option Base 1
Private Sub Пифагор()
Dim P(1 To 9, 1 To 9) As Integer, i As Integer, j As Integer, Str As String
Str = ""
For i = 1 To 9
For j = 1 To 9
P(i, j) = i * j
Str = Str & P(i, j) & " "
Next
Str = Str & Chr(13)
Next
MsgBox "Матрица Пифагора: " & Chr(13) & Str, , "Результат"
```

End Sub

_	Результат 🔀	
_	Матрица Пифагора:	
	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
	ок	

Рисунок 77 – Результат работы программы задачи 5

Примечание. Chr(13) в данной программе является кодом символа «пробел»

## 2.10.3 Задания для самостоятельной работы.

1) Известны результаты 5 студентов по итогам экзаменов по химии и информатике. Найти фамилии студентов, сдавших оба экзамена на отлично.

2) Пусть задана матрица А из действительных чисел размера 3х4. Найти наибольший элемент во второй строке данной матрицы.

3) Найти среднее арифметическое всех элементов массива, состоящего из 10 действительных чисел.

4) Найти наименьший целое число из 7 целых чисел.

5) Найти количество положительных элементов массива A(5x5).

6) Найти количество отрицательных элементов массива B(2x3).

7) Определить, сколько раз встречается число 7 среди элементов массива M, состоящего из 15 целых чисел.

8) Определить, сколько элементов массива С(5х4) меньше, чем число 6.

9) Определить, сколько элементов массива А(4х3) больше, чем число 3.

- 10) Найти сумму всех неотрицательных элементов массива.
- 11) Найти наименьшее из чисел a<sub>1</sub>, 2a<sub>2</sub>, 3a<sub>3</sub>,..., 8a<sub>8</sub>.
- 12) Найти наибольшее из чисел 2a<sub>1</sub>, 3a<sub>2</sub>, 4a<sub>3</sub>,..., 9a<sub>8</sub>.
- 13) Найти сумму  $a_1+2a_2+3a_3+\ldots+8a_8$ .
- 14) Найти наименьший по модулю элемент массива А(5х6).

15) Найти разность между наибольшим и наименьшим элементами массива А, стоящего из 7 действительных чисел.

16) Отсортировать массив С, состоящий из 5 целых чисел, в порядке убывания элементов.

17) Найти произведение отрицательных элементов массива D(4x4).

18) Найти сумму всех элементов массива A(3x5), имеющих четные индексы.

19) Найти наибольший из элементов массива C(2x6), имеющих нечетные индексы.

20) Найти среднее арифметическое всех положительных элементов массива H, состоящего из 11 действительных чисел.

21) Найти среднее арифметическое всех отрицательных элементов массива К(3х3).

22) Найти сумму элементов массива Р(4х2), превышающих число 5.

#### 2.10.4 Вопросы для самоконтроля

- 1) Дайте определение массива.
- 2) Какие массивы называются одномерными? Приведите примеры одномерных массивов.
- 3) Какие массивы называются двумерными? Приведите примеры одномерных массивов.
- 4) Как описать массив в VBA?
- 5) Как можно заполнить массив элементами?

- 6) Какое действие выполняется при применении оператора *ReDim?*
- 7) Приведите пример программы, использующей массив в своем решении.

## 2.11 Работа со строками

#### 2.11.1 Строковые выражения и строковые функции

Строковые выражения, это выражения, возвращающие строковый тип данных. Для хранения строковых выражений в VBA используется два типа данных: String и Variant. Над строками можно выполнять операцию сложения (склеивания, конкатенации). Такая операция обозначается символом &. Например, при выполнении следующего макроса, будет результат, показанный на рисунке 78.

Private Sub skleivanie() d = "V" & "B" & "A" MsgBox d End Sub

Microsoft Excel	×
VBA	
ОК	

Рисунок 78 – Результат работы макроса skleivanie

К переменным и выражениям строкового типа можно применять различные функции. В таблице 13 приведены все функции VBA, предназначенные для строк.

Функция	Возвращаемое значение
1	2
Asc(string)	ANSI-код первого символа в строке string
AscB(string)	Байт, соответствующий первому символу в строке string
AscW(string)	Код символа Unicode, соответствующий первому симво-
	лу в строке string
Chr(charcode)	Символ, соответствующий ANSI-коду charcode (тип ре-
	зультата — Variant)

Таблица 13 – Строковые функции VBA

Продолжение таблицы 13

1	2
Chr\$(charcode)	Символ, соответствующий ANSI-коду charcode (тип ре-
	зультата — String)
ChrB(charcode)	Байт, соответствующий ANSI-коду, заданному аргумен-
	том charcode
ChrW(cbarcode)	Символ Unicode, который соответствует ANSI-коду
	charcode
Filter(sourcearray,	Из заданного массива строк (sourcearray) возвращает
match,)	подмножество строк (т.е. другой массив), которые удо-
	влетворяют критерию match
InStr(start, stringl,	Позиция первого появления символа подстроки string2 в
string2)	строке stringl, начиная с позиции start
InStrB(start, Btringl,	Номер байта первого появления подстроки string? в
String2)	строке atringl, начиная с позиции start
inStrRev (,	Позиция символа при первом появлении подстроки
stringbatcn, start)	stringcheck, В строке stringmatch, начиная с позиции
	start (функция проверяет строку с конца до начала)
LCase(atring)	Преобразует символы строки string в нижний регистр
	(тип результата — Variant)
LCase\$(string)	Преобразует символы строки string в нижний регистр
	(тип результата — string)
Left(string, length)	Из строки string возвращает length символов, начиная с
	крайнего слева символа (тип результата — Variant)
Left\$(string, length)	Из строки string возвращает length символов, начиная с
	крайнего слева символа (тип результата — string)
LeftB(string, length)	Из строки string возвращает length байт, начиная слева
	(тип результата — variant)
LeftB\$(string, length)	Из строки string возвращает length байт, начиная слева
	(тип результата — string)
Len (string)	Количество символов в строке string
LTrim(string)	Возвращает строку string без пробелов в начале строки
	(тип результата — variant)
LTrim\$(string)	Возвращает строку string без пробелов в начале строки
	(тип результата — string)
Mid(string,	Возвращает 1 ength символов из строки string, начиная с
start,length)	позиции start (тип результата — variant)
Mid\$(string, start,	Возвращает length символов из строки string, начиная с
length)	позиции start (тип результата — string)
Right(string, lengtn)	Возвращает length символов справа из строки string (тип
	результата — variant)
Right\$(string, length)	Возвращает lengtn символов справа из строки string (тип
	результата — string)

Продолжение таблицы 13

1	2
RightB(string,	Возвращает lengtn байт справа из строки string (тип ре-
length)	зультата — Variant)
RightB\$(string,	Возвращает lengtn байт справа из строки string (тип ре-
length)	зультата — String)
RTrim( string)	Возвращает строку string без пробелов в конце строки
	(тип результата — variant)
RTrim\$(string)	Возвращает строку string без пробелов в конце строки
	(тип результата — string)
Space(number)	Возвращает строку длиной number символов, заполнен-
	ную пробелами (тип результата — variant)
Space\$(number)	Возвращает строку длиной number символов, заполнен-
	ную пробелами (тип результата — string)
Split(expression, de-	Возвращает массив, состоящий из подстрок строки ех-
limiter)	pression, разъединенных разделителем delimiter
Str(number)	Строковое представление числа number (тип результата
	— variant)
Str\$(number)	Строковое представление числа number (тип результата
	— String)
StrComp(stringl,	Результат сравнения строк stringl и string
string2, compare)	
String(number, charac-	Возвращает символ character, который повторяет number
ter)	раз (тип результата — variant)
StrReverse(expression)	Возвращает строку, состоящую из символов строки ех-
	pression, записанных в обратном порядке
String\$(number, char-	Возвращает символ character, который повторяет number
acter)	раз (тип результата — string)
Trim(string)	Возвращает строку string без пробелов в начале и в кон-
	це строки (тип результата — variant)
Trim\$(string)	Возвращает строку string без пробелов в начале и в кон-
	це строки (тип результата — string)
UCase(string)	Возвращает строку string, преобразованную в верхи,-
	регистр (тип результата — Variant)
UCase\$(string)	Возвращает строку string, преобразованную в верхи/,
	регистр (тип результата — string)
Val(string)	Возвращает число, которое содержится в строке string

# 2.11.2 Пример решения задачи, использующей в решении строки

Задача. Из строки, в которой записаны фамилия и имя человека, извлечь имя.

Решим эту задачу, используя:

а) пользовательскую функцию,б) процедуру пользователя

Задача 1, а) в окне модуля запишем программный код функции Имя. Function Имя(Фамилия\_Имя) Dim пробел As Integer пробел = InStr(Фамилия\_Имя, "") Имя = Mid\$(Фамилия\_Имя, пробел + 1, Len(Фамилия\_Имя) - пробел) End Function

Перейдем в Excel, и через команду Вставка – Функция, в диалоговом окне Мастера функций в категории Определенные пользователем, найдем функцию Имя. Появится диалоговое окно (рисунок 79), в котором необходимо вписать любые фамилию и имя.

Аргументы функции			×
Имя	Фамилия_Имя Иванов Иван	<u>.</u> =	
Справка недоступна.		=	
	Фамилия_Имя		
 Справка по этой функции	і Значение:	ОК От	мена

Рисунок 79 – Диалоговое окно Аргументы функции

После нажатия кнопки ОК в выделенную ячейку электронных таблиц будет введено только имя от введенного аргумента (рисунок 80)

M	licrosoft	Excel - R	(нига2						
:	<u>Ф</u> айл	Правка	<u>В</u> ид	Вст <u>а</u> вка	Фор <u>м</u> ат С	ервис Дан	іные <u>О</u> кно	<u>С</u> правка	
1	🚰 🚽	l 🖪 🔒		🗟   💞 🛍	1 🖌 🗈	🖺 • 🝼	<b>v) -</b> (°' -	😫 Σ 🛛	А Я
Ari	al Cyr		<b>-</b> 10	- Ж	К П  ≣	≣ ≣ ∎	1 🕎 % 0	00, 0, 00, 00	¥
	C2	-		<i>f</i> x =Имя("	Иванов Ив	ан")			
	A		В	С	D	E	F	G	
1									
2				Иван	1				
3									
4									
5									
6									

Рисунок 80 - Результат работы пользовательской функции Имя

Задача 1, б) Создадим для решения этой задачи пользовательскую процедуру (рисунок 81). Ввод данных показан на рисунке 82, а результат работы на рисунке 28.

odule2 (Code)]	
<u>D</u> ebug <u>R</u> un <u>I</u> ools <u>A</u> dd-Ins <u>W</u> indow <u>H</u> elp	
🝽 🗼 💵 🔤 🔡 🚰 🚰 🎌 🞯 🖬 Ln 9, Col 1 🔤 💂	
(General)	▼ rew
Sub rew() Dim пробел As Integer, a As String, Имя As String a = InputBox("Введите фамилию и имя", "Ввод аргументов") пробел = InStr(a, " ") Имя = Mid\$(a, пробел + 1, Len(a) - пробел) MsgBox Имя, 0, "Имя человека" End Sub	

Рисунок 81 – Процедура пользователя

Ввод аргументов			×
Введите фамилию и и	1M9		OK Cancel
Иванов Иван			

Рисунок 82- Диалоговое окно ввода аргументов

Имя	і человека 🛛 🗙
И	ван
	ОК

Рисунок 83 – Окно вывода результата

#### 2.11.3 Задачи для самостоятельной работы

- 1) Создайте макрос, вычеркивающий из слова все русские буквы К.
- 2) Создайте макрос, который преобразовывает все символы строки в символы верхнего регистра
- 3) Создайте макрос, который проверяет, есть ли в выражении буквы «b» и, если есть, то на какой позиции стоит первая буква.
- 4) Определить, имеются ли в выражении две одинаковые буквы, идущие подряд.
- 5) Определить, является ли введенное выражение палиндромом.
- 6) Создайте макрос, вычеркивающий из строки символ «а».
- 7) Создайте макрос, удваивающий символы, стоящие на четных местах.

## 2.12 Управление свойствами объектов VBA

#### 2.12.1 Общие сведения

В VBA реализована модель объектно-ориентированного событийно управляемого программирования. В такой модели предполагается наличие некоего блока, реализующего ответ на произошедшие события. Эти события могут быть порождены операционной системой (системные события) или действиями оператора (пользователя). Таким образом, выполнение событийно – управляемого приложения не идет предопределенным путем, а зависит от сложившейся ситуации.

Важным компонентом VBA является наличие библиотек типовых объектов. Эти объекты можно вставлять в программу без каких-либо изменений или изменив некоторые их свойства. В VBA свойства большинства объектов представлены в окне свойств **Properties** и легко доступны для изменений.

Видимые или визуальные объекты VBA используются для создания средства общения оператора (пользователя) и приложения. Это средство принято называть пользовательским интерфейсом. Большинство объектов служат для организации ввода данных в программу, вывода данных на печатающие устройства или экран, для изменения хода работы программы. В конечном счете каждый из объектов передает в приложение сообщение о произошедшем событии – нажатии клавиши, щелчке мыши, т.е. о действии оператора. В связи с этим их называют еще элементами управления. Это, например, командные кнопки, клавиши, текстовые поля, рамки и окна для рисунков и пр. Элементы управления, расположенные на панели объектов (Рисунок 7) называют стандартными. Стандартные объекты поддерживают технологию ActiveX, ранее называвшуюся OLE технологией.

Каждый из объектов обладает определенным наборов свойств, изменяя которые мы можем воздействовать на эти объекты и, тем самым, использовать их для назначенной цели. Свойства объектов можно разбить на несколько групп. Одни из свойств управляют внешним видом самих объектов - высота **Height**, ширина **Width**, цвет **BackColor** и **ForeColor**, шрифт надписи и т.д. Другие свойства управляют расположением объектов на окне форм: **Top** - расстояние от верхнего края формы, **Left** - расстояние от левого края, взаимным расположением на форме и т.д.

Рассмотрим работу со свойствами объектов на примере следующего задания.

#### Задание 1. Работа со свойствами окна форм

1) <u>Выполните команду Inset-Userform</u>. После этого действия на экране должно появиться окно форм. Это окно – один из видимых элементов Visual Basic, оно послужит в дальнейшем фоном для размещения элементов управления.

Как все визуальные объекты, окно форм обладает определенным набором свойств Properties. Свойства, перечисленные в окне **Properties**, поддаются изменению. Изменения производятся приданием свойствам некоторых новых значений. Убедитесь, что в окне **Properties** выбрана вкладка **Alphabetic** (Алфавитный).

2) <u>Измените заголовок окна Caption</u>. Для этого в окне свойств **Proper**ties нужно найти слово Caption и щелкнуть по нему мышью. Заголовок окна UserForm1 в правом поле строки Caption подсветится после этого синим прямоугольником. Это означает, что выделенное слово подготовлено к удалению. Прямо поверх старого заголовка UserForm1 можно набрать новый - Форма1. Заголовок на самом окне форм должен после этого измениться на Форма1. Аналогично можно изменить и имя окна Name.

3). <u>Измените цвет фона</u>. Для этого нужно выбрать свойство **BackColor** и дважды щелкнуть по нему мышью. Все действия выполняются аналогично предыдущему пункту. Из списка возможных значений фона, можно выбрать поочередно понравившиеся цвета и оттенки, одновременно наблюдя за изменениями в окне форм. Цвет фона не должен быть слишком ярким, иначе трудно будет разглядеть детали объектов, помещенных на окне форм.

4) <u>Измените свойство Font (Шрифт)</u>. Найдите свойство Font в окне Properties и щелкните по нему мышью. В правом поле строки Font появится кнопка с многоточием (тремя точками). Щелчок на этой кнопке откроет стандартное диалоговое окно Windows под названием Выбор шрифта. Можете выбрать любой шрифт - никаких изменений в окне форм не произойдет. Шрифт предназначен для выполнения надписей на объектах, которые будут в дальнейшем расположены на окне форм.

5) <u>Измените свойство Picture (Рисунок).</u> Щелкните мышью на названии свойства Picture, в открывшемся диалоговом окне поиска файлов выберите любую папку с рисунками, в ней найдите любой файл с рисунком и нажмите кнопку ОК. Рисунок установится фоном окна форм. Изменяя значения свойств **PictureAlignment** (расположение рисунка) и **PictureSizeMode** (размер рисунка), можно добиться нужно размера и положения рисунка на окне форм.

6) Свойства Wigth и Height позволяют задать размер формы в типах.

## 2.12.2 Элементы управления панели Toolbox

На стандартной палитре объектов размещено 15 кнопок с изображениями объектов, предназначенных для управления работой проекта. Щелчок мышью по кнопке делает ее активной.

Поместить объект на окно форм можно двумя способами:

1) Дважды щелкнув мышью на кнопке панели. При этом все объекты помещаются в одно и то же место - в центр окна форм, сохраняются стандартные размеры всех объектов.

2) Выбрать при помощи курсора мыши кнопку панели, затем переместить курсор на нужное место окна форм, и, удерживая левую клавишу мыши нажатой (при этом появится «резиновый» прямоугольник), отметить область, в которую будет помещен создаваемый объект. Размер объекта устанавливается произвольно.

*Внимание:* на экране может быть размещено несколько видимых объектов – окон, рамок, кнопок. Один из объектов (это может быть и окно форм!) обязательно является активным. Окно свойств **Properties** всегда содержит свойства того объекта, который находится в активном состоянии.

Для удаления видимого элемента с окна форм, выберите его, щелкнув мышью (элемент окажется помеченным 8-ю черными квадратиками), затем щелкните правой кнопкой мыши – появится контекстное меню, выберите Cut (Вырезать) и щелкните левой кнопкой мыши.

Рассмотрим назначение и способы работы с некоторыми кнопками панели **Toolbox** 

Указатель (Pointer) изображен в левом верхнем углу панели. Назначение указателя - дезактивировать все кнопки панели.

Метка. Выберите па панели элементов кнопку Метка (Label) и, поместив курсор мыши в левой верхней части окна форм, создайте там видимый элемент. Измените заголовок элемента Caption, введя любой текст, например "Hallo, Basic!". Измените шрифт надписи. Для этого в окне Properties отыщите свойство Font и щелкните мышью по прямоугольнику с тремя точками. В появившемся стандартном диалоговом окне выбора шрифтов выберите шрифт Times New Roman полужирный, наклонный, размером 24. Возможно, что он не поместится в прямоугольнике, отведенном под видимый элемент. Измените значение свойства AutoSize (автоматическое изменение размера) на True. Подберите цвет фона BackColor и цвет шрифта ForeColor. Текст метки можно изменить, только изменяя значение свойства Caption – Заголовок. Метка дает возможность создать на форме пояснительную надпись. Другое название объекта Label - этикетка. Командная клавиша (CommandButton) - один из наиболее часто используемых объектов. Практически каждое диалоговое окно содержит клавиши типа "Пуск", "Выход", "Отмена" и т.п. Установите на окне форм 2 три командных клавиши. Измените по своему усмотрению их размеры и заголовки Caption. Попытайтесь изменить цвет фона и цвет шрифта.

Рамка и Переключатель. Очистите окно форм, поместите на него и растяните объект Frame (Рамка), внутри рамки поместите одно под другим изображения трех кнопок переключателя **OptionButton**, присвойте им следующие заголовки: **Вперед**, **Стоп**, **Назад**. Заголовку рамки присвойте значение **Реверс**. Измените фон рамки и кнопок, сделайте их одинакового цвета. Цвет заголовков сделайте разным: Вперед - зеленый, Назад - красный, Стоп - желтый. Шрифт (Font) заголовков выберите по своему усмотрению.

Вы сконструировали объект управления - реверсивный пускатель. В таких устройствах обязательно должна присутствовать защита (блокировка) от одновременного включения двух двигателей вперед и назад - это неизбежно привело бы к аварии.

Для того, чтобы проверить как действуют кнопки в рабочем состоянии, нажмите клавишу [F5]. Тем самым Вы перейдете в режим выполнения проекта. Поочередно выберите кнопки переключателя, щелкая на них мышью. Вы быстро убедитесь, что из кнопок **OptionButton** выбрать можно только одну.

Флажки. Очистите окно форм. Поместите в верхней части окна метку Label с текстом «Я знаю языки», выберите подходящий шрифт и размер. Ниже поместите одно под другим четыре изображения независимой кнопки (флажка) CheckBox и присвойте им заголовки Русский, Английский, Французский, Японский. Нажмите [F5] и пометьте щелчком любые кнопки. Вывод: кнопки CheckBox могут быть выбраны в любой комбинации.

Редактор текста. Разместите на окне форм объект TextBox – окно для текста или редактор текста. Нажав клавишу [F5], перейдите в режим выполнения проекта. Попытайтесь ввести какой-нибудь текст, например список фамилий и изменять его произвольным способом. Если записать список Вам не удалось, вернитесь в режим конструирования, выберите свойство MultiLine (Многострочный) и придайте ему значение True. Повторите ввод текста.

Полосы прокрутки. Продолжим работу с рамкой TextBox. Установите свойства MultiLine=True, Scrollbars=0, перейдите в режим выполнения проекта, затем ведите несколько строк текста. Можно просто нажать на клавишу с любой буквой и держать ее, пока все окно не заполнится изображениями этой буквы. Смените букву и Вы увидите, что строки текста исчезают, теряются за пределами окна. Вернитесь в режим конструирования и присвойте свойству ScrollBars объекта TextBox значение 3 (Обе). Вновь перейдите в режим выполнения проекта. Попробуйте ввести длинную строку, уходящую за пределы текстового окна, затем, нажимая поочередно какую-либо из алфавитных клавиш и клавишу [Enter], введите длинный столбец символов. Опишите изменения, происходящие с полосами прокрутки. Проделайте то же самое при значениях Scrollbars = 1 и Scrollbars = 2.

## 2.13 Списки в VBA. Создание программного кода для подсчета стоимости недвижимости

# 2.13.1 Виды списков в VBA

Списки, размещаемые в формах Visual Basic for Application, позволяют пользователю выбрать один из возможных вариантов ответа. Для создания списков используются две кнопки на панели элементов управления:

**СотвоВох** - создает в форме раскрывающийся список, представляющий собой объект типа **СотвоВох.** Этот объект позволяет пользователю осуществлять выбор вводимого в размещаемое сверху поле ввода, выбирать значение из списка, открываемого нажатием кнопки со стрелкой, размещаемой с правой стороны. Список данного типа удобно использовать в том случае, если вводимых значений много, а места в форме для расположения обычного списка не хватает

**ListBox.** Создает в форме список, в котором элементы расположены в одну или несколько колонок. В случае, если элементы списка не помещаются в созданном объекте **ListBox**, в нем появляются полосы прокрутки, располагаемые снизу и/или с правой стороны.

На рисунке 84 показаны списки, представляющие собой элементы List-Box (обозначен как «список») и ComboBox (обозначен как «раскрывающийся список»).



Рисунок 84 – Списки на форме

Чтобы разместить значения в списке во время разработки, используется специальный метод AddItem, применяемый к объектам ListBox и Combobox. Формат записи данного метода такой:

ComboBox. AddItem "значение элемента списка" или ListBox. AddItem "значение элемента списка"

Например, результатом работы следующего программного кода станут списки, показанные на рисунке 85.

Private Sub CommandButton1\_Click() ComboBox1.AddItem "крыша" ComboBox1.AddItem "черепица" ComboBox1.AddItem "шифер" ComboBox1.AddItem "оцинкованный лист" ComboBox1.AddItem "рубероид" ListBox1.AddItem "крыша" ListBox1.AddItem "черепица " ListBox1.AddItem "шифер" ListBox1.AddItem "шифер" ListBox1.AddItem "рубероид" End Sub



Рисунок 85 - Два типа списков, используемых в формах

Значения элементов списка можно также брать из ячеек электронных таблиц. В приведенном ниже примере показано, как в список ListBox1 добавляются элементы, взятые из первых трех строк столбца А электронных таблиц Excel.

Private Sub CommandButton1\_Click()

Dim s As Integer, s1 As Integer, s2 As Integer, p As Integer, d As String, a As String

```
For p = 1 To 3
d = "A" & Trim(Str(p))
MsgBox d
s = Лист1.Range(d).Value
ListBox1.AddItem s
Next
End Sub
```

Примечание: здесь функция Trim(Str(p)) убирает пробелы справа и слева от каждого значения p, переведенного в символьный вид, для того, чтоб правильно сформировать адрес ячейки в виде номера столбца и номера строки и подставить его затем в качестве значения аргумента в метод Range.

Первый элемент раскрывающего списка отображается пустым. Для того, чтобы он отображался непустым, необходимо присвоить нужное значение свойству **Text**.

Для списков существуют также встроенные методы, позволяющие во время выполнения приложения удалять элементы списка (Removeltem) или полностью очищать список (Clear).

# 2.13.2 Создание программного кода для подсчета стоимости недвижимости

Итак, мы рассмотрели наиболее часто встречающиеся при создании приложений элементы управления, а также увидели, как можно, управляя их свойствами, создавать простые окна. Рассмотрим создание программного кода, включающего в себя не только простое изменение свойств, но также и функции и операции над объектами. На основе введенных данных (длина и ширина дома, количество этажей) будет производится оценочная стоимость строящегося дачного домика. Кроме того, на форме будет выведена информация о материале, из которого будет строиться дачный дом, материале, используемом для крыши, и сведения о том, будет ли у дома мансарда. Эта информация также будет учитываться при подсчете стоимости.

Задача. Дачный домик имеет форму параллелепипеда и крышу. Рассчитать стоимость дачного домика, учитывая, что 1 м<sup>2</sup> стоит 20000 рублей. Если крыша дома сделана из черепицы, то ее стоимость 20000 рублей, из шифера – 15000 рублей, из рубероида – 25000 рублей. Наличие мансарды увеличивает стоимость дома на 30 тысяч рублей.

На рисунке 84 показано окно форм для данной процедуры и из него видно, что форма содержит:

- 1) текстовую информацию пояснительного характера;
- 2) текстовые поля для ввода длины, ширины дома и количества этажей;
- флажок, используемый дли указания, будет ли дом содержать мансарду
- 4) список, позволяющий выбрать используемый кровельный материал,
- 5) раскрывающийся список для выбора материала, из которого будут строиться стены;
- кнопка, при нажатии на которую будут производиться расчет стоимости дачного дома;
- 7) поле, размещенное в нижней части формы с правой стороны кнопки, предназначено для отображения вычисленной стоимости.

Для создания процедуры по расчету стоимости домика, выполните следующие действия:

1) Выполните команду Insert-UserForm

2) Увеличьте размер формы так, чтобы в ней поместились все элементы управления. Для этого установите курсор в нижний правый угол и, когда он примет вид двунаправленной стрелки, перемещайте его вниз и вправо получения необходимого размера.

Расчет стоимости	дачного домика		x						
Сто	Стоимость дачного домика								
Длина	1	Ширина							
Количество этажей		🗆 Мансарда							
Крыша 	Брус 10x15 Брус 15x20 Брус 10x10	Стены шифер 💌							
Стоимость		Рассчитать стоимость Завершить							

Рисунок 84 Форма приложения, позволяющего производить вычисления на основе введенных данных

3) Свойству Caption формы присвойте следующее значение: *Расчет* стоимости дачного домика.

4) Разместите на форме элементы управления класса Label, TextBox, Combobox, ListBox, CheckBox, присвойте их свойствам значения, соответствующие значениям на рисунке 84. Элементы раскрывающего списка показаны на рисунке 85.

Расчет стоимости ,	дачного домика		×	
Стоимость дачного домика				
Длина	2	Ширина 3		
Количество этажей	3	🗹 Мансарда		
Крыша	Брус 10x15 Брус 15x20 Брус 10x10	Стены нерепица - черепица - шифер руберониа		
Стоимость		Рассчитать стоимость Завершить		

Рисунок 85 – Окно приложения с раскрытым списком СотвоВох

5) На этом этапе необходимо задать процедуру обработки Click (щелчок) для объекта CommandBotton1. Двойным щелчком мыши щелкните по объекту CommandBotton1 и запишите в обработку этого события расчет стоимости дома в виде, показанном на рисунке 86.

a Microsoft Visual Basic - КнигаЗ.xls - [UserForm3 (Code)]				
III File Edit <u>Vi</u> ew Insert F <u>o</u> rmat <u>D</u> ebug <u>R</u> un <u>T</u> ools <u>A</u> dd-Ins <u>W</u> indow <u>H</u> elp				
N 19 - N X 19 (B AB 19 (C ) ) II II V (S A 19 77 (C ) )				
Project - VBAProject 🛛 🗶	UserForm Click			
- 🧏 VBAProject (Книга3.xls)	Private Sub CommandButton1 (lick()			
□ · · · · · · · · · · · · · · · · · · ·	Dim a As Single, b As Single, c As Single, d As Single, f As Single			
	b = Val(TextBox1.Text)			
	c = Val(TextBox2.Text)			
Пист3 (Лист3)	d = Val(TextBox3.Text)			
— Такнига	If ComboBox1.Text = "черепица" Then			
E Corms	a = 20000 Florid CarboRevil Taut = "wetter" There			
Properties - CommandButten1	a = 15000			
Fropercies - Commandbucconi	Else: a = 25000			
	End If			
Alphabetic Categorized	If CheckBox1.Value = "False" Then			
(Name) CommandButto 🔺	f = (b * c * 20000 * d) + a			
Accelerator	Else: $f = (b * c * 20000 * d) + a + 30000$			
AutoSize False	End If			
BackColor BackColor	TextBox4. Text - Str(I)			
BackStyle 1 - fmBackStyle	End Sub			
Caption Paccyutate ctc	Private Sub CommandButton2 Click()			
ControlTipText	End			
Default False	End Sub			
Enabled True				
Font Tahoma	Private Sub UserForm_Activate()			
ForeColor & 8H80000012	ComboBox1.AddItem "черепица" ComboBox1.hddItem "черепица"			
Height 24	ComboBox1.AddItem "HMMpep"			
HelpContextID U	ListBox1.AddItem "Envo: 10x15"			
Lert 190	ListBox1.AddItem "Epyc 15x20"			
MouseIcon (None)	ListBox1.AddItem "Bpyc 10x10"			
MousePointer 0 - fmMousePoin	End Sub			
Picture (None)				
PicturePosition 7 - fmPicturePos	Private Sub UserForm_Click()			
TabIndex 12	End Sub			
TabStop True				
TakeFoorusOpClid True				
Visible True				

Рисунок 86 – Программный код процедуры по расчету стоимости дачно-го домика

6) Создание приложения завершено. Запустите его на выполнение, нажав клавишу <F5>.

#### 2.13.3 Задания для самостоятельной работы

1) Модифицировать задачу 1 из пункта 10.2, добавив на форму и программный код вывод стоимости домика в евро по текущему курсу.

2) Модифицировать задачу 1 из пункта 10.2, добавив на форму и программный код вывод стоимости домика в долларах по текущему курсу.

3) Создать процедуру пользователя, подсчитывающую сумму за ЖКХ услуги за истекший месяц с владельцев квартиры, в которую входят: оплата за

газ, оплата за телефон, оплата за электроэнергию и оплата за квартиру. На форме должна располагаться информация о: количестве человек, прописанных в квартире, о количестве пенсионеров и инвалидов, о том, сколько было израсходовано электроэнергии за истекший месяц. Тарифные ставки по каждой услуге составляют: 1кВт стоит 30 копеек, оплата за газ на одного человека в месяц - 42 рубля, оплата за квартиру на одного человека в месяц – 600 рублей. Плата за телефон является абонентской и составляет 250 рублей в месяц. Если в квартире имеются пенсионеры, то скидка на все услуги – 10%, если имеются инвалиды – 40%.

4) Некоторый покупатель в магазине по продаже сотовых телефонов имеет скидку 8% на некоторые модели телефонов. К ним относятся: телефон фирмы Nokia по цене – 8000 рублей, Samsung по цене 7200 рублей, Alcatel по цене 4300 рублей, Motorola – 5000 рублей и LG по цене 5500 рублей. Телефон Fly по цене 1500 рублей продается всем покупателям без скидки. Поместить на форму список с названиями фирм моделей телефонов и посчитать сумму к оплате со скидкой для этого покупателя.

5) Составить процедуру пользователя, рассчитывающую размер стипендии студента в зимнюю сессию, в зависимости от его оценок. Предметы: экономика, информатика, философия, иностранный язык. За каждый предмет возможна оценка от 2 до 5. Если студент сдал сессию на отлично, то его стипендия составит 2000 рублей, если имеется одна четверка, а остальные пятерки – 1500, две четверки и две пятерки – 1000 рублей, в противном случае студент стипендии получать не будет.

6) Составить процедуру для расчета заработной платы. Для работников некоторого предприятия, в зависимости от их стажа работы на этом предприятии, к заработной плате рассчитывается определенная надбавка. Если человек проработал на предприятии 6 лет, то ему добавляется 6 % от его заработной платы, если 10 лет – 10%, свыше 15 – 15%.

7) Дан список фигур – квадрат, треугольник, трапеция, ромб, прямоугольник. В зависимости от выбора пользователя запросить у него данные, необходимые для расчета площади выбранной фигуры и вывести на форму значение ее площади.

	A	В	С	
1		Отдел 1	Отдел 2	
2	Доходы отдела			
3	Расходы отдела			
4	Стоимость имущества			
5	Траснспортные расходы			
6	Коммунальные услуги			
7	Оплата персонала			
8	Реклама			
9	Всего расходов			
10	Прибыль			
11				
10				

8) В Excel дана следующая таблица

Заполните ее и создайте процедуру пользователя, которая берет данные из ячеек с доходами и расходами по каждому отделу, рассчитывает прибыль и выводит данные в ячейку с прибылью. К данной процедуре создайте форму, где отображены расходы, доходы и прибыль по каждому отделу.

9) Создайте проект, в котором пользователь выбирает из одного списка любую цифру от 2 до 9, а затем во второй список выводится таблица умножения на эту цифру.

10) Кредитор берет у банка кредит на сумму 100 000 рублей. У него имеются три варианта выбора срока и процентной ставки: под 20% годовых на 10 месяцев, под 15% годовых на год и под 12% на 18 месяцев. Выбрать наиболее выгодный вариант для кредитора.

#### 2.13.4 Вопросы для самоконтроля

- 1) Для чего предназначены объекты управления?
- 2) Какие свойства объектов управления вы знаете?
- 3) Перечислите свойства объекта UserForm, который вы знаете. На что они влияют?
- 4) Какую функцию выполняет объект Label? Перечислите его свойства.
- 5) Какую функцию выполняет объект **CommandButton**? Перечислите его свойства.
- 6) Какую функцию выполняет объект Frame?
- 7) Какую функцию выполняет объект **OptionButton**? Перечислите его свойства.
- 8) Какую функцию выполняет объект **CheckBox**? Перечислите его свойства.
- 9) Какую функцию выполняет объект **TextBox**? Перечислите его свойства.
- 10) Какую функцию выполняет объект Scrollbar? Перечислите его свойства.
- 11) Какую функцию выполняет объект **ComboBox**? Перечислите его свойства.
- 12) Какую функцию выполняет объект ListBox? Перечислите его свойства.

#### 2.14 Финансово-математические функции VBA

## 2.14.1 Пример решения задачи с использованием финансовоматематической функции

VBA имеет определенный набор встроенных функций. Среди них имеются финансово-математические функции, которые позволяют решить некоторые классы финансовых задач. С помощью этих функций, например, можно

вычислить размер месячного платежа по ссуде, будущего дохода или размер годовой амортизации имущества. Этих функций 13. Они требуют задания множества различных аргументов. В таблице 14 приведем описание тех аргументов, которые используются наиболее часто.

Название аргумента	Описание аргумента		
Rate	Процентная ставка, фиксированная для всего срока дей- ствия ссуды или инвестиции		
Nper	Общее число периодов выплат по ссуде или инвестиции		
Pmt	Размер периодической выплаты по ссуде или прибыли по депозиту		
Pv	Приведенная или основная стоимость стоимость ссуды или первоначальный вклад по инвестиции		
Fv	Будущая стоимость ссуды или инвестиции		
Туре	Тип выплат. Если аргумент равен 0 (по умолчанию), то выплаты производятся в конце каждого периода; если равен 1 – в начале периода		

Таблица 14 – Аргументы финансово-математических функций

При работе с финансовыми функциями следует помнить следующие правила.

- Единицы измерения процентной ставки и периода должны быть согласованы. Например, если значение аргумента rate – годовая процентная ставка, то аргумент прег также должен быть задан количеством лет. Аналогично, если задана месячная процентная ставка, то количество периодов должно задаваться в месяцах.
- Денежные суммы. Которые вы получаете, следует вводить как положительные величины, а суммы, которые вы выплачиваете, как отрицательные. Например, основную сумму по ссуде всегда следует вводить как положительное число, потому что эти деньги вы получаете от банка.
- 3) Аргумент прег всегда должен быть положительной целой величиной.

Приведем полный список финансовых функций VBA в таблице 15.

## Таблица 15- Финансово-математические функции VBA

Функция	Описание функции		
1	2		
DDB (стоимость, стра-	Значение амортизации актива за определенный пери-		
ховка, длит_жизни, пе-	од, вычисленное методом двойного уменьшения		
риод, фактор)	остатка		

#### Продолжение таблицы 15

1	2		
FV(Rate, Nper, Pmt, Pv, Type)	Будущая стоимость инвестиции или ссуды		
IPMT(Rate, Per, Nper, Pv, Fv, Type)	Сумма платежей по процентам за определенный пери- од для ссулы		
IRR(значение, оценка)	Внутренняя ставка доходности для ряда денежных по-		
MIRR(значение, ставка_доходности,)	Модифицированная внутренняя ставка доходности для ряда периодических денежных потоков		
NPER(Rate, Pmt, Pv, Fv, Type)	Количество периодов для инвестиции или ссуды		
NPV (Rate, значение1, значение2)	Чистая приведенная стоимость инвестиции, вычис- ленная на основе ряда денежных потоков и ставки дискон тирования		
PMT(Rate, Nper, Pv, Fv, Type)	Размер периодической выплаты по ссуде или инвести- ции		
PPMT(Rate, период, Nper, Pv, Fv, Type)	Размер основной выплаты за определенный период по ссуде		
PV(Rate, Nper, Pv, Fv, Type)	Приведенная стоимость инвестиции		
RATE(Nper, Pmt, Pv, Fv, Туре, оценка)	Периодическая процентная ставка для ссуды или ин- вестиции		
SLN(стоимость, стра- ховка, длит_жизни)	Величина амортизации актива за один период, вычис- ленная линейным методом		
SYD(стоимость, стра- ховка, длит_жизни, пе- риод)	Величина амортизации актива за один период, ленная методом «суммы годовых чисел»		

Задача. Клиент (рантье) заключил договор с банком на следующих условиях: он вносит первоначальный вклад в банк, а затем регулярно каждый месяц вносит некоторую сумму, величина которой не меняется. Эти платежи могут производиться либо в начале периода, либо в его конце. В договоре опре-

делена норма прибыли (процентная ставка) по вкладу. Требуется вычислить сумму, которую банк должен выплатить клиенту (величину накопленной стоимости) по окончании срока действия договора при условии постоянных платежей как в конце периодов, так и в их начале. Исходными данными для решения задачи являются:

- 1) Величина начального вклада 10 000 руб.;
- 2) Норма прибыли (процентная ставка) 20%;
- 3) Величина постоянных взносов за один период 100 руб.;
- 4) Срок договора (в периодах) 18 месяцев.

Для решения задачи можно воспользоваться финансово-математической функцией *FV*.

Синтаксис функции

## FV(Rate (ставка), Nper (кпер), Pmt (плата), Pv (нз), Туре (тип))

Функция FV имеет следующие параметры:

- 1) Ставка процентная норма прибыли за период;
- 2) Кпер срок действия договора по вкладу в периодах;
- 3) Плата величина постоянных взносов за период;
- 4) Нз величина первоначального вклада;
- 5) *Тип* признак платежа, учитывающий, когда вносится взнос за период (1 в начале периода, 0-в конце периода).

Программа для выполнения указанных расчетов может иметь следующий код:

Dim Nacoplennaya\_stoimost As Double

Nacoplennaya\_stoimost1 = FV(20/12/100, 18, -100, 10000, 0)Nacoplennaya\_stoimost2 = FV(20/12/100, 18, -100, 10000, 1)MsgBox Nacoplennaya\_stoimost1

MsgBox Nacoplennaya\_stoimost2

Значения первоначального вклада и постоянных выплат в параметрах функции имеют отрицательный знак, т.к. эти выплаты получает банк. Если выплаты получает клиент, то эти значения должны иметь положительный знак.

После выполнения программы результаты будут следующие:

Mici	osoft Excel	×		
-11386.1012677283				
	,	_		
	ОК			

Рисунок 87 – Значение переменной Nacoplennaya\_stoimost1

Microsoft Excel 🛛 🗙
-11351,4487360351
ОК

Рисунок 88 - Значение переменной Nacoplennaya\_stoimost2

Создадим форму с объектами для данной задачи (рисунок 89).

🦳 Расчет накопленной стоимости 📃 🗖 🗙				
Начальный вклад				
Процентная ставка				
Количество периодов выплаты годовой ренты или Срок договора (в месяцах)				
Постоянная величина				
Накопленная стоимость				
Выплаты	-			
<ul> <li>С В начале периода</li> <li>С В конце периода</li> </ul>	Вычислить накопленную стоимость			
:				

Рисунок 89 – Форма с объектами для решения задачи

Программный код для события «щелчок по командной кнопке» должен выглядеть следующим образом

Private Sub Command1 Click()

Dim kolper As Integer, procent As Integer, velich vyplat As Single

Dim srok dogovora As Single, nacopl stoim As Currency, nach vklad As Double, priznac As Integer

If Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Then MsgBox "Нет данных для расчета", vbExclamation, "Расчет накопленной стоимости"

```
Exit Sub
       nach vklad = Val(Text1.Text)
       procent = Val(Text2.Text)
       kolper = Val(Text3.Text)
       velich vyplat = Val(Text4.Text)
       If Option1 = True Then priznac = 1 Else priznac = 0
       nacopl stoim = FV(procent / 12 / 100, kolper, -velich vyplat, -nach vklad,
priznac)
       Text6.Text = nacop1 stoim
```

EndSub

Программу условно можно разбить на три блока. В первом блоке объявляются все необходимые переменные. Во втором проверяется корректность данных, т.е. при отсутствии хотя бы одного значения, необходимого в формуле появляется сообщение (рисунок 90).



Рисунок 90 - Диалоговое окно, появляющееся в случае, если не введены данные в окно

В третьем блоке идет непосредственный подсчет накопленной стоимости и вывод ее на форму.

## 2.14.2 Задания для самостоятельной работы

- 1) Физическое лицо продает машину, бывшую в эксплуатации 12 месяцев и стоившую на момент покупки 250 000 рублей. Рассчитать амортизацию машины на момент продажи, без учета инфляции.
- 2) Клиент взял ссуду в банке в размере 100 000 рублей на период 24 месяцев под годовую процентную ставку размером в 18%. Какова будет сумма его платежей через 6 месяцев?

- 3) Банк инвестировал некоторому юридическому лицу сумму в размере 150 тысяч рублей под 20% годовых. Ежемесячный размер инвестиционных выплат составит составляет 9 000 рублей, срок инвестирования – 36 месяцев. Рассчитать количество периодов данной инвестиции.
- Дом был куплен в марте 2005 года по цене 2 млн. рублей. Рассчитать его стоимость на март 2008 года с учетом амортизации, но без учета инфляции, линейным методом и методом суммирования годовых значений.

#### Список использованных источников

- 1 Глотова М.И. Самостоятельная работа по информатике. Основы разработки Web-сайтов: самоучитель / М.И. Глотова. - Оренбург: ГОУ ОГУ, 2007. - 139 с.
- 2 Шафран, Э. Создание Web-страниц / Э. Шафран. СПб.: Питер, 2000. 320 с.
- 3 Якушина, Е. Изучаем Интернет, создаем Web-страничку / Е. Якушина. - СПб.: Питер, 2002. - 256 с.
- 4 Матрасов А.В., Учебник HTML 4.0 в подлиннике/ Матрасов А.В., Сергеев А.О., Чаунин. ВНУ-Санкт-Петербург, 2004. 672 с.
- 5 Жадаев А.Г. Самоучитель HTML 4 / Жадаев А.Г. Юниор, 2004. 296 с.
- 6 Комолова Н.В. НТМL: учебный курс/ Комолова Н.В. Питер, 2007. 272 с.
- 7 Харт-Дэвис Г. HTML / Харт-Дэвис Г. HT Пресс, 2005. 216 с.
- 8 Кастро Э. Создание Web-страниц с помощью HTML / Кастро Э. HT Пресс, 2005. 144 с.
- 9 Симонович С.В. Информатика. Базовый курс / Симонович С.В. СПб: Питер, 2000. 640 с.
- 10 Мак-Фердис П. Моя первая книга о VBA / Мак-Фердис П. Пер. с англ. М: Эксмо, 2005. 352 с.
- 11 Сафронов И.К. Visual Basic в задачах и примерах / Сафронов И.К. СПб: БХВ-Петербург, 2006. 400 с.
- 12 Ананьев А.И. Самоучитель Visual Basic 6.0. / Ананьев А.И., Федоров А.Ф. СПб: БХВ-Петербург, 2005, 624 с.
- 13 Токарева М.А. Алгоритмизация и программирование. Лабораторный практикум по информатике для студентов экономических специальностей / Токарева М.А., Корякина М.А. Оренбург: ИПК ГОУ ОГУ, 2006. 161 с.
- 14 Угринович Н.Д. Инофрматика и информационные технологии: Учебное пособие для образовательных учреждений / Угринович Н.Д. М.: Бином. Лаборатория Знаний, 2002. 400 с.

15 Назаров С.В. Программирование на MS Visual Basic: Учебное пособие / Назаров С.В., Мельников П.П. – М.: Финансы и статистика, 2002. – 320 с.

# Приложение А

(справочное)

# Цвета, поддерживаемые HTML

# Таблица А1 - Цвета, поддерживаемые HTML

Название цвета	Красная состав-	Зеленая	Синяя
	ляющая	составляющая	составляющая
1	2	3	4
alicemblue	F0	F8	FF
antiquewhite	FA	EB	D7
aqua	00	FF	FF
aquamarine	7F	FF	D4
azure	F0	FF	FF
beige	F5	F5	DC
bisque	FF	E4	C4
black	00	00	00
blanchedalmond	FF	EB	CD
blue	00	00	FF
blueviolet	8A	2B	E2
brown	A5	2A	2A
burlywood	DE	B8	87
cadetblue	5F	9E	A0
chartreuse	7F	FF	00
chocolate	D2	69	1E
coral	FF	7F	50
cornflowerblue	64	95	ED
crimson	DC	14	3C
cyan	00	FF	FF
darkblue	00	00	8B
darkcyan	00	8B	8B
darkgoldenrod	B8	86	0B
darkgray	A9	A9	A9
darkgreen	00	64	00
darkkhaki	BD	B7	6B
darkmagenta	8B	00	8b
darkolivegreen	55	6B	2F
darkorange	FF	8C	00
darkred	8B	00	00
darkseagreen	8F	BC	8F
darkslateblue	48	3D	8B

Продолжение таблицы А.1 3 4 1 2 2F 4F 4F darkslategray darkturquoise 00 CE D1 94 D3 darkviolet 00 FF 93 deeppink 14 deepskyblue BF FF 00 69 69 69 dimgray dodgerblue FF 1E 90 22 22 firebrick B2 floralwhite FF FA F0 8B forestgreen 22 22 FF FF 00 fushsia DC DC DC gainsboro FF F8 ghostwhite F8 gold FF D7 00 20 goldenrod A5 DA 80 80 grav 80 80 00 00 green greenyellow FF 2F AD honeydew FF F0 F0 FF 69 B4 hotpink 5C 5C indiandred CD indigo 82 4B00 FF FF F0 ivory 8C khaki F0 E6 lavender E6 E6 FA FF F0 F5 lavenderblush FC 00 7C lawngreen FA lemonchiffon FF CD D8 E6 ligtblue AD F0 80 80 lightcoral FF FF lightcyan E0 lightgoldenrodyellow FA FA D2 lightgreen 90 ΕE 90 lightgrey D3 D3 D3 lightpink FF B6 C1 FF lightsalmon A0 7A

20

87

77

**B0** 

B2

CE

88

C4

AA

FA

99

DE

lightseagreen

lightslategray

lightsteelblue

lightscyblue

Продолжение таблицы А.1

1	2	3	4
lightyellow	FF	FF	E0
lime	00	FF	00
limegreen	32	CD	32
linen	FA	F0	E6
magenta	FF	00	FF
maroon	80	00	00
mediumaquamarine	66	CD	AA
mediumblue	00	00	CD
mediumorchid	BA	55	D3
mediumpurple	93	70	DB
mediumseagreen	3C	B3	71
mediumslateblue	7B	68	EE
mediumspringgreen	00	FA	9A
mediumturquoise	48	D1	CC
medium violetred	C7	15	85
midnightblue	19	19	70
mintcream	F5	FF	FA
mistyrose	FF	E4	E1
moccasin	FF	E4	B5
navajowhite	FF	DE	AD
navy	00	00	80
oldlace	FD	F5	E6
olive	80	80	00
olivedrab	6B	8E	23
orange	FF	A5	00
orengered	FF	45	00
orchid	DA	70	D6
palegoldenrod	EE	E8	AA
palegreen	98	FB	98
paleturquose	AF	EE	EE
palevioletred	DB	70	93
papayawhop	FF	EF	D5
peachpuff	FF	DA	B9
peru	CD	85	3F
pink	FF	C0	CB
plum	DD	AO	DD
powderblue	B0	E0	E6
purple	80	00	80
red	FF	00	00
rosybrown	BC	8F	8F

Продолжение таблицы А.1

1	2	3	4
royalblue	41	69	E1
saddlebrown	8B	45	13
salmon	FA	80	72
sandybrown	F4	A4	60
seagreen	2E	8B	57
seashell	FF	F5	EE
sienna	A0	52	2D
silver	C0	C0	C0
skyblue	87	CE	EB
slateblue	6A	5A	CD
slategray	70	80	80
snow	FF	FA	FA
springgreen	00	FF	7F
steelblue	46	82	B4
tan	D2	B4	8C
teal	00	80	80
thistle	D8	BF	D8
tomato	FF	63	47
turquose	40	E0	D0
violet	EE	82	EE
wheat	F5	DE	B3
white	FF	FF	FF
whitesmoke	F5	F5	F5
yellow	FF	FF	00
yellowgreen	9A	CD	32

# Приложение Б

(справочное)

## Теги и атрибуты HTML

# Таблица Б.1 – Теги и атрибуты HTML

Название тега	Атрибуты тега	Возможные значения	Назначение тега или атрибута			
		атрибутов				
1	2	3	4			
Основные теги						
<html></html>			Парный тег начала HTML-			
			кода			
<head></head>			Парный тег описания раз-			
			дела заголовков			
<title></title>			Парный тег создания заго-			
			ловка окна			
<body< td=""><td></td><td></td><td>Парный тег описания тела</td></body<>			Парный тег описания тела			
			страницы			
	BGCOLOR	цвет	Цвет фона документа			
	TEXT	цвет	Цвет текста			
	LINK	цвет	Цвет непосещенных ссы-			
			лок,			
	ALINK	цвет	Цвет активной (нажатой в			
			данный момент) ссылки,			
	VLINK	цвет	Цвет посещенных пользо-			
			вателем ссылок			
	BGGRAUND	URL-адрес	Адрес файла с изображе-			
			нием для формирования			
			графического фона стра-			
			ницы			
Теги форматирования текста						
<hi></hi>			Парный тег первого уров-			
			ня абзацев. Толстый,			
			очень крупный шрифт,			
			текст центрирован. Между			
			заголовком и последую-			
			щим текстом вставляется			
			одна или две пустых стро-			
			ки. При выводе на принтер			
			заголовок печатается на			
			новой странице.			

Продолжение таблицы Б.1							
1	2	3	1	4			
<h2></h2>				Парный тег второго уров-			
				ня абзацев. Толстый круп-			
				ный шрифт. Без отступа.			
				До и после заголовка по-			
				мещаются одна или две			
				пустых строки.			
<h3></h3>				Парный тег третьего уров-			
				ня абзацев. Крупный			
				шрифт. До и после заго-			
				ловка помещаются одна			
				или две пустые строки. С			
				небольшим отступом.			
<h4></h4>				Парный тег четвертого			
				уровня абзашев. Толстый			
				нормальный шрифт Ло и			
				после заголовка помеща-			
				ется пустая строка			
<h5></h5>				Парный тег пятого уровня			
11.07				абзацев Нормальный			
				ирифт. Пустая строка ста-			
				вится перед заголовком,			
<u>&lt;по&gt;</u>				Парный тег шестого уров-			
				ня аозацев. Толстый			
				шрифт. Перед заголовком			
				Ставится пустая строка.			
< <u></u>				Парный тег начала аозаца			
<bk></bk>				Парныи тег начала новои			
				строки			
<hr/> HR>				Непарный тег создания			
				разделителя абзацев в			
				виде текстовой линейки.			
	ALIGN			Атрибут выравнивания			
				линейки			
		right		Выравнивание линейки по			
				правому краю страницы			
		left		Выравнивание линейки по			
				левому краю страницы			
		center		Выравнивание линейки по			
				центру			
	SIZE	число	пик-	Высота линейки в пиксе-			
		селов		лах			
Продолжение таблицы Б.1

1	2	3	4
	WIDTH	число про-	Ширина в процентах отно-
		центов	сительно ширины экрана
	COLOR	цвет	Настраивает цвет линии
<font></font>			Парный тег, управляющий
			параметрами шрифта.
	SIZE	целые числа	Задает размер шрифта в
		ОТ	относительных единицах
		-7 до 7	
	COLOR	цвет	Задает цвет текста
	FACE	наименова-	Задает гарнитуру шрифта
		ние шрифта	
<cite></cite>			Парный тег для вывода
			цитат (отображаются кур-
			сивом)
<em></em>			Парный тег выделения
<strong></strong>			Парный тег сильного вы-
			деления
<u></u>			Парный тег подчеркнутого
			шрифта
<s></s>			Парный тег зачеркнутого
			шрифта
<b></b>			Парный тег полужирного
			шрифта
<i></i>			Парный тег курсива
<sub></sub>			Парный тег нижнего ин-
			декса
<sup></sup>			Парный тег верхнего ин-
			декса
<marquee></marquee>			Парный тег бегущей стро-
			КИ
<basefont/>			Парный тег задания пара-
			метров вывода текста по
			умолчанию
	COLOR	цвет	Задает цвет текста
	FACE	наименова-	Задает гарнитуру шрифта
		ние шрифта	
	SIZE	целые числа	Задает размер шрифта в
		ОТ	относительных единицах
		-7 до 7	

Продолжение таблицы Б.1

1	2	3	Δ
	<u> </u>		к К
$\langle \Lambda \rangle$			
			парный тег создания ти-
	Прее		
		UKL - adpec	задает адрес страницы, на
	NAME	una	
		ИМЯ	Отмечает якорь на страни-
	Тап работи с		
		і изооражения	
			пепарный тег внедрения
		LIDI	изооражения в страницу
	SKC	URL - adpec	задает адрес изооражения
			для внедрения
	WIDTH	число	Задает ширину рисунка
			для масштабирования
	HEIGHT	число	Задает высоту рисунка для
			масштабирования
	ALIGN		Атрибут взаимодействия
			рисунка с текстом
		botton	Нижняя граница изобра-
			жения совмещается с
			основанием строки.
		middle	Середина изображения
			совмещается с серединой
			текстовой строки.
		top	Верхняя граница изобра-
			жения совмещается с
			верхним краем строки
		left	Изображение размещается
			у левого края страницы, а
			последующий текст разме-
			щается справа от него
		right	Изображение размещается
		C	у правого края страницы, а
			последующий текст разме-
			щается слева от него
	HSPACE	число	Величина промежутка
			между изображением и
			текстом (в пикселах) по
			горизонтали

Продолжение таблицы Б.1

1	2	3	4
	VSPACE	число	Величина промежутка
			между изображением и
			текстом (в пикселах) по
			вертикали
	ALT	текст	Отображение альтернатив-
			ного текста для изображе-
			ния. Атрибут используется
			для тех случаев, если кар-
			тинка не может быть за-
			гружена, т.е. нет средства
			для ее просмотра. Альтер-
			нативный текст отобража-
			ется в виде всплывающей
			подсказки, появляющейся
			при наведении на картин-
			ку указателя мыши
	USEMAP	имя карты	Связывает рисунок с
			картой
	Теги работ	ы с таблицами	4
<table></table>			Парный тег создания та-
			блицы
	CELLSPACING	число	Зазор между ячейками в
			пикселах
	CELLPADDING	число	Зазор между содержимым
			ячейки и ее границей в
			пикселах
	BORDER	число	Отображение границ ячеек
			и внешней рамки таблицы
			в пикселах
	BORDERCOLOR	цвет	Задает цвет границ табли-
			цы
	ALIGN		Выравнивание текста от-
			носительно таблицы, в ко-
			торой он находится
		left	выравнивает текст по ле-
			вому краю ячейки
		center	выравнивает текст по цен-
			тру
		right	выравнивает текст по пра-
			вому краю ячейки

Продолжение таблицы Б.1

1	2	3	4	
	HSPACE	число	Размер свободного про-	
			странства слева и справа	
			от таблицы, в пикселах	
	VSPACE	число	Размер свободного про-	
			странства сверху и снизу	
			от таблицы, в пикселах	
	WIDTH	число, про-	Требуемая ширина табли-	
		цент	цы в пикселах или в про-	
			центах от ширины окна	
			бруазера	
<caption></caption>			Парный тег описания заго-	
			ловка таблицы	
<tr></tr>			Парный тег задания стро-	
			ки таблицы	
	ALIGN	left, center,	Выравнивание текста от-	
		right	носительно строки, в кото-	
			рой он находится	
	BGCOLOR	цвет	Цвет фона строки	
<th></th>				Парный тег задания ячеек
			строки заголовка таблицы	
	COLSPAN	число	Протяженность ячейки в	
			несколько столбцов, в ко-	
			личестве столбцов	
	ROWSPAN	число	Протяженность ячейки в	
			несколько строк, в количе-	
			стве срок	
	BGCOLOR	цвет	Цвет фона ячейки	
	ALIGN	left, center,	Выравнивание текста от-	
		right	носительно ячейки, в ко-	
			торой он находится	
	VALIGN		Выравнивание текста и	
			графики внутри ячейки по	
			вертикали	
		top	выравнивает содержимое	
			ячейки по ее верхней гра-	
			нице	
		middle	центрирует содержимое	
			ячейки по вертикали	
		bottom	выравнивает содержимое	
			ячейки по ее нижней гра-	
			нице	

Продолжение таблицы Б.1

1	2	3	4	
<td></td> <td></td> <td></td> <td>Парный тег задания ячеек</td>				Парный тег задания ячеек
			обычной строки	
	COLSPAN	число	Протяженность ячейки в	
			несколько столбцов, в ко-	
			личестве столбцов	
	ROWSPAN	число	Протяженность ячейки в	
			несколько строк, в количе-	
			стве срок	
	BGCOLOR	цвет	Цвет фона ячейки	
	ALIGN	left, center,	Выравнивание текста от-	
		right	носительно ячейки, в ко-	
			торой он находится	
	VALIGN		Выравнивание текста и	
			графики внутри ячейки по	
			вертикали	
		top	выравнивает содержимое	
			ячейки по ее верхней гра-	
			нице	
		middle	центрирует содержимое	
			ячейки по вертикали	
		bottom	выравнивает содержимое	
			ячейки по ее нижней гра-	
			нице	
	Теги созд	ания списков		
<ol></ol>			Парный тег создания ну-	
			мерованного списка	
<li></li>			Парный тег задания эле-	
			ментов нумерованного	
			списка	
<ul></ul>			Парный тег создания не-	
			нумерованного списка	
<li></li>			Парный тег задания эле-	
			ментов ненумерованного	
			списка	
	TYPE		Задает вид маркера нену-	
			мерованного списка	
		disc	Маркер в виде круга	
		square	Маркер в виде квадрата	
		circle	Маркер в виде окружно-	
			сти	

Продолжение таблицы Б.1

1	2	3	Λ
		5	
∼DL>			парный тег списка опре-
			делении
<d1< td=""><td></td><td></td><td>Парныи тег создания тер-</td></d1<>			Парныи тег создания тер-
			мина определения
<dd></dd>			Парный тег создания
			расшифровки термина
			определения
	Теги рабо	гы с фреймами	1
<frameset>.</frameset>			Парный тег описания
			внешнего вида фреймов
	ROWS	список ве-	Количество строк во
		пичин	фреймах
	COLS	список ве-	Количество столбнов во
		пициц	фреймах
	BORDER		
		число	
			ления всех рамок для всех
			кадров
	FRAMEBORDER	no, yes	Включает или выключает
			отображение обрамления
			кадров. В случае уез рамка
			имеет трехмерную форму,
			иначе она невидима, то
			есть имеет цвет фона окна
			по умолчанию. Некоторые
			браузеры "понимают" эту
			опцию только в виде
			frameborder=1 или
			frameborder=0, поэтому
			обычно указывают оба
			способа
	FRAMESPACING	число	Указывает ширину проме-
			калрами в пискалах. На
			кадрами в пискелах. Пе
			Leucinger B inciscape
<fkame></fkame>			Описание отдельных кад-
			ров фрейма
	SRC	URL-адрес	URL исходного документа
			для данного кадра
	NAME	ИМЯ	Указывает имя кадра

Продолжение таблицы Б.1

1	2	3	4
	SCROLLING		Управляет линейками про-
			крутки кадра
		yes	Линейки есть всегда
		no	Линеек нет никогда
		auto	Линейки появляются, если
			это необходимо
	NORESIZE		Запрещает изменять раз-
			меры кадра; данный атри-
			бут, указанный для кон-
			кретного кадра, влияет
			также и на все кадры,
			смежные с ним
	FRAMEBORDER	no, yes	Включает или выключает
			отображение обрамления
			кадров
	MAGREENHEIGHT	число	Задаёт размещение по
			верхней и нижней и сторо-
			нам кадра областей сво-
			бодного пространства
	MAGREENWIDTH		Задаёт размещение по бо-
			ковым сторонам кадра об-
			ластей свободного про-
			странства
	Теги рабо	оты с картами	r
<map></map>			Парный тег создания
			меню-карты
	NAME	ИМЯ	Задает имя карты
<area/>			Парный тег описания
			отдельной чувствительной
			области карты
	SHAPE		Определяет форму чув-
			ствительной области
		rect	Задает область в форме
			прямоугольника
		poly	Задает область в форме
			многоугольника
		circle	Задает область в форме
			круга
	COORDS	список	Определяет расположение
		координат	чувствительной зоны на
		точек	карте

Продолжение таблицы Б.1

1	2	3	4
	HREF	URL-адрес	Указывается URL-адрес
			документа, адресуемого
			данной областью
	Теги соз	здания форм	
<form></form>			Парный тег создания фор-
			МЫ
	ACTION	URL-адрес	Указывает URL-адрес, ко-
		-	торый примет и обработа-
			ет данные формы. Если
			эта опция не указана, дан-
			ные отправляются по ад-
			ресу страницы, на которой
			размещена форма
	METHOD		Указывает метод передачи
			данных программе-обра-
			ботчику формы
		post	Предписывает посылать
			информацию формы
			отдельно от URL. Обычно
			используется в случае от-
			правки данных формы по
			электронной почте или
			при необходимости пере-
			давать значительный
			объем информации.
		get	Предписывает посылать
			информацию формы вме-
			сте с URL
	NAME	ИМЯ	Указывает имя формы.
			Это необходимо, если тре-
			буется доступ к данным
			формы с помощью
			встроенного скрипта на
			JavaScript или Visual Basic
			Script, а также в том слу-
			чае, если данные формы
			предназначены для от-
			правки по электронной по-
			чте

Продолжение таблицы Б.1

1	2	3	4
	ENCTYPE	способ ко-	Задает способ кодирова-
		дирования	ния данных формы. В слу-
		данных	чае отправки данных как
			текста указывается в виде
			enctype="text/plain".
<textarea></textarea>			Парный тег для создания
			многострочного поля вво-
			да
	COLS	число	Указывает число столбцов
			вполе
	ROWS	число	Указывает число строк в
			поле
	WRAP		Уазывает на режим авто-
			матического распределе-
			ния текста в ячейке
		off	Режим выключен
		virtual	Распределять текст по
		Viituui	всей днейке но на сервер
			передарать как одну стро-
			передавать как одну стро-
		nhysical	ку Репределять текст по всей
		physical	переделить текст по веси
			сервер так как он отобра
			сервер Так, как он отоора-
			жается
		ИМЯ	
<selec 1=""></selec>			Парный тег определяю-
			щии в форме меню с од-
			ним или несколькими ва-
			риантами выоора или спи-
			сок с полосои прокрутки
	SIZE	число	Организуется список с
			указанным числом види-
			мых элементов
	MULTIPLE		Разрешает выбирать из
			списка более одного зна-
			чения
	NAME	ИМЯ	Имя списка
<pre><option></option></pre>			Парный тег описания эле-
			мента списка

Продолжение таблицы Б.1

1	2	3	4
	VALUE	значение	Указывает значение, воз-
			вращаемое программе об-
			работки при выборе поль-
			зователем данного элемен-
			та
	SELECTED		Указывает на элемент
			списка, выбранный по
			умолчанию
<input/>			Непарный тег для созда-
			ния текстовых полей, по-
			лей для ввода пароля,
			переключателей, флажков,
			кнопок для отправки дан-
			ных и очистки формы
	TYPE		Атрибут указания созда-
			ния конкретного поля
			формы
		text	Создает поле ввода
		password	Создает текстовое поле с
			защитой вводимых симво-
			ЛОВ
		checkbox	Создает поле флажок
		radio	Создает поле радиокнопка
		submit	Создает кнопку отправки
			данных формы
		reset	Создает кнопку очистки
			формы
	MAXLENGTH	число	Ограничивает максималь-
			ную длину вводимого тек-
	0.000		ста
	SIZE	число	Показывает максимальное
			количество отображаемых
			СИМВОЛОВ
	VALUE		Определяет значение, воз-
			вращаемое программе об-
			работки при выборе поль-
			зователем флажка, радиок-
			нопки, создает надпись
			для кнопок

Продолжение таблицы Б.1

продоля		1	
1	2	3	4
	CHECKED		Делает флажок или ра-
			диокнопки выбранными
			по умолчанию
	Теги работь	і с мультимеді	1a
<bgsound/>			Тег внедрения звукового
			файла
	LOOP	целое чис-	Определяет количество
		ло;	воспроизведений
		"infinite"-	
		повторение,	
		пока поль-	
		зователь на-	
		ходится на	
		странице;	
	VOLUME	число в	Назначает уровень вос-
		диапазоне	произведения звука
		(-10000;0)	
	BALANCE	число в	Настраивает стереобаланс
		диапазоне	
		(-10000;100	
		00)	
<embed/>			Тег встраивания в страни-
			цу аудиоплеера
	WIDTH	число; про-	устанавливает ширину па-
		цент	нели управления проигры-
			вателя
	HEIGHT	число; про-	устанавливает высоту па-
		цент	нели управления проигры-
		6-1	вателя
	AUIUSIAKI	Talse –	назначает спосоо включе-
		включение	ния звука
		звука поль-	
		subarellem,	
		матическое	
		включение;	

### Продолжение таблицы Б.1

1	2	3	4
	HIDDEN	false – отоб-	управляет отображением
		ражает па-	панели проигрывателя
		нель;	
		true – не	
		отображает	
		панель;	

#### Приложение В

(справочное)

#### Регистрация сайта в информационно-поисковых системах и другие мероприятия по его развитию

Прежде, чем размещать свой сайт в Internet, нужно позаботиться о том, чтобы он попал в индекс информационно-поисковых систем. Желательно для всех страниц Вашего сайта необходимо составить наборы ключевых слов, которые наиболее точно и полно отражают содержание каждой страницы. Эти наборы слов помещаются в специальный тег <META>.

<МЕТА> - одиночный тег, который задает метаинформацию (информацию об информации), располагается в контейнере <HEAD> и предназначен для информирования серверов о Вашей странице (сайте).

Формат использования тега:

<HEAD>

<TITLE>Название документа</TITLE>

<META name="имя" content="набор ключевых слов">

</HEAD>

В случае обнаружения поисковой системой Вашего сайта она просмотрит содержимое всех тегов <META> и включит указанные Вами ключевые слова в свою базу данных.

После того, как заданы ключевые слова, можно приступать к регистрации сайта в поисковых системах. Следует отметить, что все поисковые системы регулярно в автоматическом режиме выполняют индексацию новых Web-страниц. Однако лучше не ждать автоматической индексации, а зарегистрировать свой сайт вручную.

Для регистрации сайта необходимо выполнить действия по следующей общей схеме:

- 1 загрузить начальную страницу информационно-поисковой системы, например, системы **Rambler**;
- 2 щелкнуть по ссылке Добавить ресурс, после чего откроется страница Регистрация, представленная на рисунке В.1 (в случае работы с другой системой щелкнуть по аналогичной ссылке, например для регистрации сайта в системе Yandex нужно щелкнуть по ссылке Добавить сайт);
- 3 в появившемся окне *Регистрация* щелкнуть по кнопке **Начать регистрацию**, после чего откроется страница, представленная на рисунке В.2 (перед регистрацией следует проверить не нашел ли робот системы Ваш сайт самостоятельно, для чего в поле *Имя сайта* введите его имя и щелкните по кнопке **Проверить**; если Ваш сайт проиндексирован, то Вам система выдаст соответствующее сообщение);
- 4 заполнить форму, представленную на рисунке В.2.

9 Регистрация - Орега	
📔 File Edit View Navigation Bookmarks Mail Chat Tools Window Help	-
Copen Save Print Find Home Panels Tile Cascade	
🗀 Opera 🔹 🚱 Opera Community 📓 Opera Web Mail 🔍 Price Comparison search	
🖀 New page 🛞 Rambler: информационно 🔽 http://www.osu.ru/doc/93 🔊 Transfers 💭 Пидкасистый П.И Орга 📑 Регистрация	
📢 • 🔄 • 🗩 • 😥 • 💋 🖉 📔 http://www.rambler.ru/doc/add_site.shtml 💽 🔍 Google se	arch 💌 🥶 90% 💌 重
Рамблер	
Rambler	Расширенный поиох Найти! Помощь в поиоке
Регистрация	О Рамблере
Приглашаем Вао зарегиотрировать Ваш сайт: • в поисковой онотеме Rambler; • в интернет-рейтикте Rambler Top100; • в рейтикте интернет-магазинов Rambler's TopShop. • в рейтикте интернет-магазинов Rambler's TopShop. • Сакат-Петербург • Сакат-Сасе Hanke • Сакат-Сасе Hanke • Сакат-Сасе Hanke • Сакат-Сасе Hanke • Сакат-Сасе Hanke • Навости	
гелистрация самта в помосноко мотеме капприет есть приглашение росоту посетить страницы ресуроа. Обычно между регистрациеи - Новости компании сайта и полялением Ваших документов в результатах поиска проходит 2-4 недели. - Оклайновые СМИ о нас	
Начать регистрацию	- Пресс-служба
Перед региотрацией рекомендуем Вам проверить, не нашел ли ухе робот Ваш сайт оамоотолтельно. Проверка показывает, какие отраницы с Вашего сайта уже обработаны Рамблером	Поиск - Зарегистрировать ресурс - Удалить ресурс - Поисковая машина - Поисковье фоомы
Има сайта Напоммер: science rambler, ru, www.ienta.ru, www.bostinu.ua/~name(поспе "/" - только каталоги, начинающиеся с	- Часто задаваемые вопросы - Помощь
"~") Выдавать по ⊙ 15 _30 _60 документов на отраницу	Реклама - Реклама на Рамблере - Статиотика поисковых сапросов - Технические требования
Проверить!	

# Рисунок В.1 – Страница Регистрация поисковой системы Rambler

Lon view Navigation bookmans mai Chet Loois Window Heip Print Find Home Panels The Cascade ra  Page @Rambler: HeidporeautorHom  Phttp://www.rambler.ru/doc/add_site_form.shtm? Page @Rambler: HeidporeautorHom  Phttp://www.rambler.ru/doc/add_site_form.shtm? Cocode search	erpanyoreavita 8 1109		
Save Print Find Hore Panels Te Cascade   ra • © Opera Community © Opera Web Mall Price Comparison search Panels Panels Panels Price Comparison search Panels Panels Panels Panels Price Comparison search Panels </th <th>Edit View Navigation</th> <th>n Bookmarks Mail Chat Tools Window Help</th> <th></th>	Edit View Navigation	n Bookmarks Mail Chat Tools Window Help	
Save Print Find Home Panels Tile Cascade ra	🚽 👼 🔍		
ra · Opera Community Opera Web Mall Price Comparison search Page Rambler: Hebpophauutoeko Phttp://www.cou.uru/doc/93 Transfers Pruzkackonski RLM Opra Perketpouula cakina b now	Save Print Find	Home Panels Tile Cascade	
Раде Мальвіет: информационно № http://www.osu.ru/doc/93 № Transfers Пидкасистый П.И Орга № Perистрация Сайта в поли № № № № № № № № № № № № № № № № № № №	ra 🝷 🛃 Opera Communit	y 🗟 Opera Web Mail 🔍 Price Comparison search 🔍 Amazon.com search	
	page 🛞 Rambler: инфо	ормационно 📔 🗣 http://www.osu.ru/doc/93 🚺 Transfers 📔 Пидкасистый П.И Орга 📘	Регистрация сайта в пои
яклен  Контактное лицо Контактное лицо Сортака с поиходых призедениях  Контактное лица Сортака с поиходых призедениях  Сортака с поиходых  Со	套 • 🍌 • 😥	🔹 🍲 🥖 📔 http://www.rambler.ru/doc/add_site_form.shtml?	Google search
Интернет         На кооти         Картинии         Покулки         Люди         Рісе. ли         Слогари           На йт         О Рамбле         О Рамбле         О Рамбле         О Пороекте         О пороекте         О пороекте         О количе         О рамбле           Владельцам русокоязычных сайтов, расположенных в доменах. сот, огд. лет и др. необходимо прочесть ракомендациях.         О рамбле         О рамбле         О рамбле           ИВ толовной отраницы         Покулки         др. необходимо прочесть ракомендациях.         О рамбле         О рамбле           И Па роскова         О рамбле         О рамбле         О рамбле         О рамбле         О рамбле           И Па роскова         О рамбле         О рамбле         О рамбле         О рамбле         О рамбле           И Па роскова         О рамбле         О рамбле         О рамбле         О рамбле         О рамбле           И Па роскова         О рамбле         О рамбле         О рамбле         О рамбле         О рамбле           И Па роскова         О рамбле         О рамбле         О рамбле         О рамбле         О рамбле           И Па роскова         О рамбле         О рамбле         О рамбле         О рамбле         О рамбле           Контактиков лицо         По котактиков         О рамбле	блер		
Полосии         Найт           Регистрация         0 Разбле           ваш сайт может быть зарегистрирован с помощью приведенной ниже формы, если он находится на территории (в национальных доменах) России и отран СНГ.         0 Разбле           Владельцам русскоязычных сайтов, расположенных в доменах .com, .org, .net и др. необходимо прочесть рекомендации.         0 Фри с К           И. R. головной отраницы         - Офи с С         - Склаться           И. R. головной отраницы         - Прессо сл         - Орес сли           Короткое описание региотрируемого сайта         - Орес сли         - Орес сли           Контактное лицо         - Статься         - Орес сли           Етлаї Контактного лица         - Статься         - Статься           Зарагистриковала         - Статься         - Статься			
Контактное лицо      Контактное лица      Ко	Pomble	интернет повости (партинки (покупки (пюди (поела (словари	
енистрация аш сайт может быть зарепистрирован с помощью приведенной ниже формы, если он находится на территории (в национальных доменах) России и отран СНГ. Владельцам русскоязычных сайтов, расположенных в доменах .com, .org, .net и др. необходимо прочесть рекомендации. Название сайта URL головной страницы http:// Короткое описание регистрируемого сайта Composition Compositi	Kunole		Най
Тегнатрация  аш сайт может быть зарепистрирован с помощью приведенной ниже формы, если он находится на территории (в национальных доменах) России и отран СНГ.  Владельцам русокоязычных сайтов, расположенных в доменах .com, .org, .net и др. необходимо прочесть рекомендации.  Название сайта  URL головной страницы http:// Короткое описание регистрируемого сайта  Контактное лицо Е-mail контактного лица Зарагистрировать Ористить фолци			
аш сайт может быть зарепистрирован с помощью приведенной ниже формы, если он находится на территории (в национальных доменах) России и огран СНГ. Владельцам русскоязычных сайтов, расположенных в доменах .com, .org, .net и др. необходимо прочесть рекомендации. Владельцам русскоязычных сайтов, расположенных в доменах .com, .org, .net и др. необходимо прочесть рекомендации. URL головной ограницы http:// Короткое описание регистрируемого сайта Почск Зарегисто лица E-mail контактного лица Dukotatta.com	егистрация		U Pambi
Владельщая русскоязычных сайтов, расположенных в доменах .com, .org, .net и др. необходимо прочесть рекомендации. Наша стат Владельщая русскоязычных сайтов, расположенных в доменах .com, .org, .net и др. необходимо прочесть рекомендации. Наша стат Владельщая стат Владельшая стат Владельщая стат Владельщая стат Владельшая стат Владельствивая стат В	аш сайт может быть зарегистр	жрован с помощью приведенной ниже формы, если он находится на территории (в национальных доменах) Росси	иистранСНГ. Опроект
Владельцая русскоязычных сайтов, расположенных в доменах кот, org, net и др. необходимо прочесть рекомендации. - Одио в С - С налаться - Инаселии - Илание - Инаселии - Инаселии - Инаселии - Пресса о - Онлание - Почкок - Зарегистрируемого сайта - Почкок - Почкока - П			- U KOMITA - Hawa cr
Название сайта - Офис в С - Стязаться ИRL головной страницы - Новости - Новости http:// Короткое описание регистрируемого сайта - Пресс-ол Почек - Зарегистр - Удалить р - Помосье - Часто зад - Помосье - Часто зад - Помосье - Часто зад - Помосье - Стятистик - Техническ - Стятистик - Техническ - Стятистик - Техническ - Стятистик - Техническ - Стятистик - Техническ - Стятистик - Техническ	Владельцам русскоязычны	их сайтов, расположенных в доменах .com, .org, .net и др. необходимо прочесть рекомендации.	- Вакансия
Название сайта - Славноя ИRL головной ограницы http:// Короткое описание регистрируемого сайта - Пресса оп Короткое описание регистрируемого сайта - Пресса оп Поиск - Зарегистрируемого сайта - Пресса оп -			- Офис в
название саита - Силавесо название саита - Силавесо ИRL головной ограницы			- Офис в
Новости URL головной ограницы http:// Короткое описание регистрируемого сайта Короткое описание регистрируемого сайта Контактное лицо Контактное лицо Е-mail контактного лица Зарагиотр лица Опистрировать Опистрировать	название саита		- Censale
URL головной ограницы         - Нотости         - Поская           http://         - Пресса ол         - Онлайнов           Короткое описание регистрируемого сайта         - Пресса ол         - Пресса ол           Короткое описание регистрируемого сайта         - Пресса ол         - Пресса ол           Короткое описание регистрируемого сайта         - Пресса ол         - Поиске           - Поиске         - Зарегистр         - Удалить р           - Поиска         - Реклама         - Поиске           - Поиска         - Реклама         - Статистик           E-mail контактного лица         - Статистик         - Техническ			Новости
http:// Короткое описание регистрируемого сайта - Пресса от - Оклайное - Пресса от - Пресса - Пресса	URL головной страницы		- Новости
Короткое описание регистрируемого сайта - Пресо слу Почок Заретистри Контактное лицо Е-mail контактного лица Е-mail контактного лица Варетистри роката Варетистри роката Варети	http://		- Hpecca (
Контактное лицо Е-тай контактного лица Зарегиотр Удалить Контактное лицо Солтактное лицо Солтактное лицо Станотика Солтактное лицо Станотика Станотика Солтактное лицо Станотика Солтактное лицо Станотика С	Короткое описание регистр	рируемого сайта	- Пресс-с
Контактное лицо Е-тай контактного лица Зарегиотр Удализь р Сонтактное лицо Сонтактное лицо Сонтактное лицо Сонтактное лицо Сонтактное лицо Становка Сонтактное лицо Становка Сонтактное лицо Становка Стан	· · · ·		
- Зарегистрировать форму			Поиск
Сонтактное лицо Е-тодика Контактное лицо Е-тодика Сонтактное лица Е-тодика Сонтактное лица Станотика			- Зарегис
- Помозовы - Чаото заа - Помозовы - Чаото заа - Помозовы - Чаото заа - Реклама - Статиотик - Статиотик - Статиотик - Техническ			- Удалить
- Чаото зад - Помоц - Селина - Статистик - Статистик			- Поисков
Контактное лицо — Помощь Контактное лицо — Реклама Е-mail контактного лица — Статистик Залегистрировать — Опистить форму			- 4acto 32
Контактное лицо Реклама - Реклама - Реклама - Реклама - Реклама - Статионик Е-mail контактного лица - Статионик Зарегистрировать Оцистить форму			- Помощь
Контактное лицо Реклама Е-mail контактного лица - Статиотик Залегистрировать Очистить форму			
- Реллана - Статиотик Е-mail контактного лица - Статиотик Зарагиоточногать Оринстить форму	Контактное лицо		Реклама
- Станония Е-mail контактного лица - Техническ Зарегистрировать - Оцистить форму			- Реклама
	E mail yoursymana muss		- Статисти
Зарагистрировать Пинстить форму	счтал контактного лица		- Техниче
Зарегистрировать Пинстить форму			
	Зарагистриров	Онистить форму	

Рисунок В.2 – Страница Регистрация сайта в поисковой машине Rambler

Если Вы внесли изменения в страницы Вашего сайта, то его рекомендуется перерегистрировать.

После того, как сайт зарегистрирован, следует подумать о его рейтинге. Рейтинги – это каталоги в Internet, в которых размещаются данные о посещаемости сайта. Существует достаточно много российских рейтингов ,предоставляющих эту услугу (Rambler Top 100, Mail.Ru, HotLog и др.). Для участия Вашего сайта в рейтинге необходимо установить счетчик на вашем узле. Рассмотрим алгоритм установки счетчика в бесплатном рейтинге Rambler Top 100. Для этого необходимо:

- 1 загрузить начальную страницу рейтинга **Rambler Top 100** (<u>http://top100.rambler.ru/</u>);
- 2 щелкнуть по ссылке Добавить ресурс;
- 3 в появившемся окне (рисунок В.3, В.4) заполните форму (укажите адрес сайта, краткое описание, раздел рейтинга, в котором будет участвовать Ваш сайт, логотип счетчика и др.);
- 4 щелкните по кнопке Зарегистрироваться.

🚷 Rambler's Top100 - Opera	
📔 File Edit View Navigation Bo	okmarks Mail Chat Tools Window Help
Open Save Print Find Ho	ne Panels Tile Cascade
🗀 Opera 🝷 隆 Opera Community 🛛 🔮	Opera Web Mail Sprice Comparison search
New page Rambler's Top100	🏽 http://www.osu.r [ 🔊 Transfers 🗌 Пидкасистый П 🛞 Rambler: 'тамбов 🛟 Открытки 🌉 Служба знакомс
🛃 • 🛫 • 🖻 • 🖻 • 😵	🛿 🖉 📔 http://top100.rambler.ru/cgi-bin/mod_top100.cgi?reg=accept
	Rombler's TOP100
Rambler's Top 100 >> Регистрационная	форма
	Поля, обязательные для заполнения, помечены симеолом * Если у вы есть копросы по заполнению <u>склюненсь с номе</u> .
	*URL Вашей странюцы:
	*Название ресурса:
	Краткое описание:
•	
	*Апрес e-mail, худа будет кыслано сообщение о регистрации:
	*Вы регистрируетесь как: физическое лицо представитель юридического лица
	*Ф.И.О. лица, производящего регистрацию:
	Снююнимы Вашей страницы (по одному на каждую строку): Это пове должео запонаться в тех случаю, когда у вашей страенцы некотся рызвенанае адреса (URL) содержане которых в точности однаности. Если Баша, спранца нас http://donain.ru/dir/index.html it http://donain.ru/dir/isysteo указывать URL с index.html, a коронизй спеконен будет стеверярован системой алгонентески (как названия недеосных самента formonianers index.*, home.* it default.*)

Рисунок В.3 – Страница регистрации Rambler's Top100

После чего на следующей странице появится сообщение, что счетчик для Вашего ресурса успешно зарегистрирован, и будет указан ID Вашего счетчика.

🕄 Rambler's Top100 - Opera
📄 File Edit View Navigation Bookmarks Mail Chat Tools Window Help
Open Save Print Find Home Panels Tile Cascade
🗀 Opera 🝷 🛃 Opera Community 📓 Opera Web Mail 🔍 Price Comparison search
Rambler's top100 Chttp://www.osu.r Transfers OTlugkacucituu II CRambler's tamboe) Otkputku - C
😪 🔨 👻 🕐 👻 🥙 🖉 📋 http://top100.rambler.ru/cgi-bin/mod_top100.cgi?reg=accept 🛛 💽 🔍 Google se
Это ССІ скрытт?
+II-1
Bombler's Bombler's
Τορ100 Τορ100
00004875 counter
•Да Онет
Будет использован счетчик 1х1 с установкой логотипа:
Statements for Rembler's bo
VMACTHINK TOP         VMACTHINK TOP         VMACTHINK TOP         VMACTHINK TOP           Bambler's DD         Bambler's DD         Bambler's DD         Bambler's DD
Kamhler's     Kamhler's     Kamhler's       Top Infr     Top Infr     Top Infr
VUDCTIME VUD
VIACTNAK VIZCTNAK VIACTNAK VIACTNAK VIACTNAK KANDA
TOP ION         TOP ION         TOP ION         TOP ION           VIRCTNAK         VIRCTNAK         VIRCTNAK         VIRCTNAK
C Randber C Rand
Viacrane Rambierk Tarpano
±۳
·raздел ректинга тортоо в котором оудет участвовать ваш ресурс: Other(Прочее)
Зарегистрироваться (же намалийте деледы)

Рисунок В.4 – Страница регистрации Rambler's Top100

В целях развития сайта осуществляются и другие мероприятия по его раскрутке:

1 размещение описания сайта в регулярных рассылках;

- 2 обмен ссылками с другими сайтами;
- 3 включение адреса сайта в сообщения почты;
- 4 публикация на сайте собственных статей;
- 5 баннерная реклама.

Указанные мероприятия отличаются степенью эффективности, оперативностью результатов, затраченным на них временем.

Познакомиться с технологией проведения таких мероприятий Вы можете при углубленном изучении темы и их систематическое выполнение в итоге приведет к значительному росту популярности Вашего сайта.

# Приложение Г

(справочное)

### Функции VBA

Таблица Г.1 Математические функции VBA

Функция	Возвращаемое значение
Abs (x)	модуль числа
Atn (x)	arctg (x) – арктангенс от значения параметра, заданного в ра-
	дианах
Sin (x)	sin (x) – возвращает синус угла от значения параметра, задан-
	ного в радианах
Cos (x)	cos (x) – косинус указанного в радианах угла
Tan (x)	tg (x) – возвращает тангенс угла от значения параметра, задан-
	ного в радианах
Exp (x)	е <sup>х</sup> – возвращает число е, возведенное в указанную степень, где
	е – основание натурального логарифма
Log (x)	ln (x) – возвращает натуральный логарифм от значения число-
	вого выражения
Sqr (x)	- возвращает квадратный корень числового выражения
Rnd (x)	Случайное число из интервала [0,1). Перед вызовом функции
	надо использовать оператор Randomize (рандомизации) -
	запуск генератора псевдослучайных чисел)
Sgn (x)	Возвращает +1, если значение параметра положительно,
	-1, если отрицательное,
	0, если 0
Fix (x)	Отсекает дробную часть числа с плавающей точкой
Int (x)	Возвращает результат округления выражения с плавающей
	точкой до целой части. В случае отрицательного параметра
	возвращает ближайшее меньшее отрицательное число
Hex(x)	Возвращает значение аргумента в шестнадцатеричной системе
	счисления
Oct(x)	Возвращает значение аргумента в восьмеричной системе счис-
	ления
Round(x, чис-	Число, округленное до заданного числа десятичных знаков
ло знаков)	

Таблица Г.2 Функции преобразования типов

Функция	Возвращаемое значение
Val (аргумент)	Возвращает числа, содержащиеся в строке, как числовое зна-
	чение соответствующего типа
Str (аргумент)	Возвращает значение типа variant (String), являющееся стро-
	ковым представлением числа
Int(аргумент)	Возвращает аргумент, преобразованный в значение типа Inte-
	ger

## Таблица Г.3 Функции работы со строками

Функция	Возвращаемое значение
1	2
Asc(string)	ANSI-код первого символа в строке string
AscB(string)	Байт, соответствующий первому символу в строке string
AscW(string)	Код символа Unicode, соответствующий первому симво-
	лу в строке string
Chr(charcode)	Символ, соответствующий ANSI-коду charcode (тип ре-
	зультата — Variant)
Chr\$(charcode)	Символ, соответствующий ANSI-коду charcode (тип ре-
	зультата — String)
ChrB(charcode)	Байт, соответствующий ANSI-коду, заданному аргумен-
	том charcode
ChrW(cbarcode)	Символ Unicode, который соответствует ANSI-коду
	charcode
Filter(sourcearray,	Из заданного массива строк (sourcearray) возвращает
match,)	подмножество строк (т.е. другой массив), которые удо-
	влетворяют критерию match
InStr(start, stringl,	Позиция первого появления символа подстроки string2 в
string2)	строке stringl, начиная с позиции start
InStrB(start, Btringl,	Номер байта первого появления подстроки string? в
String2)	строке atringl, начиная с позиции start
inStrRev (,	Позиция символа при первом появлении подстроки
stringьatcn, start)	stringcheck, В строке stringmatch, начиная с позиции
	start (функция проверяет строку с конца до начала)
LCase(atring)	Преобразует символы строки string в нижний регистр
	(тип результата — Variant)
LCase\$(string)	Преобразует символы строки string в нижний регистр
	(тип результата — string)
Left(string, length)	Из строки string возвращает length символов, начиная с
	крайнего слева символа (тип результата — Variant)

Продолжение таблицы Г.3

1	2
Left\$(string, length)	Из строки string возвращает length символов, начиная с
	крайнего слева символа (тип результата — string)
LeftB(string, length)	Из строки string возвращает length байт, начиная слева
	(тип результата — variant)
LeftB\$(string, length)	Из строки string возвращает length байт, начиная слева
	(тип результата — string)
Len (string)	Количество символов в строке string
LTrim(string)	Возвращает строку string без пробелов в начале строки
	(тип результата — variant)
LTrim\$(string)	Возвращает строку string без пробелов в начале строки
	(тип результата — string)
Mid(string,	Возвращает 1 ength символов из строки string, начиная с
start,length)	позиции start (тип результата — variant)
Mid\$(string, start,	Возвращает length символов из строки string, начиная с
length)	позиции start (тип результата — string)
Right(string, lengtn)	Возвращает length символов справа из строки string (тип
	результата — variant)
Right\$(string, length)	Возвращает lengtn символов справа из строки string (тип
	результата — string)
RightB(string,	Возвращает lengtn байт справа из строки string (тип ре-
length)	зультата — Variant)
RightB\$(string,	Возвращает lengtn байт справа из строки string (тип ре-
length)	зультата — String)
RTrim( string)	Возвращает строку string без пробелов в конце строки
	(тип результата — variant)
RTrim\$(string)	Возвращает строку string без пробелов в конце строки
	(тип результата — string)
Space(number)	Возвращает строку длиной number символов, заполнен-
	ную пробелами (тип результата — variant)
Space\$(number)	Возвращает строку длиной number символов, заполнен-
	ную пробелами (тип результата — string)
Split(expression, de-	Возвращает массив, состоящий из подстрок строки ех-
limiter)	pression, разъединенных разделителем delimiter
Str(number)	Строковое представление числа number (тип результата
	— variant)
Str\$(number)	Строковое представление числа number (тип результата
	— String)
StrComp(stringl,	Результат сравнения строк stringl и string
string2, compare)	

Продолжение таблицы Г.3

1	2
String(number, charac-	Возвращает символ character, который повторяет number
ter)	раз (тип результата — variant)
StrReverse(expression)	Возвращает строку, состоящую из символов строки ех-
	pression, записанных в обратном порядке
String\$(number, char-	Возвращает символ character, который повторяет number
acter)	раз (тип результата — string)
Trim(string)	Возвращает строку string без пробелов в начале и в кон-
	це строки (тип результата — variant)
Trim\$(string)	Возвращает строку string без пробелов в начале и в кон-
	це строки (тип результата — string)
UCase(string)	Возвращает строку string, преобразованную в верхний
	регистр (тип результата — Variant)
UCase\$(string)	Возвращает строку string, преобразованную в верхний
	регистр (тип результата — string)
Val(string)	Возвращает число, которое содержится в строке string

Таблица Г.3. Финансово-математические функции VBA

Функция	Возвращаемое значение
1	2
DDB (стоимость, стра-	Значение амортизации актива за определенный пери-
ховка, длит_жизни, пе-	од, вычисленное методом двойного уменьшения
риод, фактор)	остатка
FV(Rate, Nper, Pmt, Pv,	Будущая стоимость инвестиции или ссуды
Type)	
IPMT(Rate, Per, Nper,	Сумма платежей по процентам за определенный пери-
Pv, Fv, Type)	од для ссуды
IRR(значение, оценка)	Внутренняя ставка доходности для ряда денежных по-
	ТОКОВ
MIRR(значение,	Модифицированная внутренняя ставка доходности
ставка_доходности,)	для ряда периодических денежных потоков
NPER(Rate, Pmt, Pv, Fv,	Количество периодов для инвестиции или ссуды
Type)	
NPV (Rate, значение1,	Чистая приведенная стоимость инвестиции, вычис-
значение2)	ленная на основе ряда денежных потоков и ставки
	дискон тирования
PMT(Rate, Nper, Pv,	Размер периодической выплаты по ссуде или инвести-
Fv, Type)	ции
PPMT(Rate, период,	Размер основной выплаты за определенный период по
Nper, Pv, Fv, Type)	ссуде

продолжение таолицы 1.5		
1	2	
PV(Rate, Nper, Pv, Fv,	Приведенная стоимость инвестиции	
Type)		
RATE(Nper, Pmt, Pv,	Периодическая процентная ставка для ссуды или ин-	
Fv, Туре, оценка)	вестиции	
SLN(стоимость, стра-	Величина амортизации актива за один период, вычис-	
ховка, длит_жизни)	ленная линейным методом	
SYD(стоимость, стра-	Величина амортизации актива за один период,	
ховка, длит_жизни, пе-	ленная методом «суммы годовых чисел»	
риод)		