

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

Н. Н. Манаева, Е. А. Мучкаева, Э. И. Мурзаханова

ОСНОВЫ ПРОГРАММИРОВАНИЯ В СРЕДЕ VISUAL BASIC FOR APPLICATIONS

Учебное пособие

Рекомендовано ученым советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательным программам высшего образования по направлению подготовки 08.03.01 Строительство

Оренбург
2021

УДК 004.42(075.8)
ББК 32.973я73
М 23

Рецензент - доцент, кандидат педагогических наук М.И. Глотова

М 23 **Манаева Н.Н.**
Основы программирования в среде Visual Basic for Applications:
учебное пособие/ Н.Н. Манаева, Е.А. Мучкаева, Э.И. Мурзаханова;
Оренбургский гос. ун-т. – Оренбург: ОГУ, 2021. - 116 с.
ISBN

Учебное пособие содержит теоретическую, практическую часть, варианты индивидуальных заданий, ориентированные на реализацию Государственного образовательного стандарта по дисциплине «Информатика» (раздел основы программирования), рекомендовано для студентов строительных вузов и факультетов, обучающихся по направлению подготовки 08.03.01 Строительство.

УДК 004.42(075.8)
ББК 32.973я73

© Манаева Н.Н., 2021
Мучкаева Е.А., 2021
Мурзаханова Э.И., 2021
© ОГУ, 2021

ISBN

Содержание

Введение.....	5
1. Понятие алгоритма и программы. Языки программирования.....	7
1.1 Понятие и свойства алгоритма	7
1.2 Виды алгоритмов.....	9
1.3 Классификация языков программирования	11
1.4 Вопросы для самоподготовки	14
2 Объектно-ориентированная интегрированная среда разработки приложений	
VBA. Макросы.....	15
2.1 Основные понятия объектно-ориентированного программирования	15
2.2 Интерфейс VBA.....	16
2.3 Алфавит VBA	19
2.4 Использование макросов	20
2.5 Лабораторная работа №1 «Создание и выполнение макросов».....	24
2.6 Вопросы для самопроверки.....	35
3 Основные понятия VBA. Программирование алгоритмов линейной	
структуры. Создание форм.....	36
3.1 Синтаксис процедур Sub и Function	36
3.2 Данные, константы, переменные, идентификаторы.....	38
3.3 Типы данных.....	40
3.4 Ввод и вывод данных.....	41
3.5 Операторы и выражения языка VBA	44
3.6 Лабораторная работа №2 «Программирование алгоритмов линейной	
структуры при помощи пользовательских функций и форм».....	48
3.7 Вопросы для самопроверки.....	58
4 Программирование разветвляющихся вычислительных процессов.....	59
4.1 Инструкция If...Then	59
4.2 Инструкция If...Then...Else	60

4.3 Инструкция Select Case.....	64
4.4 Лабораторная работа №3 «Программирование алгоритмов разветвляющейся структуры».....	65
4.5 Вопросы для самопроверки.....	74
5 Программирование циклических вычислительных процессов.....	75
5.1 Команды организации циклов	75
5.2 Циклы с фиксированным числом итераций	75
5.3 Неопределенные циклы	80
5.4 Вложенные циклы	84
5.5 Лабораторная работа №4 «Программирование алгоритмов циклической структуры»	87
5.6 Вопросы для самопроверки.....	93
6 Одномерные массивы в среде VBA	94
6.1 Общие сведения.....	94
6.2 Лабораторная работа № 5 «Обработка одномерных массивов в среде VBA»	97
6.3 Вопросы для самопроверки.....	102
7 Двумерные массивы в среде VBA	102
7.1 Общие сведения о двумерных массивах.....	102
7.2 Лабораторная работа № 6 «Обработка двумерных массивов в среде VBA»	104
7.3 Вопросы для самопроверки.....	114
Список использованных источников	115

Введение

В настоящее время существует множество источников для изучения основ программирования в среде Visual Basic for Applications, ориентированных на пользователей разного уровня подготовки. Студенту, осваивающему дисциплину «Информатика», трудно выбрать литературу для изучения языка программирования самостоятельно. Учебное пособие предназначено для оказания помощи студентам строительного факультета при изучении основ программирования в среде Visual Basic for Applications.

В учебном пособии содержится теоретический материал, а также задания для выполнения лабораторных работ во 2 семестре. Каждая тема пособия сопровождается контрольными вопросами, позволяющими студентам самостоятельно определить уровень освоения материала.

Все предложенные примеры решения задач сопровождаются наглядными иллюстрациями, что облегчает студентам понимание материала. Задания лабораторных работ носят профессионально ориентированный характер, что способствует интеграции знаний, повышает познавательный интерес и мотивацию студентов.

Варианты индивидуальных заданий охватывают достаточно разнообразные типы примеров, которые позволяют закреплять теоретические знания и практические навыки по учебному курсу. Кроме того, предложенный теоретический материал и подробные практические примеры, позволяют студентам самостоятельно изучить ряд вопросов, связанных с освоением приемов решения задач в среде Visual Basic for Applications.

В учебном пособии рассматриваются сведения необходимые при разработке алгоритмов и программ на языке программирования высокого уровня Visual Basic for Applications. Данные сведения содержат материал, позволяющий изучить:

- ввод и вывод данных;
- составление арифметических выражений;
- создание и выполнение макросов;

- программирование разветвляющих алгоритмических структур;
- программирование циклических алгоритмических структур, использующих различные типы циклов: циклы с предусловием, циклы с постусловием, циклы с параметром и вложенные циклы;
- составление программ с использованием одномерных и многомерных массивов.

Учебное пособие предназначено для бакалавров направления подготовки 08.03.01 Строительство, изучающих основы программирования в среде Visual Basic for Applications. Пособие также будет полезно для студентов первых курсов, обучающихся по образовательным программам высшего образования на инженерных непрофильных направлениях.

1. Понятие алгоритма и программы. Языки программирования

1.1 Понятие и свойства алгоритма

Алгоритм – это точное предписание, определяющее последовательность действий исполнителя, направленных на решение поставленной задачи.

Слово «алгоритм» появилось в результате искаженного перевода с арабского на европейские языки имени узбекского ученого IX века Аль-Хорезми, который изложил правила арифметических действий над числами в десятичной системе. Эти правила и называли алгоритмами (Альхорезми «имя»+ Аритмос «число»= алгоритм).



Рисунок 1 – Узбекский ученый IX века Аль-Хорезми

Свойства алгоритма:

Дискретность. Описываемый процесс должен представлять собой упорядоченную совокупность четко разделенных друг от друга предписаний, образующих прерывную (дискретную) структуру алгоритма.

Понятность. Предписания должны быть написаны на понятном исполнителю языке.

Результативность. Возможность получения результата после выполнения конечного количества операций




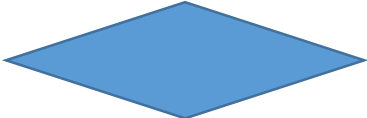
Определенность. Совпадение получаемых результатов независимо от пользователя и применяемых технических средств.

Массовость. Возможность применения алгоритма к целому классу однотипных задач, различающихся конкретными значениями исходных данных.


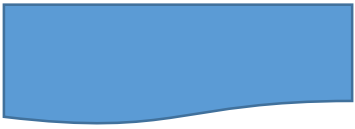
В зависимости от поставленной задачи и исполнителя алгоритм может быть записан различными способами. К наиболее часто используемым способам записи алгоритма относят словесно-формульный, графический, в виде блок-схем, с помощью таблиц.

Для обеспечения возможности реализации на ЭВМ алгоритм должен быть записан на языке понятном компьютеру, т.е. на **языке программирования**. При составлении программы удобно использовать блок-схемы. Основные блоковые конструкции, используемые при записи алгоритма представлены в таблице 1.

Таблица 1 – Виды блоков, используемых при записи алгоритма

Блок	Название, назначение
1	2
	Пуск-останов. Блок используется для обозначения начала и конца алгоритма
	Ввод-вывод. Блок используется при вводе и выводе данных алгоритма
	Процесс. Блок отображает вычислительный процесс по стандартной программе или подпрограмме
	Решение. Блок используется для описания ветвления, условного алгоритма

Продолжение таблицы 1

1	2
	<p>Границы цикла. Блоки используются для описания начала и конца цикла.</p>
	<p>Документ. Блок указывает результаты алгоритма, выводимые на печать</p>

1.2 Виды алгоритмов

Любой вычислительный процесс может быть представлен тремя основными видами алгоритмов: линейным, разветвляющимся и циклическим.

Линейным принято называть алгоритм, в котором действия (операторы) выполняются последовательно, в порядке их записи.

Каждый оператор является самостоятельным и обозначается на схеме отдельным блоком (рисунок 2).

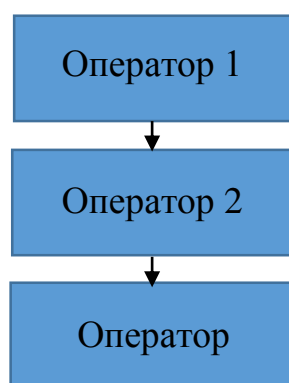


Рисунок 2 – Блок-схема линейного алгоритма

Разветвляющимся (условным) называют алгоритм, содержащий хотя бы одно условие, в результате которого обеспечивается переход на один из двух возможных шагов.

При реализации условных алгоритмов используют схемы полного и неполного ветвления. При полном ветвлении алгоритм работает следующим образом: проверяется истинность *Условия*, если условие истинно выполняется *Оператор 1*, если ложно – *Оператор 2* (рисунок 3а). При неполном ветвлении если *Условие* истинно выполняется *Оператор*, если ложно – управление передается следующему за условием оператору (рисунок 3б).

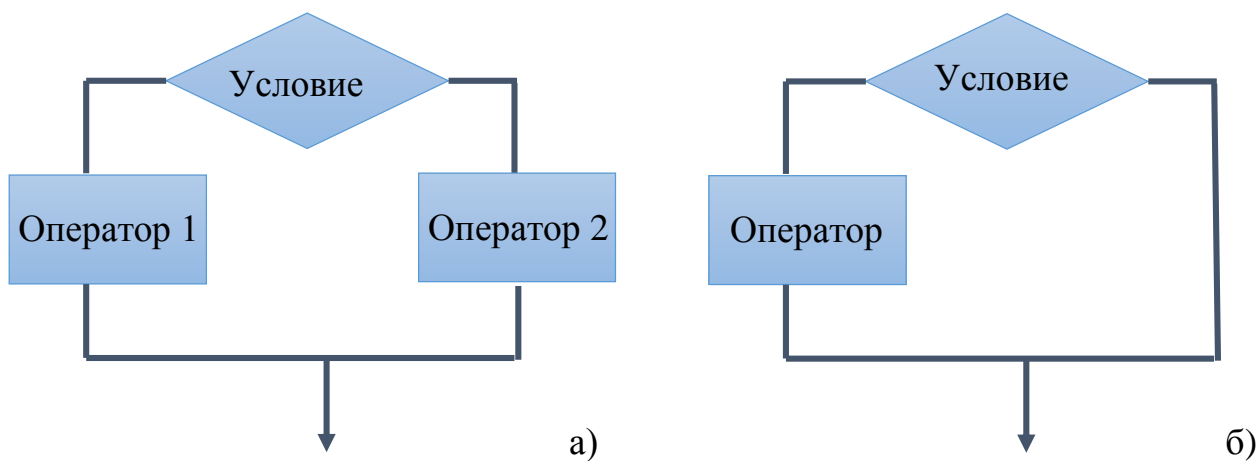


Рисунок 3 – Блок-схемы: а) полного ветвления; б) неполного ветвления

Циклическим называется алгоритм, предусматривающий многократное повторение одного и того же действия над новыми данными.

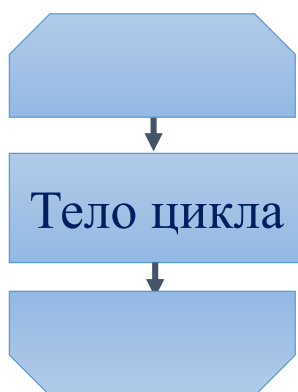


Рисунок 4 – Блок-схема циклического алгоритма

В зависимости от того, как определяется количество повторений цикла выделяют два вида циклов: арифметический (итерационный) и условный.

Цикл, число повторений которого известно или может быть вычислено называется *арифметическим*.

Цикл, число повторений которого не известно называется *условным*.

В том случае, когда условие стоит перед телом цикла, цикл называется с *предусловием*. При реализации такого цикла сначала проверяется истинность условия, если условие истинно выполняется тело цикла, если ложно – цикл завершается. Если же условие ложно изначально цикл не выполнится ни разу.

Когда условие стоит после тела цикла, цикл называется с *постусловием*. При реализации данного вида цикла сначала выполняется тело цикла, а затем проверяется условие. Если условие ложно идет возврат к телу цикла, если истинно – цикл завершается. Если же условие истинно изначально цикл выполнится один раз.

1.3 Классификация языков программирования

Язык программирования – это система обозначений, служащая для точного описания программ или алгоритмов для ЭВМ.

Поскольку любой алгоритм представляет собой последовательность предписаний, то в зависимости от детализации предписаний определяют уровень языка программирования. Если детализация языка высокая, то такие языки называют языками *низкого уровня*, к ним относятся *машинные* и *машинно-ориентированные языки (ассемблеры)*.

Машинные и машинно-ориентированные языки обращаются напрямую к устройствам машины, следовательно, требуют подробного описания самых мелких деталей процесса обработки данных. Языки низкого уровня были первой ступенью в развитии языков программирования, они имеют как свои достоинства, так и недостатки. К достоинствам можно отнести то, что программы, созданные с использова-

нием языков низкого уровня, имеют небольшой размер и высокую степень эффективности, так как разработчик получает доступ ко всем возможностям процессора. Недостатками языков низкого уровня являются трудности, возникающие при отладке больших приложений, а также то, что программа не может быть перенесена на компьютер с другим типом процессора, поскольку она машинно-зависимая. В современных реалиях языки низкого уровня применяются при разработке небольших системных приложений, драйверов, модулей стыковки с нестандартным оборудованием и др.

Следующей ступенью в развитии языков программирования стали языки программирования **высокого уровня**. В отличие от языков низкого уровня, языки программирования высокого уровня используют некоторые термины обычного языка и общепринятые математические символы. Они универсальны и независимы от аппаратуры – программа, написанная на таком языке, может выполняться на разных машинах, что облегчает работу программиста и повышает надежность создаваемых продуктов. В настоящее время насчитывается более 2000 языков программирования высокого уровня, их можно классифицировать по их возможностям, назначению и способам реализации алгоритмов.

Первые языки высокого уровня были **процедурными** языками программирования, которые предназначены для однозначного описания алгоритмов и требуют в той или иной форме явно записать порядок (процедуру) решения.

Изначально процедурные языки программирования создавались под конкретную задачу, такие языки назывались **проблемно-ориентированными** языками программирования. К наиболее популярным проблемно-ориентированным языкам относят: *Fortran* и *Algol* (научно-технические задачи), *Cobol* (экономические задачи), *Basic* (вычислительные задачи). Затем разрабатываются **универсальные** процедурные языки программирования, они объединяют возможности проблемно-ориентированных языков и подходят для более широкого спектра задач. Примерами универсальных языков служат такие языки, как *Pascal*, *C*, *C+*, *Ada*. Далее развитие языков высокого уровня происходит по нескольким направлениям.

Одно из направлений – это **логические** языки программирования, которые ориентированы не на запись алгоритма задачи, а на систематическое и формализованное ее описание. Требуемое решение логически вытекает из составленного описания. В таких языках указывается, что дано и что требуется получить, а поиск решения возлагается на компьютер. Примеры: *Prolog, Lisp, Mercury*.

Другой тенденцией развития высокого уровня являются **объектно-ориентированные** языки, базирующиеся на моделировании предметной области при помощи объектов. Объектно-ориентированное программирование можно рассматривать как модульное программирование, в основе которого лежит понятие **объекта**. Объект представляет собой некоторую сущность, объединяющую в себе **поля** (данные) и **методы** (выполняемые объектом действия). Наиболее известные объектно-ориентированные языки: *Object Pascal, C++, Java*.

Третьим направлением развития языков высокого уровня можно считать **языки запросов**, которые позволяют пользователю получать информацию из баз данных. Языки запросов имеют свой особый синтаксис, который должен соблюдаться, как и в традиционных языках высокого уровня, но при этом проще в использовании. Среди языков запросов фактическим стандартом стал язык *SQL*.

К интенсивно развивающимся направлениям языков программирования относятся языки **параллельного программирования**, ориентированные на создание программного обеспечения для вычислительных средств параллельной архитектуры (модификация *Fortran*, языки *Occam, SISAL, FP* и др.) и **языки искусственного интеллекта**, экспертных систем, баз знаний (*InterLisp, ExpertLisp, IQLisp, SAIL* и др.).

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

Алфавит – это фиксированный для данного языка набор основных символов, из которых должен состоять любой текст на этом языке - никакие другие символы в тексте не допускаются.

Синтаксис – правила построения фраз, устанавливающих, какие комбинации символов являются осмысленными предложениями.

Семантика – определяет смысловое значение предложений языка. Устанавливает, какие последовательности действий описываются теми или иными фразами языка и, в конечном итоге, какой алгоритм определен данным текстом на алгоритмическом языке.

1.4 Вопросы для самоподготовки

1. Что такое алгоритм, какими свойствами он обладает?
2. Какие способы записи алгоритмов вы знаете?
3. Перечислите и охарактеризуйте основные блоки, используемые при составлении блок-схем.
4. Какой алгоритм называется линейным, каким образом он изображается в блок-схеме?
5. Какой алгоритм называется разветвляющимся (условным)? Как работает и отображается на блок-схеме алгоритм с полным и неполным ветвлением?
6. Какой алгоритм называется циклическим, чем отличаются инерционный и условный цикл?
7. В чем особенности циклов с предусловием и постусловием?
8. Что такое языки программирования? От чего зависит уровень языков программирования?
9. Выделите особенности, достоинства и недостатки языков программирования низкого уровня.
10. Чем языки высокого уровня отличаются от языков низкого уровня?
11. Охарактеризуйте процедурные проблемно-ориентированные и универсальные языки программирования, приведите примеры.
12. Назовите наиболее известные логические языки программирования, в чем их особенность?
13. В чем особенность объектно-ориентированного программирования? Приведите примеры объектно-ориентированных языков.

14. Для чего используются языки запросов? Какой из них наиболее популярный?

15. Назовите тенденции развития современных языков программирования.

16. Что такое алфавит синтаксиса и семантика языка программирования?

2 Объектно-ориентированная интегрированная среда разработки приложений VBA. Макросы

2.1 Основные понятия объектно-ориентированного программирования

VBA (*Visual Basic for Applications*) – это один из наиболее доступных языков программирования, который представляет собой упрощенную реализацию языка программирования *Visual Basic*.

Он интегрирован в линейку продуктов Microsoft Office, также во многие другие программные пакеты, такие как *AutoCAD*, *SolidWorks*, *CorelDRAW*, *Word-Perfect* и т.д. **VBA** позволяет автоматизировать процессы выполнения различных операций и пользовательских действий, конструировать профессиональные приложения, решающие сложные практические задачи.

VBA поддерживает три стиля программирования: визуальное, событийно-управляемое и объектно-ориентированное программирование.

Визуальное программирование позволяет разработчику самому создавать интерфейс, помещая мышью нужные элементы управления – кнопки, строки ввода, списки выбора, картинки и другие элементы, привычные ему по работе в Windows.

Событийно-управляемое программирование представляет собой технологию, при которой выполнение блоков программы происходит не в заранее определенном порядке, а в ответ на события, источником которых является пользователь (наведение курсора, одинарный или двойной клик мыши и др.).

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

Объект отражает определенную самостоятельную сущность предметной области со своими данными и своими возможностями.

На объект можно воздействовать вызовом принадлежащих объекту процедур - методов, или установкой его параметров и характеристик – **свойств**.

Класс – представляет собой множество объектов, имеющих общую структуру и общее поведение.

Событие – это результат одного или нескольких внешних воздействий. К простейшим событиям относятся: щелчок, двойной щелчок, перетаскивание, нажатие на определенную кнопку и т.д.

Каждому объекту имеет соответствовать: свойства, методы, события. В качестве примера рассмотрим объект круг, его свойства, методы и допустимые события представлены на рисунке 5.

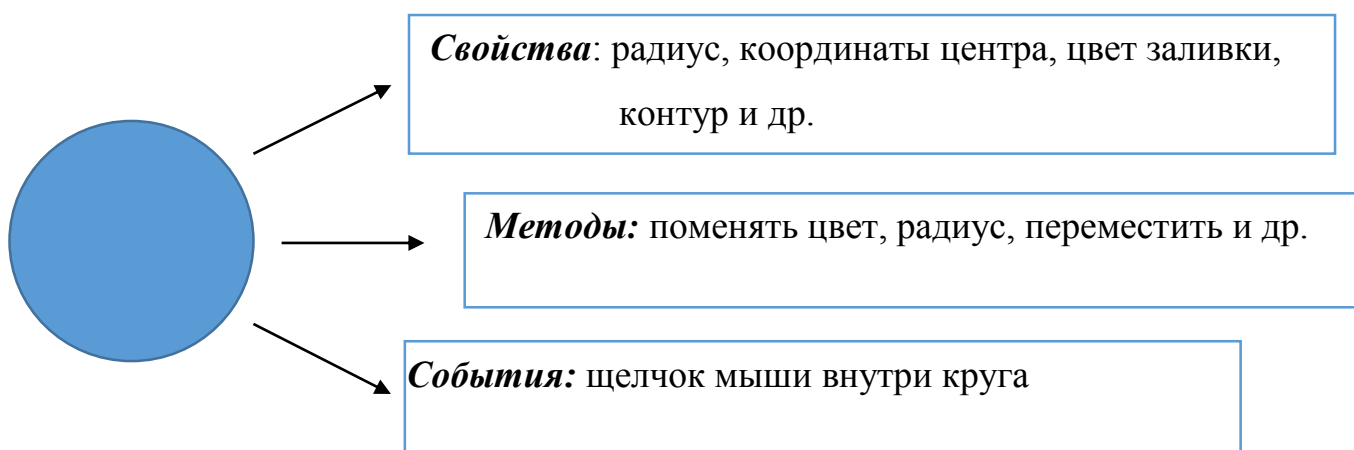


Рисунок 5 – Свойства, методы и события, характерные для объекта «круг»

2.2 Интерфейс VBA

Запустить VBA можно в любом приложении пакета MS Office, используя сочетанием клавиш Alt+F11, либо кнопку **Visual Basic** на вкладке **Разработчик** (рисунок 6).

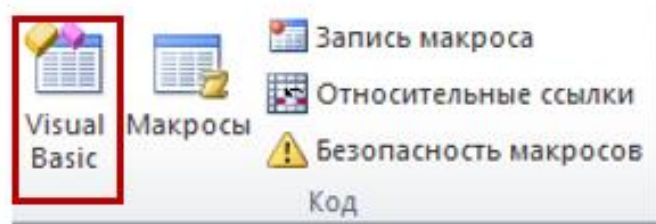


Рисунок 6 – Кнопка запуска среды VBA

После запуска открывается окно редактора VBA, интерфейс которого представлен на рисунке 7.

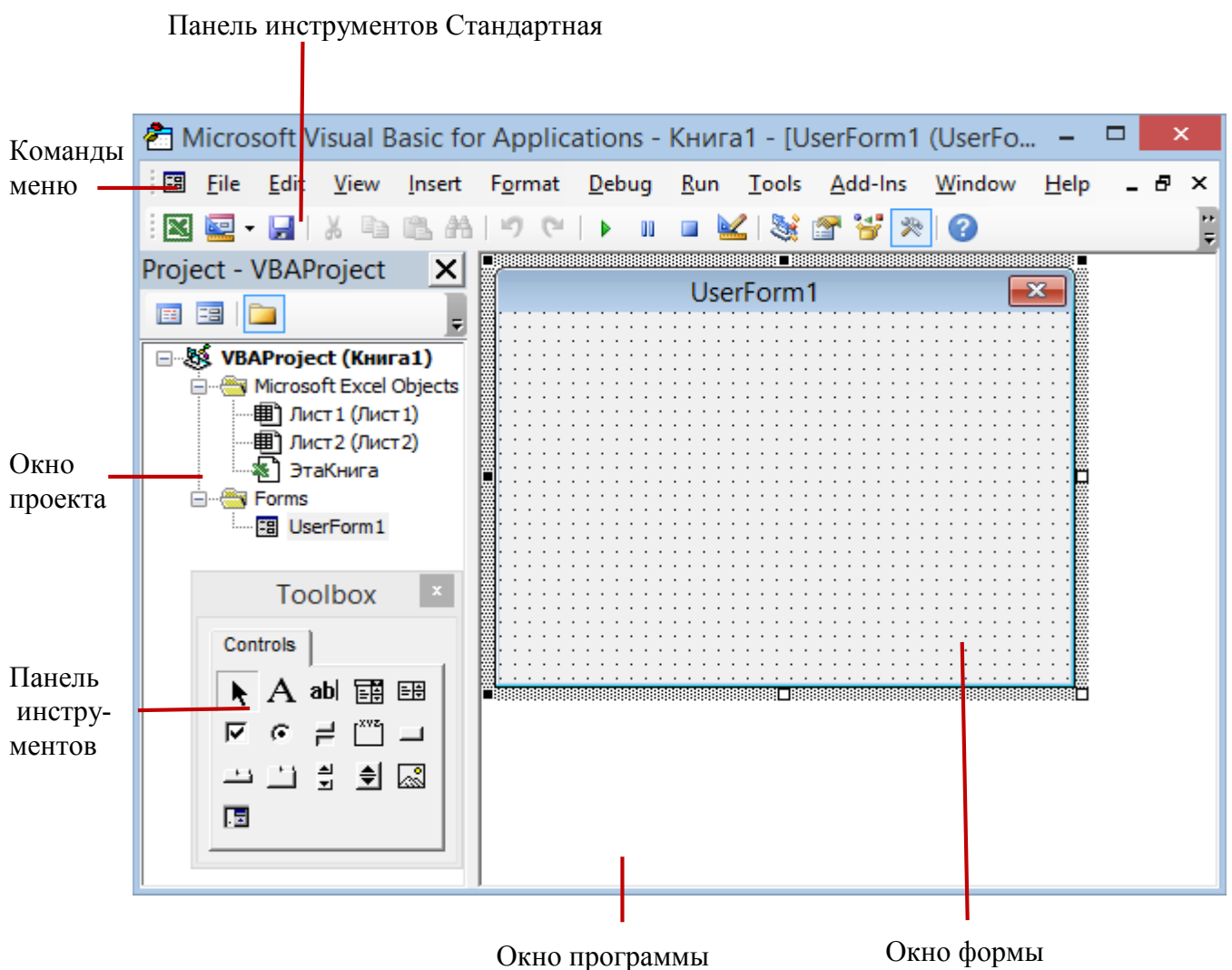


Рисунок 7 – Интерфейс VBA

Проект – часть программы, которая видна на экране при ее создании.

Проект VBA имеет иерархическую структуру и включает: объекты, формы, стандартные модули и модули классов.

Объекты Microsoft Excel - рабочие книги (WorkBooks), рабочие листы (WorkSheets), диаграммы (Charts).

С каждым из этих объектов связан специальный модуль, в который может быть помещен программный код, выполняющий определенные действия.

Формы – пользовательские окна диалога, создаваемые с помощью встроенных инструментов VBA для представления пользовательских данных. Каждая форма содержит модуль VBA с кодом обработки событий.

Модули – содержат большинство подпрограмм и функций, составляющих VBA-приложение. Модуль создается автоматически при первой записи макроса в рабочей книге.

Модули класса – собственные объекты, создаваемые и используемые в макросах.

Проект включает две части:

- **интерфейсную** – видимую при выполнении программы;
- **программную** – находится в различных модулях и реализует выполнение заданных разработчиком действий.

Весь проект, включая классы модулей, модули и пользовательские формы, сохраняется вместе с рабочей книгой в особом формате - типе файла MS Excel с поддержкой макросов (*.xlsm).

Большая часть работ при программировании выполняется в окне программы (рисунок 8).

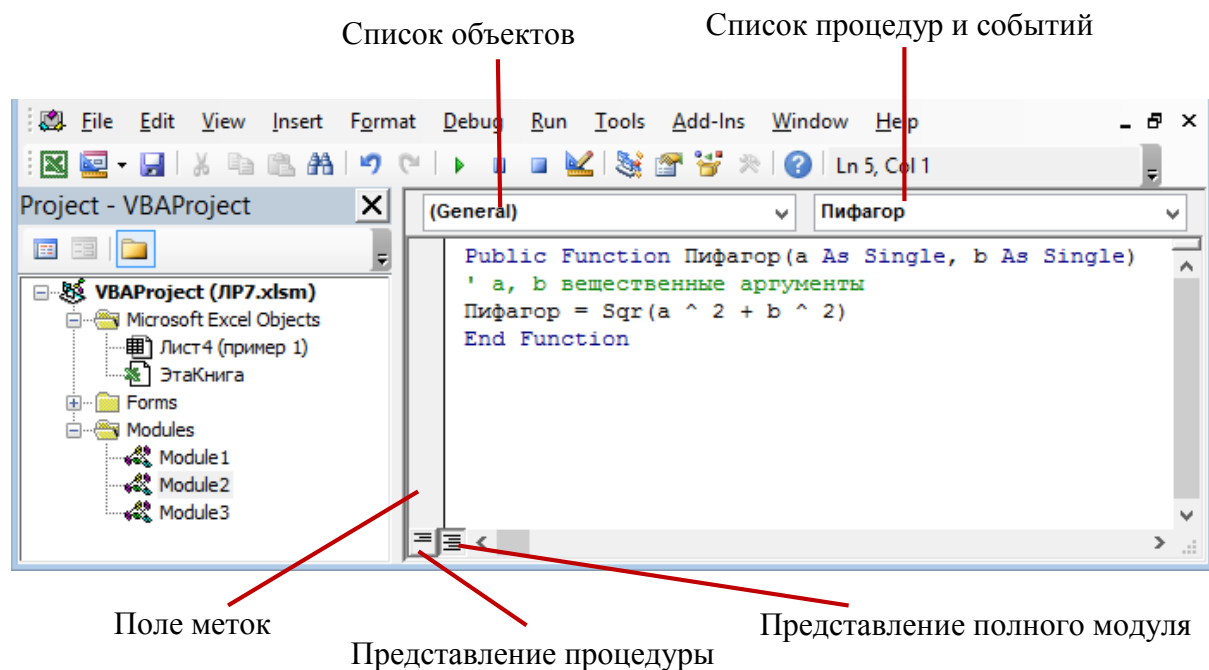


Рисунок 8 – Окно программы с записанным текстом программы.

2.3 Алфавит VBA

Символы из алфавита языка используются для построения базовых элементов программы – лексем.

Лексема – минимальная единица языка, имеющая самостоятельный смысл.

Выделяют следующие классы лексем:

2. **Константы** – это числа, символьные строки и некоторые другие значения.
3. **Идентификаторы (имена)** вводятся для обозначения в программе переменных, констант, типов, меток, процедур и функций. Идентификатор представляет собой последовательность букв, цифр и символов подчеркивания, начинающуюся с буквы.
4. **Знаки операций.** Формируются из одного или нескольких символов и предназначены для задания действий по преобразованию данных и вычислению значений.
5. **Служебные (ключевые) слова.** Каждое служебное слово представляет собой неделимое образование, смысл которого фиксирован в языке.

6. Разделители. Используются для отделения лексем друг от друга, в тех случаях, когда их слитное написание может исказить текст программы. В качестве разделителей можно использовать: пробел, символ перевода строки, комментарий. В любом месте программы, где можно поместить один разделитель, их можно поместить любое количество и в любом сочетании.

Алфавит языка VBA составляют:

- прописные и строчные буквы латинского и русского алфавитов;
- арабские цифры: 0, ..., 9.
- операторы: арифметические, текстовые, сравнения.

2.4 Использование макросов

При работе с приложениями MS Office часто возникает необходимость многократно выполнять одну и ту же последовательность действий. В этом случае можно использовать макрос, с помощью которого можно автоматизировать выполнение данных действий. Для создания и редактирования макросов используется редактор VBA.

Макрос – это запрограммированная последовательность действий, записанная на языке программирования VBA.

Для работы с макросом используется панель **Разработчик** (рисунок 9).

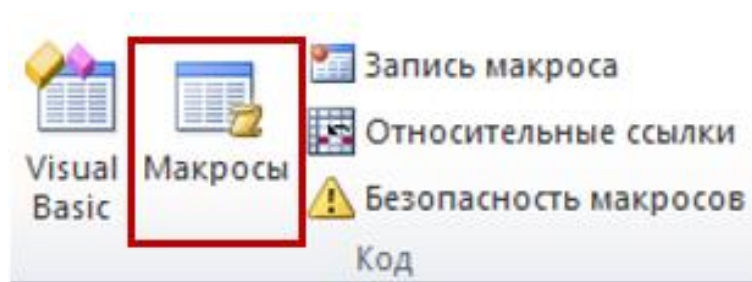


Рисунок 9 – Панель Разработчик

Макросы хранятся в программных модулях. В любой книге Excel можно создавать любое количество программных модулей, содержащих макросы. Один модуль

может содержать любое количество макросов. Доступ к модулям осуществляется с помощью окна *Project Explorer*, которое находится в верхнем левом углу редактора.

Программные модули могут быть следующих типов:

- обычный модуль;
- модуль книги;
- модуль листа.

Обычный модуль выбирается в большинстве случаев при работе с макросами. Для создания такого модуля используется команда *Module* в меню *Insert*. В появившемся окне нового модуля вводятся команды на VBA (рисунок 10).

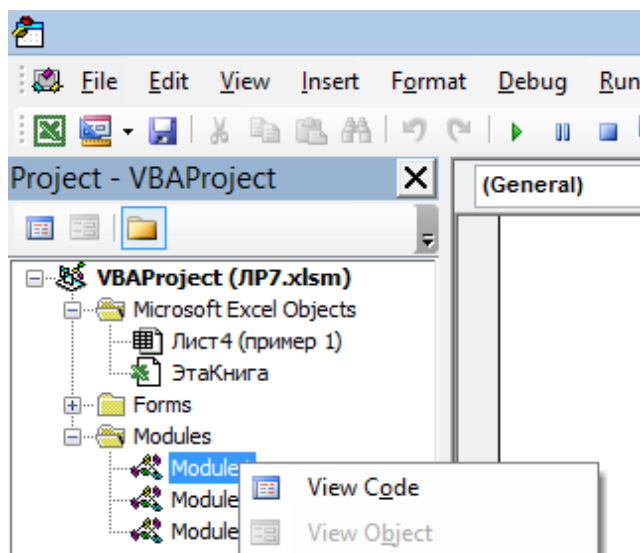


Рисунок 10 – Окно обычного модуля

Модуль книги называется *Эта книга*, он доступен через окно *Project Explorer*. В нем записаны макросы, выполняющиеся при наступлении каких-либо событий в книге (открытие или сохранение книги, печать файла и др.). Данный модуль представлен на рисунке 11.

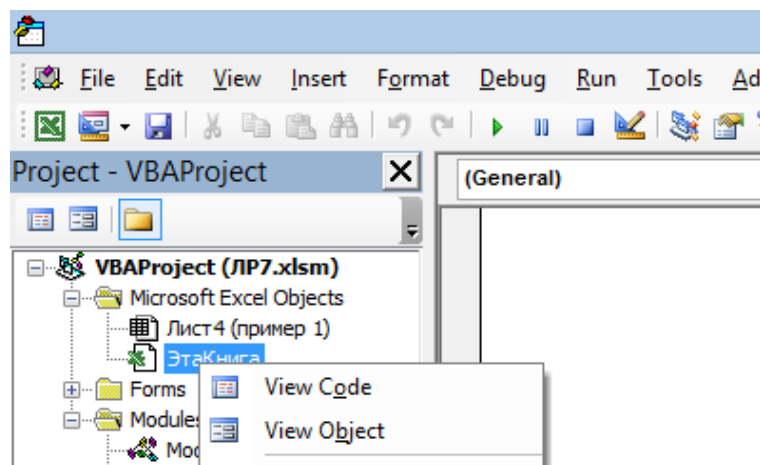


Рисунок 11 – Модуль книги

Модуль листа можно просмотреть через *Project Explorer*, либо через контекстное меню листа. Он содержит макросы, которые выполняются при наступлении определенных событий на листе (изменение данных в ячейках листа, пересчет значений листа, копирование, перемещение или удаление листа и др.). Данный модуль представлен на рисунке 12.

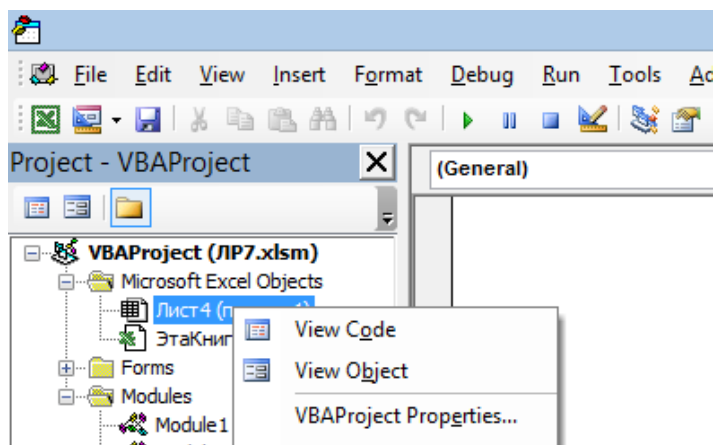


Рисунок 12 – Модуль листа

Любой макрос начинается с процедуры *Sub*, далее следует имя макроса и список аргументов в скобках (при отсутствии аргументов скобки остаются пустые). Затем идет тело макроса (команды, выполняющиеся при запуске макроса). В конце макроса используется служебное слово *End Sub*.

На рисунке 13 показан макрос, в теле которого происходит выделение ячейки (*Selection*) и заливка ее желтым цветом (код=6), далее в цикле по всем ячейкам, заменяются формулы на значения. В конце макроса выводится окно сообщения (*MsgBox*).

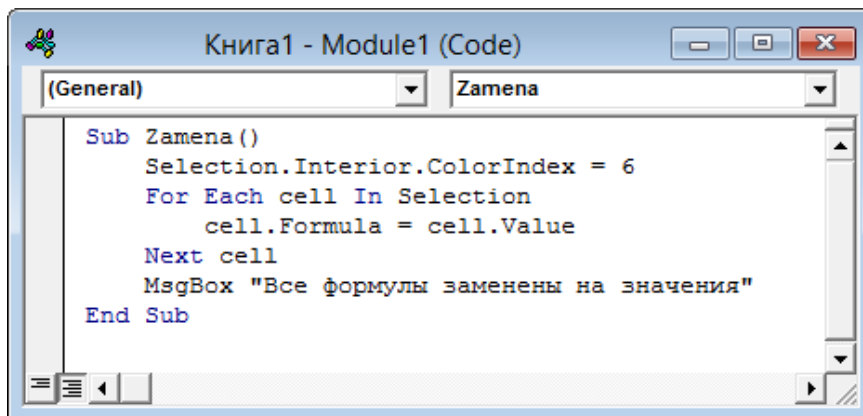


Рисунок 13 - Окно макроса

Создать макрос можно в редакторе VBA «вручную», либо с использованием макрорекордера.

Макрорекордер - программа, встроенная в приложение MS Office, которая переводит любое действие пользователя на язык программирования VBA и записывает получившуюся команду в программный модуль.

Использование макрорекордера не требует знаний о программировании и VBA. Пользователь лишь выполняет нужную последовательность действий, а рекордер сам переводит заданную последовательность программу.

К минусам использования макрорекордера относят:

1) Записываются только те действия, которые выполняются в пределах окна Microsoft Excel. Как только Excel закрывается или происходит переключение на другую программу - запись останавливается.

2) Записываются только те действия, для которых есть команды меню или кнопки в Excel.

3) Если во время записи макроса макрорекордером вы ошиблись - ошибка будет записана.

Замечание: В случае ошибки можно использовать кнопку отмены последнего действия (Undo) - во время записи макроса макрорекордером она не просто вернет в предыдущее состояние, но и удалит последнюю записанную команду на VBA.

Сохранить макрос можно в:

«Эта книга» - только в текущую книгу;

«Новая книга» – в шаблон, по которому создаются новые книги;

«Личная книга макросов» – в книгу с именем Personal.xls, используемую как хранилище макросов.

Запустить макрос можно следующими способами:

- по имени макроса на панели Разработчика;
- используя сочетание клавиш;
- с помощью кнопки на панели инструментов и панели быстрого доступа;
- с помощью кнопки на листе;
- с помощью любого графического объекта.

2.5 Лабораторная работа №1 «Создание и выполнение макросов»

Для создания макросов и работы в редакторе VBA требуется панель **Разработчик**. Если в главном меню отсутствует данная панель, ее можно добавить, используя следующую последовательность действий:

1. Раскрыть вкладку **Файл** главного меню, выбрать пункт **Параметры**. В появившемся окне выбрать пункт **Настроить ленту** (рисунок 14).

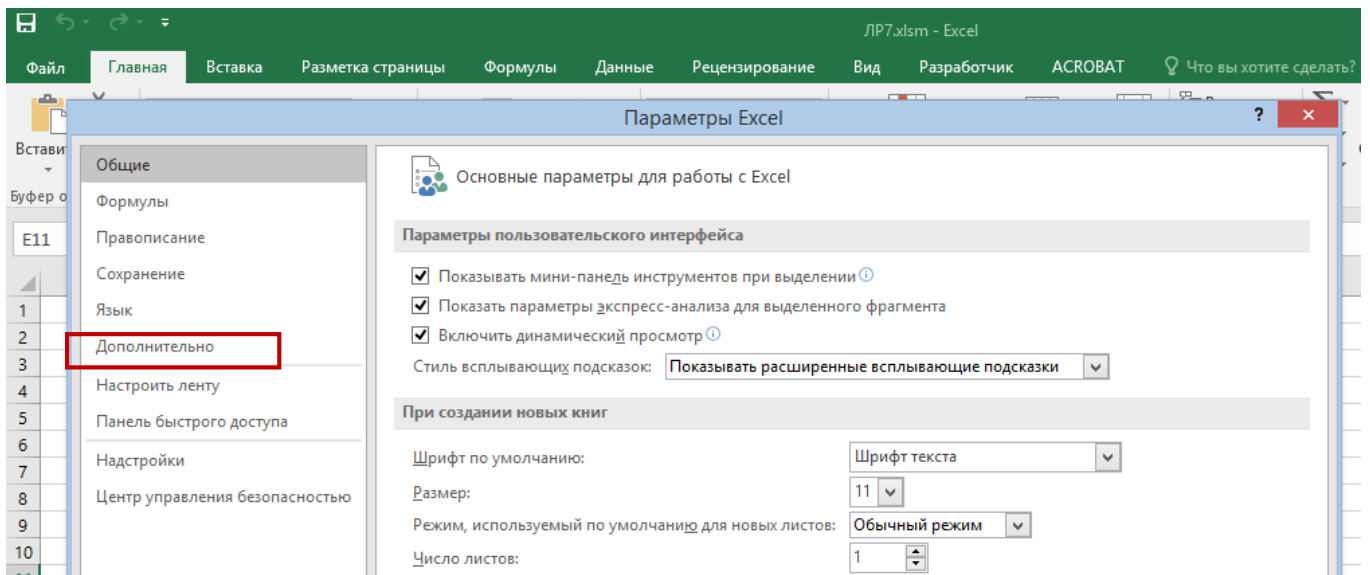


Рисунок 14 – Окно параметров

2. В настройке ленты в меню «**Выбрать команды**» выбрать пункт «**Все вкладки**», затем пункт «**Разработчик**». Нажать на кнопку «**Добавить**» и затем нажать на «**ОК**» (рисунок 15).

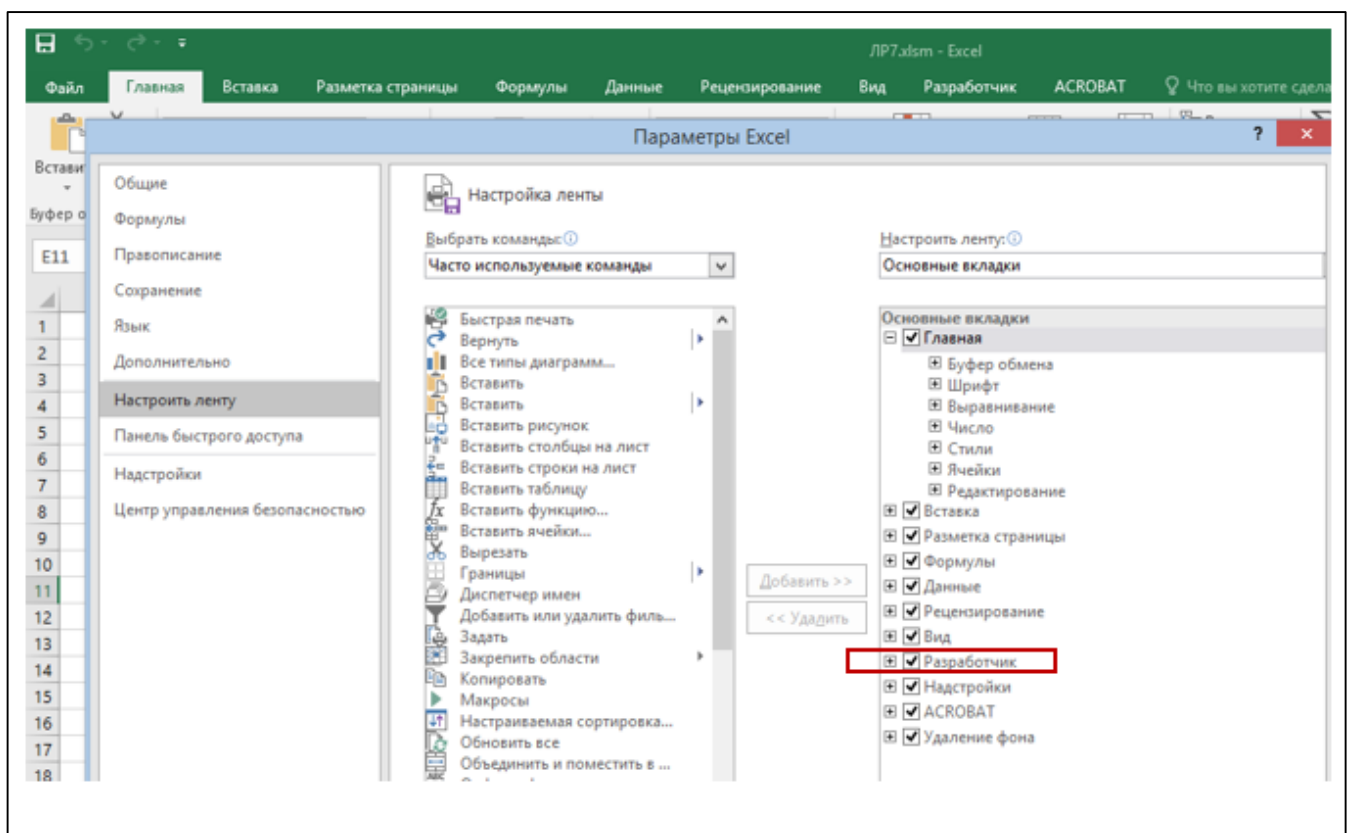


Рисунок 15 – Добавление панели Разработчик

Рассмотрим примеры создания макросов форматирования и макросов вычисления, а также различные способы их запуска.

Пример 1. Создать макрос форматирования для оформления таблицы.

Методика выполнения:

1. Открыть новую книгу в MS Excel. Выделить диапазон ячеек A1:C2, и нажать кнопку **Записать макрос** на вкладке **Код** панели **Разработчик**.

2. В диалоговом окне **Запись макроса** (рисунок 16) ввести имя макроса «Оформление_таблицы» (имя макроса должно начинаться с буквы и может содержать до 255 символов: буквы, цифры и знаки подчёркивания, пробелы не допускаются). Назначить для запуска макроса сочетание клавиш **Ctrl+q**, выбрать место сохранения – «**Эта книга**», в описании задать свою фамилию и дату выполнения работы и нажать «**ОК**».

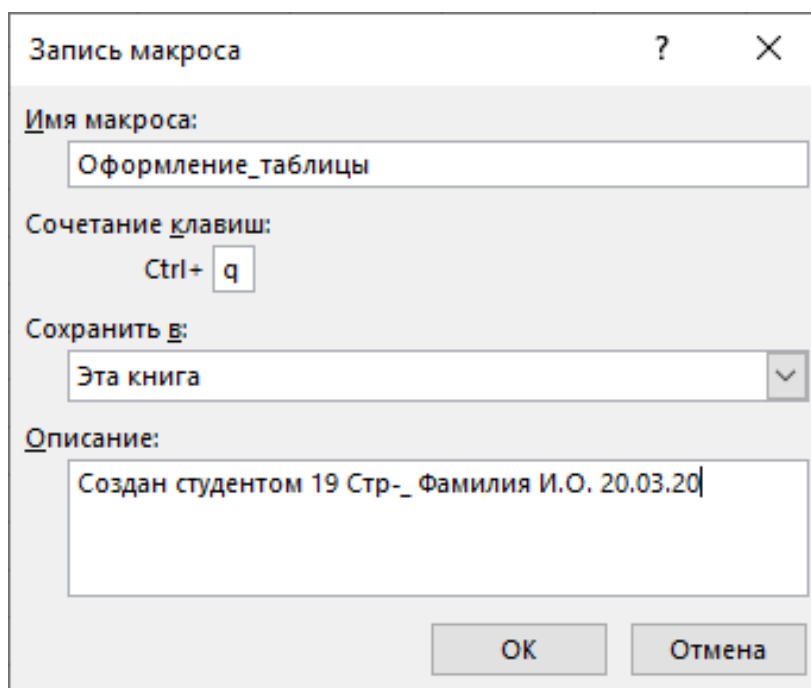


Рисунок 16 – Диалоговое окно «Запись макроса»

3. В контекстном меню выделенного диапазона выбрать команду **Формат ячейки**. В открывшемся диалоговом окне установить:

– на вкладке **Шрифт**: шрифт *Times New Roman*, размер – 16 пунктов, начертание – *курсив*;

- на вкладке **Выравнивание**: по горизонтали «по левому краю»;
- на вкладке **Границы**: внешняя толстой линией, внутренняя обычной.

4. Нажать «**ОК**» и кнопку «**Остановить запись**».

5. Заполнить несколько ячеек листа произвольными данными, выделить их и используя созданный макрос изменить их формат.

Для активации макроса можно воспользоваться комбинацией клавиш **Ctrl+q**, либо нажать кнопку **Макрос** на вкладке **Код** панели **Разработчик**, в появившемся диалоговом окне выбрать макрос «Оформление_таблицы» (рисунок 17) и нажать кнопку «**Выполнить**».

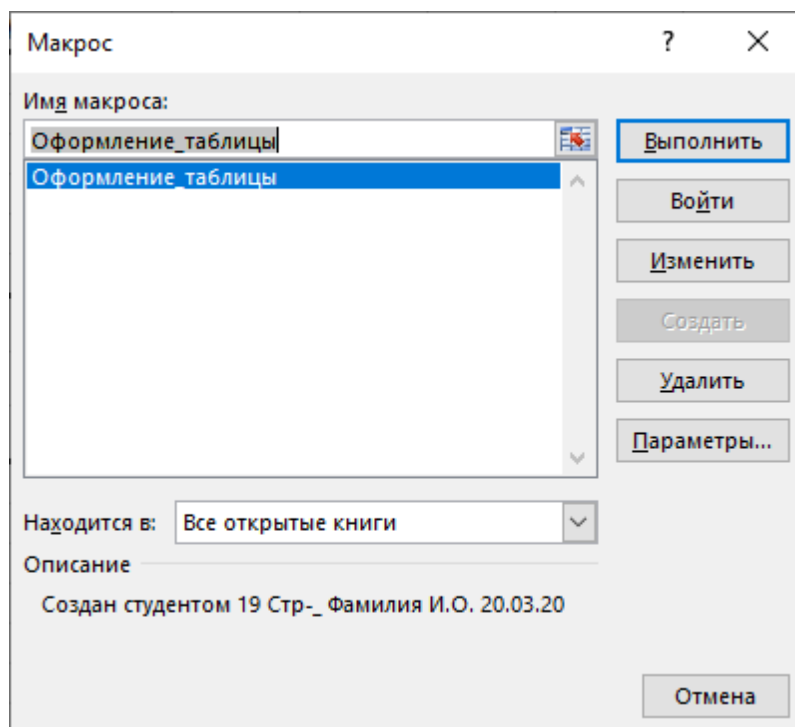


Рисунок 17 – Диалоговое окно «Макрос»

Чтобы просмотреть полученный при записи макроса код или, если требуется, отредактировать его, необходимо открыть диалоговое окно **Макрос** (рисунок 17), выделить имя нужного макроса и нажать кнопку «**Изменить**». Откроется окно редактора VBA (рисунок 18).

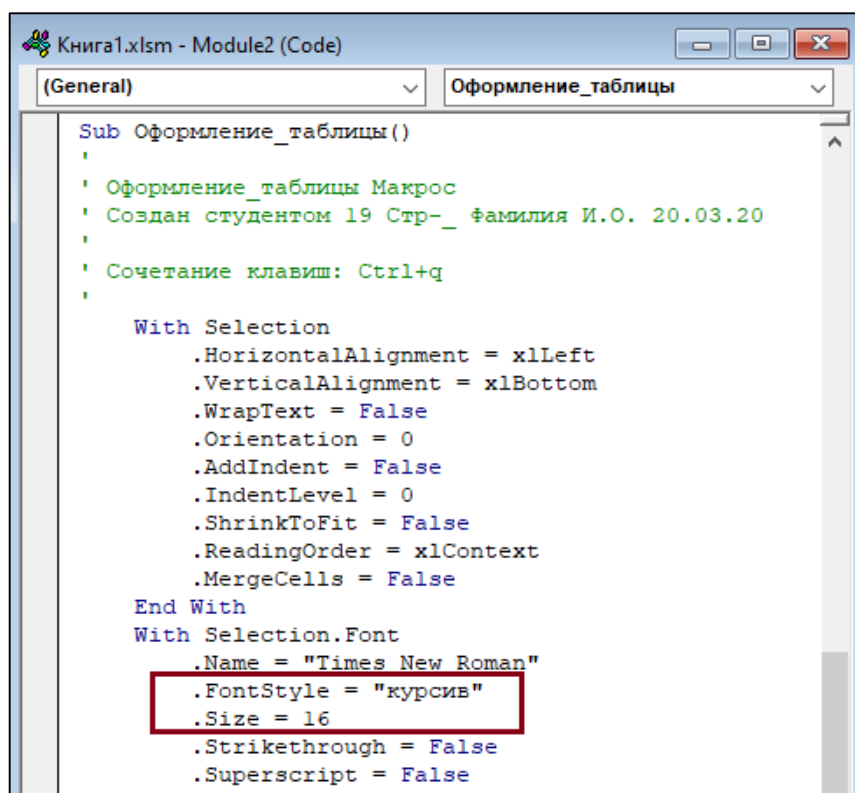


Рисунок 18 – Окно редактора VBA

В данном окне можно изменить заданные при записи макроса параметры, например, изменить размер шрифта, начертание и др.

Для запуска макроса на выполнение существуют несколько способов:

- запуск макроса через команду **Макросы** панели **Разработчик**;
- назначение комбинации клавиш для вызова макроса;
- назначение макроса командной кнопке;
- назначение макроса графическому изображению;
- назначение макроса кнопке на панели быстрого доступа.

Первые два способа были рассмотрены выше, рассмотрим остальные.

Для того чтобы создать **командную кнопку**, с помощью которой будет выполняться макрос, необходимо:

1. Открыть панель **Разработчик**, нажать кнопку **Режим конструктора** и с помощью кнопки **Вставить** выбрать элемент управления **Кнопка** (рисунок 19). Не отпуская левую кнопку мыши поместить указатель мыши в то место, где будет располагаться командная кнопка, и придать кнопке нужный размер.

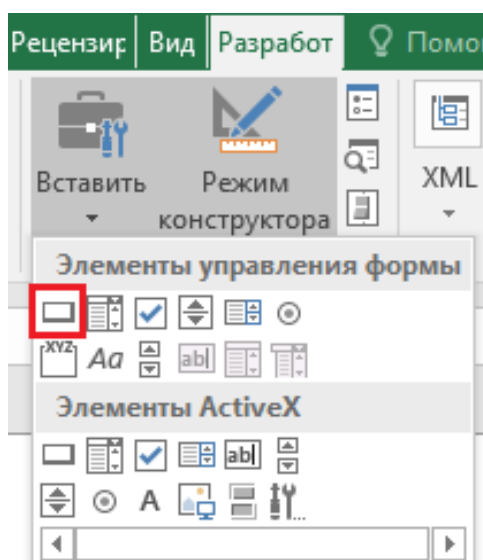


Рисунок 19 – Элементы управления формы

2. При добавлении кнопки откроется диалоговое окно *Назначить макрос объекту* (рисунок 20), где необходимо выбрать нужный макрос и нажать «ОК». Далее можно выделить надпись *Кнопка1* на командной кнопке и ввести название для данной кнопки, щёлкнуть вне кнопки, чтобы снять с неё выделение.

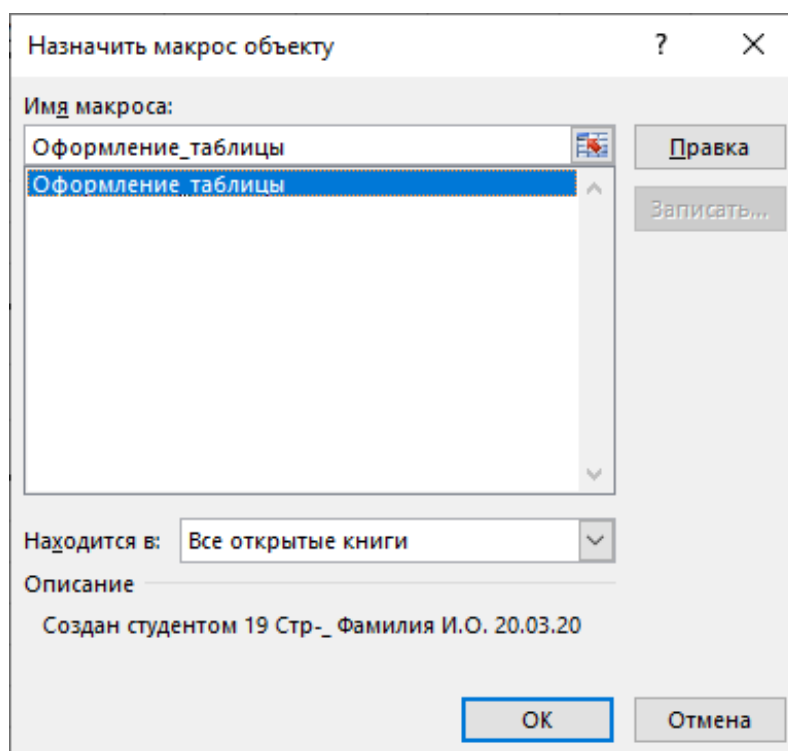


Рисунок 20 – Диалоговое окно «Назначить макрос объекту»

Назначить макрос *графическому изображению* можно следующим образом:

1. Используя панель **Вставка** вкладка **Иллюстрации** нарисовать на листе автофигуру или выбрать изображение для вставки на рабочий лист. Придать изображению соответствующие размеры.

2. Нажать на изображении правую кнопку мыши, в открывшемся контекстном меню выбрать команду **Назначить макрос**. В появившемся диалоговом окне выбрать нужный макрос и нажать «ОК».

Чтобы воспользоваться возможностью назначить макрос *кнопке на панели быстрого доступа*, необходимы следующие действия:

1. Открыть окно **Параметры Excel** используя кнопку **Файл**. В параметрах MS Excel выбрать **Настройка** (Панель быстрого доступа), установить в меню **Выбрать команды из:** «Макросы» и **Настройка быстрого доступа:** «Для Книга1.xlsx», где Книга1.xlsx – название файла с макросом.

2. Затем выбрать требуемый макрос из списка, нажать на кнопку **Добавить**, затем – на кнопку «ОК» (рисунок 21). После выполненных действий на панели быстрого доступа появится дополнительная кнопка, при нажатии на которую будет выполнен макрос.

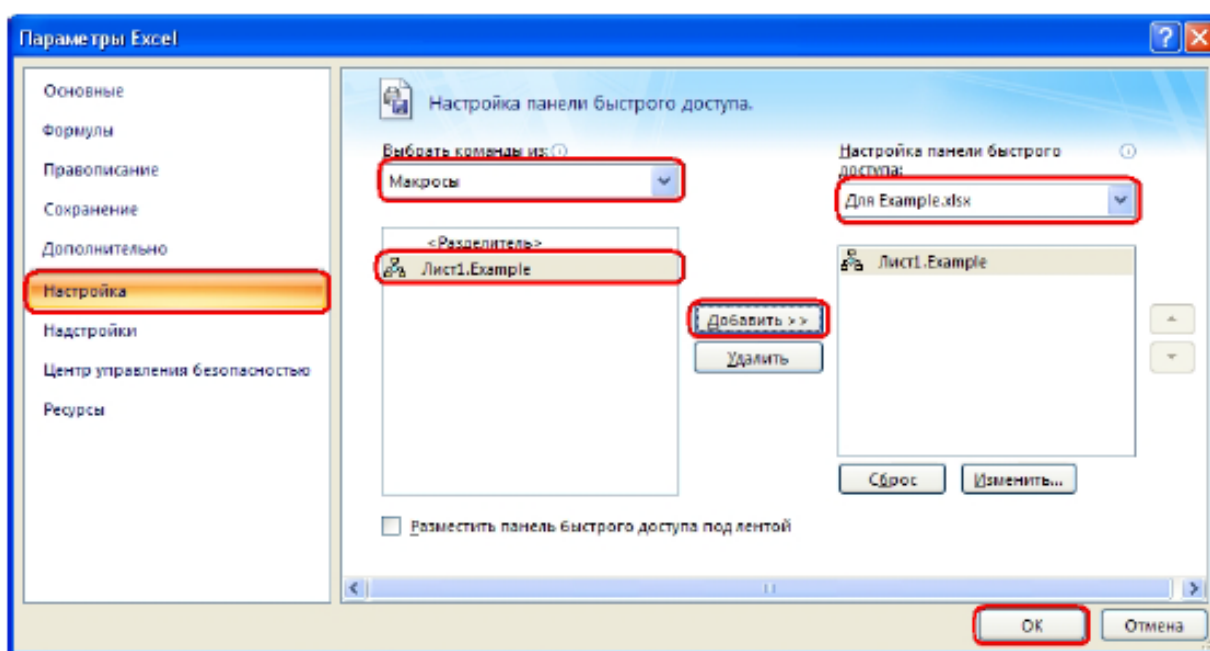


Рисунок 21 – Добавление кнопки на панель быстрого доступа

Пример 2. Разработать таблицу расчета стоимости перевозки стальных труб и создать два макроса. Первый макрос будет отвечать за расчет значений в графах «Стоимость», «Тариф перевозки», «НДС», «Всего» (ячейки C1, D1, E1); второй – за оформление шапки таблицы.

Методика выполнения задания:

1. Перейти на второй лист Excel и сформировать таблицу и заполнить исходные данные, как показано на рисунке 22.

	A	B	C	D	E	F	G
1	Расчет стоимости перевозки стальных труб						
2							
3	Оптовая цена, т/руб			500			
4	Тариф перевозки			5%			
5	Ставка НДС			28%			
6							
7	Вес трубы, т	Количество труб, шт	Стоимость труб, руб	Тариф перевозки, руб	НДС, руб	Всего, тыс. руб	
8	0,385	40					
9	0,125	50					
10	0,415	27					
11	0,443	10					
12							

Рисунок 22 – Исходные данные таблицы

2. Создать первый макрос, который будет вычислять значения в графах «Стоимость», «Тариф перевозки», «НДС», «Всего». Для этого, предварительно сделав активной ячейку C8, нажать кнопку «Записать макрос» на панели **Разработчик**, задать имя макроса «**Расчет_стоимости**», назначить сочетание клавиш **Ctrl+w**., заполнить описание макроса. Для выполнения расчета выполнить следующую последовательность действий:

- в ячейке C8 вычислить стоимость перевозки труб как произведение веса трубы, количества труб и оптовой цены;
- в ячейках D8 и E8 вычислить тариф перевозки и НДС как соответствующий процент от стоимости;
- в ячейке F8 посчитать сумму стоимости, тарифа перевозки и НДС в тыс. руб.

– выделить диапазон C8:F8 и используя маркер автозаполнения, скопировать формулы в четыре нижние ячейки;

Остановить запись макроса. Для запуска макроса создать на листе кнопку «Расчет стоимости», рассчитать значения остальных ячеек таблицы.

3. Создать второй макрос, который будет отвечать за оформление шапки таблицы. Для этого выделить диапазон ячеек A7:F7, нажать кнопку «**Записать макрос**», задать имя макроса «Формат_заголовка», назначить сочетание клавиш Ctrl+e, заполнить описание макроса и выполнить следующую последовательность действий:

– В контекстном меню выделенного диапазона выбрать команду **Формат ячейки**. В открывшемся диалоговом окне установить: на вкладке **Шрифт**: шрифт *Times New Roman*, размер – 12 пунктов, начертание – полужирный *курсив*; на вкладке **Выравнивание**: перенос по словам, по горизонтали и по вертикали «по центру»; на вкладке **Заливка**: цвет ячеек светло-серый на вкладке **Границы**: внешняя толстой линией, внутренняя обычной.

– Закрывать контекстное меню, открыть команду **Формат** на панели **Главная** вкладка ячейки. Задать ширину столбцов – 15, высоту строк – 45.

– Остановить запись макроса, для запуска макроса создать на панели быстрого доступа.

4. Применить к шапке таблицы макрос «Формат_заголовка», к остальным ячейкам таблицы макрос «Формат_таблицы». Окончательный вариант таблицы представлен на рисунке 23.

	A	B	C	D	E	F
1	Расчет стоимости перевозки стальных труб					
2						
3	Оптовая цена, т/руб			500		
4	Тариф перевозки			5%		
5	Ставка НДС			28%		
6						
7	<i>Вес трубы, т</i>	<i>Количество труб, шт</i>	<i>Стоимость труб, руб</i>	<i>Тариф перевозки, руб</i>	<i>НДС, руб</i>	<i>Всего, тыс. руб</i>
8	0,385	40	7700	385	2156	10,241
9	0,125	50	3125	156,25	875	4,15625
10	0,415	27	5602,5	280,125	1568,7	7,451325
11	0,443	10	2215	110,75	620,2	2,94595
12						
13					Расчет стоимости	
14						

Рисунок 23 – Окончательный вариант выполнения примера 2

Индивидуальное задание:

1. Разработать на 3 листе Excel таблицу в соответствии с вариантом, приведенных в таблице 2 (в скобках указаны наименования полей таблицы).
2. Заполнить поля, названия которых выделены курсивом, исходными данными (5–7 записей).
3. Создать макрос «Вычислить значения», отвечающий за расчет значений в графах, названия которых подчеркнуты.
4. При создании макроса назначить сочетание клавиш Ctrl+r, добавить описание с фамилией и датой выполнения задания. Для выполнения назначить кнопку на листе.
5. Применить к шапке таблицы макрос «Формат_заголовка», к остальным ячейкам таблицы макрос «Формат_таблицы».
6. Сохранить книгу под именем «Лабораторная работа №1», указать тип файла – «Книга Excel с поддержкой макросов».

Таблица 2 – Варианты индивидуального задания к лабораторной работе №1

№	Название и описание таблицы
1	2
1	Ведомость операций службы по прокату строительной техники (название и марка тех.оборудования, цена проката в час, количество часов проката, <u>налог на прибыль, выручка</u>).
2	Ведомость начисления заработной платы строительной организации (ФИО сотрудника, оклад, количество отработанных дней, <u>премия, начислено</u>).
3	Ведомость реализации строительной продукции предприятием (наименование товара, цена, количество, <u>НДС, стоимость с НДС</u>).
4	Ведомость использования топлива строительным предприятием (номер автомобиля, остаток на начало месяца в л, приход в л, <u>остаток на конец месяца в л</u>).
5	Отчет о продажах стройматериала за год (Наименование товара, продано во 2 полугодии, продано в 1 полугодии, <u>увеличение по сравнению с 1-м полугодием, процент роста</u>).
6	Ведомость начисления стипендии студентам строительного факультета (ФИО студента, начислено, социальная часть, <u>всего</u>).
7	Индивидуальная выписка сотруднику по оплате проведенных работ / услуг (наименование работы, кол-во часов*, тарифная ставка за час, <u>сумма к оплате</u>).
8	Бланк покупки материала для ремонта помещения (Наименование товара, цена, количество, <u>скидка стоимость</u>).
9	Ведомость операций автомобильной стоянки (номер автомобиля, время остановки, время освобождения, стоимость часа, <u>время стоянки в часах, к оплате</u>).
10	Ведомость операций оплаты за электроэнергию (ФИО плательщика, предыдущее показания, текущие показания, тариф за Кв/ч, <u>израсходовано, начислено</u>).
11	Ведомость оплаты отделочных и ремонтных работ (ФИО мастера, ФИО заказчика, наименование работы, тариф за час, количество отработанных часов, <u>скидка, стоимость</u>).

Продолжение таблицы 2

1	2
12	Ведомость операций по ремонту строительной техники (<i>Наименование организации владельца техники, наименование и марка технического оборудования, вид ремонтных работ, тариф, стоимость</i>).
13	Ведомость по прохождению практики студенческой группы (<i>ФИО студента, начало практики, конец практики, количество отработанных дней, оценка</i>).
14	Ведомость оплаты командировки сотрудников (<i>ФИО, город, транспортные расходы, питание и проживание, стоимость работы в сут., срок командировки в сут., тариф оплаты работ в сут. общие затраты, итого на руки</i>).
15	Ведомость риэлтерского агентства (<i>район, жилая площадь, количество комнат, доп. площадь, стоимость 1м², общая площадь, стоимость помещения</i>).

2.6 Вопросы для самопроверки

1. Охарактеризуйте стили программирования, которые поддерживает VBA.
2. Какие основные понятия объектно-ориентированного программирования вы можете назвать?
3. Что такое свойства, методы и события объекта, приведите примеры.
4. Что такое формы и модули, для чего они используются в VBA?
5. Что такое проект в VBA, какие части включает в себя проект?
6. Какие классы лексем используют в программировании?
7. Какие операторы используются в VBA? Назовите их виды и перечислите операторы каждого вида.
8. Что такое макросы, для чего они используются?
9. Какие достоинства и недостатки использования макрореордера при создании макроса?
10. Опишите структуру любого макроса. Какая процедура используется в начале и в конце записи любого макроса?
11. Как создать макрос, используя макрореордер?

12. Какие можно использовать способы запуска макроса? Как можно выполнить каждый из способов?

3 Основные понятия VBA. Программирование алгоритмов линейной структуры. Создание форм

3.1 Синтаксис процедур Sub и Function

Процедура – это наименьшая единица программного кода, на которую можно ссылаться по имени и которая может выполняться независимо.

В VBA основные типы процедур – это **Sub** и **Function**.

Любая процедура содержит один или более операторов, помещенных между двумя специальными операторами: объявлением процедуры в начале и оператором завершения процедуры в конце.

Процедуры **Function** предназначены для создания блока программного кода, который выполняет действие в пределах своего блока и возвращает значения

Синтаксис:

[Private /Public] Function Имя (список параметров) **As** Тип

Тело функции (строки кода)

Имя=Возвращаемое значение

End Function

Где:

– **[Private /Public]** - необязательные ключевые слова, определяющие область видимости процедуры;

– **Function** – ключевое слово, указывающее на начало процедуры, далее следует ее имя;

– **As** – ключевое слово, предваряющее определение типа;

– **End Function** – ключевые слова, указывающие на окончание процедуры

Например, составить процедуру нахождения периметра прямоугольника:

Function Perimetr (x, y)

$$Perimetr = 2*(x+y)$$

End function

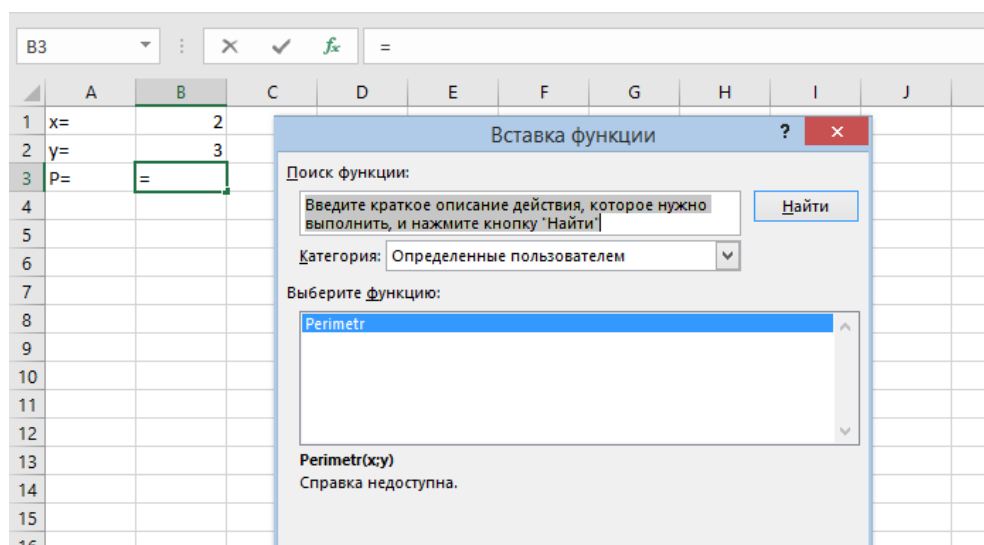


Рисунок 24 – Вызов процедуры-функции Perimetr

Особенности процедуры Function:

1. Функция вычисляет единственное результирующее значение простого типа.
2. Носителем возвращаемого значения является ее имя. Поэтому в теле функции должен присутствовать хотя бы один оператор присваивания (=), который определяет значение, вычисляемое с помощью функции

Процедура **Sub** (или подпрограмма) — это блок кода, который выполняется в ответ на событие.

Процедура может иметь параметры, значение которых изменяется при выполнении кода. В отличие от Function значения не возвращаются.

Синтаксис процедуры Sub:

[Private | Public | Static] Sub Имя(список параметров)

[Private| Public] |Dim | Const (перечисление имен) ***As*** Тип

Тело процедуры (строки кода)

End Sub

Например, составить процедуру нахождения суммы двух чисел:

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

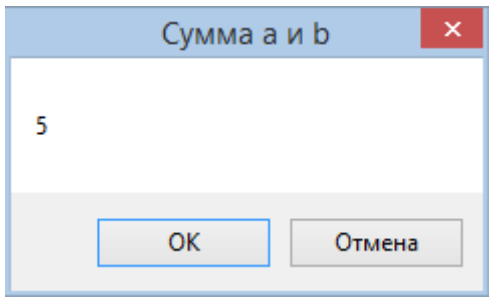


Рисунок 25 – Результат процедуры сложения целых чисел

3.2 Данные, константы, переменные, идентификаторы

Имена (идентификаторы) – употребляются для обозначения объектов программы (переменных, констант, процедур и др.).

В VBA имена констант, переменных и процедур должны удовлетворять следующим требованиям:

- начинаются с буквы;
- не содержат точки и символов объявления типа;
- не длиннее 255 символов (имена объектов не более 40 символов);
- не являются служебными словами.

Данные - величины, обрабатываемые программой и имеющие значения.

Имеется три основных вида данных: константы, переменные, массивы.

Константы - это данные, которые зафиксированы в тексте программы и не изменяются в процессе ее выполнения.

Переменная - это поименованная область компьютерной памяти, в которой хранятся данные определённого типа.

Эти данные могут меняться в ходе выполнения программы. В программе переменная может встречаться многократно – в формулах и операторах.

Любая переменная или константа имеет три обязательные характеристики: **имя**, **тип**, **значение**.

Переменные и константы в начале программы, до того момента, когда они используются, желательно объявлять (декларировать) - указывать имя и тип.

Для декларации констант используют ключевое слово **Const**

Синтаксис:

Const <имя константы> As <тип> = <значение>

или

Const <имя константы> = <значение>

Если в константе (переменной) явно не указан тип данных, то VBA назначает ей тип, который соответствует значению выражения.

Примеры:

Const pi = 3,1459

Const y = 34

Const c As Byte = 34

Const z As Single = -3,8374E-22

Строковые константы указываются в кавычках:

Const prv As String = “Язык программирования VBA”

Для декларации переменных используются ключевые слова: **Dim**, **Private** и **Public**. Первые два слова – для объявления локальных переменных, третье - для объявления глобальных переменных.

Ключевое слово **Dim** используется при объявлении переменной внутри процедуры. При таком объявлении область видимости этой переменной – только сама эта процедура. Из других процедур данная переменная не видна. Поэтому в других процедурах имя этой переменной может использоваться для любых иных целей.

Ключевое слово **Private** служит для объявления тоже локальной переменной, но используется не внутри процедуры, а снаружи. При этом объявляемая локальная переменная видна во всех процедурах данного модуля, но не видна из других модулей данного программного проекта.

Ключевое слово **Public** служит для объявления глобальной переменной, область видимости которой простирается и на другие модули проекта. Для этого необходимо в других модулях обеспечить возможность ссылаться на эту переменную, указывая перед её именем имя модуля, в котором эта переменная объявляется.

Синтаксис:

Dim/Private/Public <имя переменной> **As** <тип> = <значение>

или

Dim/Private/Public <имя переменной> = <значение>

Примеры:

Dim Z As Single

Public i As Integer, r As Long, c As Date

Private Строка As String

3.3 Типы данных

Тип переменной (и константы) – это множество значений, которые она принимает.

В языке VBA допускается случай, когда тип переменной не объявлен заранее, до присвоения этой переменной значения определённого типа. В этом случае при первом же действии с переменной, когда она получает некоторое значение, ей автоматически приписывается тип этого значения. Однако, предпочтительным считается объявление типа переменной заранее, до её использования в каких-либо действиях.

Основные типы данных, используемые в VBA приведены в таблице 3.

Таблица 3 – Типы данных VBA

Тип данных	Размер в памяти	Диапазон значений
Byte (байт)	1 байт	от 0 до 255.
Boolean (логический)	2 байта	2 значения (True,False)
Integer (короткое целое)	2 байта	От -32 768 до 32 767
Long (длинное целое)	4 байта	От -2 147 483 648 до 2 147 483 647
Single (десятичное короткое)	4 байта	от -3,402823E+38 до -1,401298E-45 от 1,401298E-45 до 3,402823E38 (7 знач. цифр)
Double (десятичное длинное)	8 байт	от -1.8E+308 до -4.9E-324 от 4.9E-324 до 1.8E+308 (15 знач. цифр)
Currency (денежное)	8 байт	от -922 337 203 685 477.5808 до 922 337 203 685 477.5807
Date (время-дата)	8 байт	От 1 января 100 г. до 31 декабря 9999 г.
String (строка переменной длины)	10 байт + длина строки	От 0 до 65 тысяч символов
String (строка переменной длины)	длина строки	От 0 до 65 тысяч символов

3.4 Ввод и вывод данных

Для ввода данных в VBA используется функция ***Inputbox***.

Синтаксис функции:

InputBox (сообщение, заголовок [значение по умолчанию], [координата x], [координата y]).

Аргументы:

- *сообщение* – строковое выражение, приглашение пользователю;
- *заголовок* – строковое выражение, определяющее заголовок окна;
- *значение по умолчанию* – строковое выражение в текстовом поле окна
- *координаты x и y* – положение окна на экране

Пример:

$x = \text{InputBox}(\text{"Введите натуральное число n"}, \text{"Ввод данных"}, \text{"a1"}, 100, 200)$

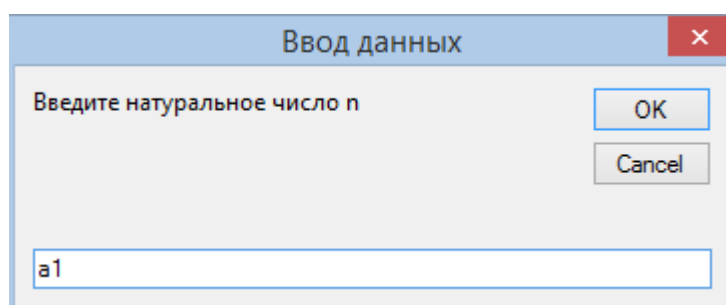


Рисунок 26 – Окно ввода данных, сгенерированное функцией InputBox

Вывод данных в VBA осуществляется при помощи функции *MsgBox*, ее синтаксис:

MsgBox "сообщение", [значение], "заголовок"

Аргументы:

- *сообщение* – строковое выражение, отображаемое в диалоговом окне;
- *значение* – числовое выражение, которое является суммой значений, указывающих число и тип отображаемых кнопок, тип используемого значка, основную кнопку (по умолчанию значение этого аргумента = 0);
- *заголовок* – строковое выражение, определяющее заголовок окна вывода.

Пример:

$a = 1$

$\text{MsgBox} \text{"Значение a="} \& a$

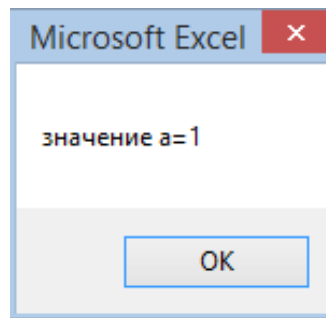


Рисунок 27 – Окно сообщений, сгенерированное функцией MsgBox

В рассмотренном примере определен только аргумент *Сообщение*, знак & используется для вывода значений в одну строку. Для того чтобы определить аргумент *Значение* необходимо воспользоваться данными из таблиц 4 и 5.

Таблица 4– Значения аргумента, определяющие отображаемые кнопки

Константа	Значение	Отображаемые кнопки
VbOKOnly	0	ОК
VbOKCancel	1	ОК, Отмена
VbAbortRetryIgnore	2	Стоп, Повтор, Пропустить
VbYesNoCancel	3	Да, Нет
VbYesNo	4	Да, Нет, Отмена
VbRetryCancel	5	Повтор, Отмена

Таблица 5 – Значения аргумента, определяющие отображаемые значки

Константа	Значение	Отображаемые кнопки
VbCritical	16	
VbQuestion	32	
VbExclamation	48	
VbInformation	64	

Рассмотрим примеры использования различных значений аргументов функции MsgBox:

- а) *MsgBox "a=1", 1, "пример1"* б) *MsgBox "a=1", 51, "пример2"*

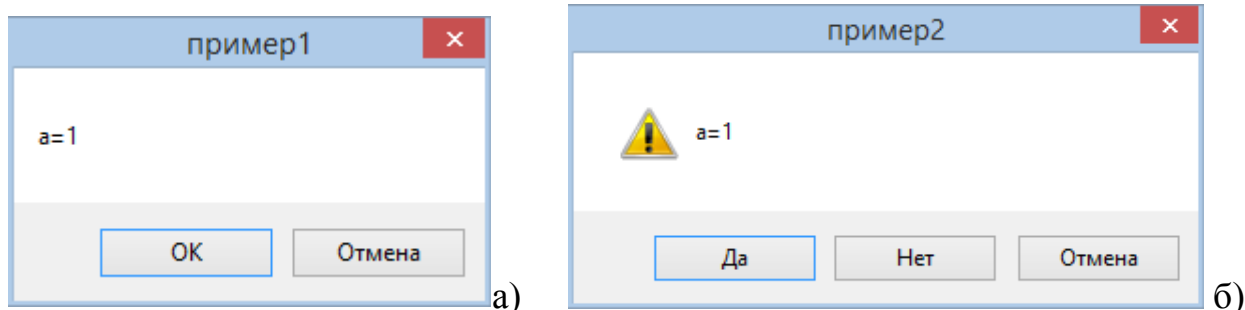


Рисунок 28 – Сообщения, сгенерированные функцией MsgBox для примеров а и б

3.5 Операторы и выражения языка VBA

Оператор – это законченная фраза языка программирования, определяющая законченный этап обработки данных.

Оператор является наиболее крупным и содержательным понятием языка, в его состав могут входить: ключевые слова, данные и выражения.

Операторы подразделяются на исполняемые и неисполняемые. **Исполняемые** операторы используются для выполнения некоторых действий (присваивание, ввод и вывод информации, определения условия, цикла и др.). **Неисполняемые** операторы предназначены для описания данных и структуры программы

Оператор присваивания - предназначен для того, чтобы присвоить переменной нужное значение.

Оператор присваивания использует операцию присваивания, которая всегда включает знак равенства «=». Данная операция предписывает выполнить выражение, заданное в правой части, и присвоить результат имени переменной, указанной в левой части.

Например: $a = 3.14;$ $a = a + 1$

Выражения - предназначены для выполнения заданных вычислений, записываются в виде линейных последовательностей символов слева направо, соблюдая приоритет операций. Различают выражения: арифметические, логические, строковые.

Арифметические выражения служат для определения одного числового значения могут содержать числовые константы, переменные, арифметические операторы и математические функции.

Логические выражения описывают некоторые условия, которые могут выполняться или не выполняться и принимают только два значения – «истина» или «ложь», содержат операторы сравнения, числовые константы, переменные и логические функции.

Значением **строковых выражений** является текст. Данные выражения могут содержать литерные константы, литерные переменные и литерные функции, объединенные знаком конкатенации (&).

Различные виды операторов, используемых при составлении соответствующих выражений приведены в таблице 6.

Таблица 6 – Операторы VBA

Оператор	Действие	Пример
1	2	3
Арифметические операторы		
+ (знак плюс)	Сложение	3+3
– (знак минус)	Вычитание	3–1
	Унарный минус	–1
* (звездочка)	Умножение	3*3
/ (косая черта)	Деление	5/8 (результат 0.625)

Продолжение таблицы 6

1	2	3
\ (обратная черта)	Целочисленное деление	5\8 (результат 0)
mod	Остаток от деления	12 mod 5 (результат 2)
% (знак процента)	Процент	20%
^ (крышка)	Возведение в степень	3^2 (аналогично 3*3)
<i>Текстовый оператор</i>		
& (амперсанд)	Объединение последовательностей символов в одну последовательность (конкатенация)	"Язык " & " программирования" = "Язык программирования"
<i>Операторы сравнения</i>		
= (знак равенства)	Равно	A1=B1
> (знак больше)	Больше	A1>B1
< (знак меньше)	Меньше	A1<B1
>= (знак больше и знак равенства)	Больше или равно	A1>=B1
<= (знак меньше и знак равенства)	Меньше или равно	A1<=B1
<> (знак меньше и знак больше)	Не равно	A1<>B1

При составлении логических выражений используются следующие логические функции:

- ***and*** – конъюнкция (логическое И);
- ***or*** – дизъюнкция (логическое ИЛИ);

- **not** – логическое отрицание;
- **xor** - логическое исключение.

Математические функции, используемые в арифметических выражениях приведены в таблице 7.

Таблица 7 – Математические функции VBA

Функция	Математическая запись, возвращаемое значение
1	2
Abs(x)	x - Модуль числа
Asin(x)	Arcsin(x) – угол, синус которого равен указанному числу
Acos(x)	Arccos(x) – угол, косинус которого равен указанному числу
Atan(x)	arctg(x) – угол, тангенс которого равен указанному числу.
Cos(x)	cos(x) – косинус указанного в радианах угла
Exp(x)	e^x – число e , возведенное в указанную степень, где e – основание натурального логарифма
Hex(x)	Возвращает значение аргумента в шестнадцатеричной системе счисления
Log(x)	ln(x) – натуральный логарифм от числового выражения
Log10(x)	lg(x) – логарифм с основанием 10 от числового выражения
Max(x,y)	Наибольшее из двух чисел
Min(x,y)	Наименьше из двух чисел
Oct(x)	Возвращает значение аргумента в восьмеричной системе счисления
Round(x, a)	Число x , округленное до числа a десятичных знаков

Продолжение таблицы 7

1	2
Rnd (x)	Случайное число из интервала [0,1). Перед вызовом функции надо использовать оператор Randomize – запуск генератора случайных чисел
Sin (x)	sin(x) – синус угла заданного в радианах
Sqrt (x)	\sqrt{x} – квадратный корень числового выражения
Sgn (x)	Возвращает +1, если значение параметра положительно, -1, если отрицательное, 0, если 0
Tan (x)	tg (x) – тангенс угла от значения в радианах

3.6 Лабораторная работа №2 «Программирование алгоритмов линейной структуры при помощи пользовательских функций и форм»

Пример 1. Создайте пользовательскую функцию, вычисляющую по формуле Пифагора длину гипотенузы прямоугольного треугольника.

Создание новой функции MS Excel, которую впоследствии можно использовать аналогично встроенным функциям является одной из возможностей VBA. Это целесообразно в тех случаях, если необходимой функции нет в стандартном наборе встроенных функций MS Excel, например, формулы Пифагора, а ей приходится часто пользоваться.

Методика выполнения:

1. Открыть MS Excel и перейти в редактор VBA, используя кнопку **Visual Basic** на панели **Разработчик** или сочетание клавиш **Alt+F11**.

2. Командой **Module** меню **Insert** добавить новый модуль, в нем разместить функцию через команду **Procedure** меню **Insert**, выбрать тип **Function**, задать имя новой функции, нажать кнопку «ОК» (рисунок 29).

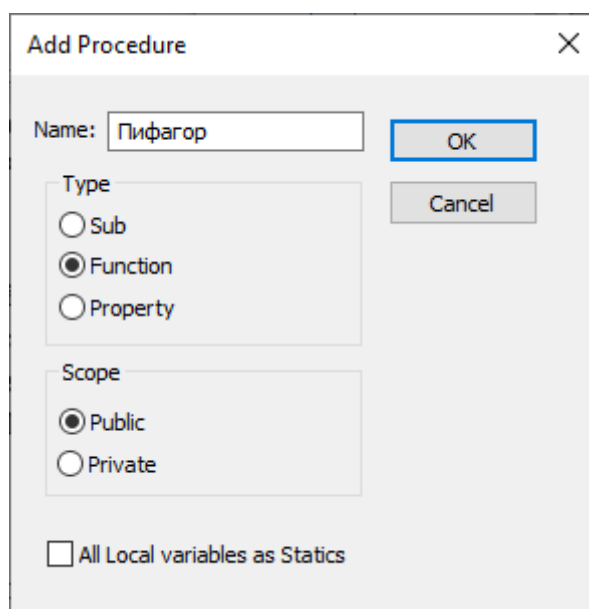


Рисунок 29 – Добавление новой функции

3. В окне кода между двумя появившимися строчками написать программный код для данной функции, учитывая, что для нахождения длины гипотенузы по формуле Пифагора нужно знать значения двух катетов (Рисунок 30).

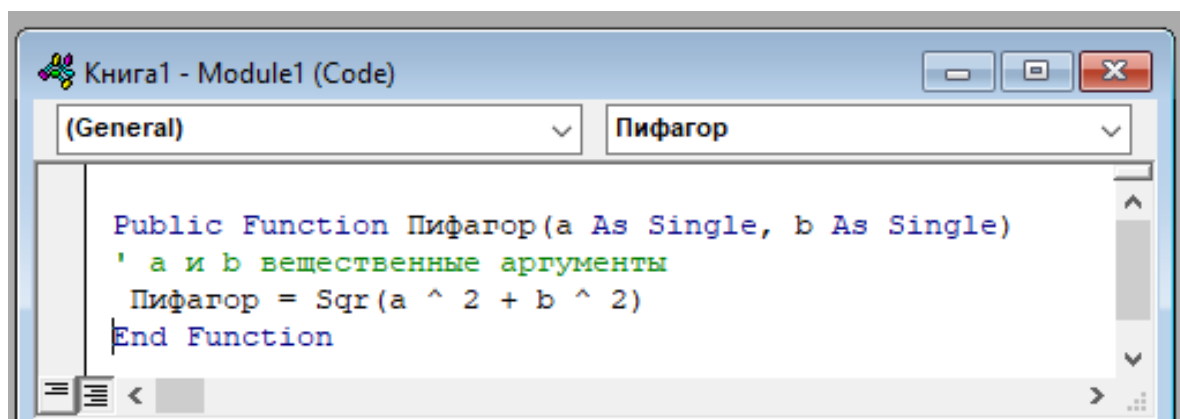


Рисунок 30 – Программный код функции

4. Закрыть редактор VBA. Ввести в ячейки A2 и B2 – значения катетов, а в ячейку C2 вставить функцию Excel, созданную функцию выбрать в категории «Определенные пользователем» диалогового окна **Вставка функций**, ввести значения аргументов (рисунок 31).

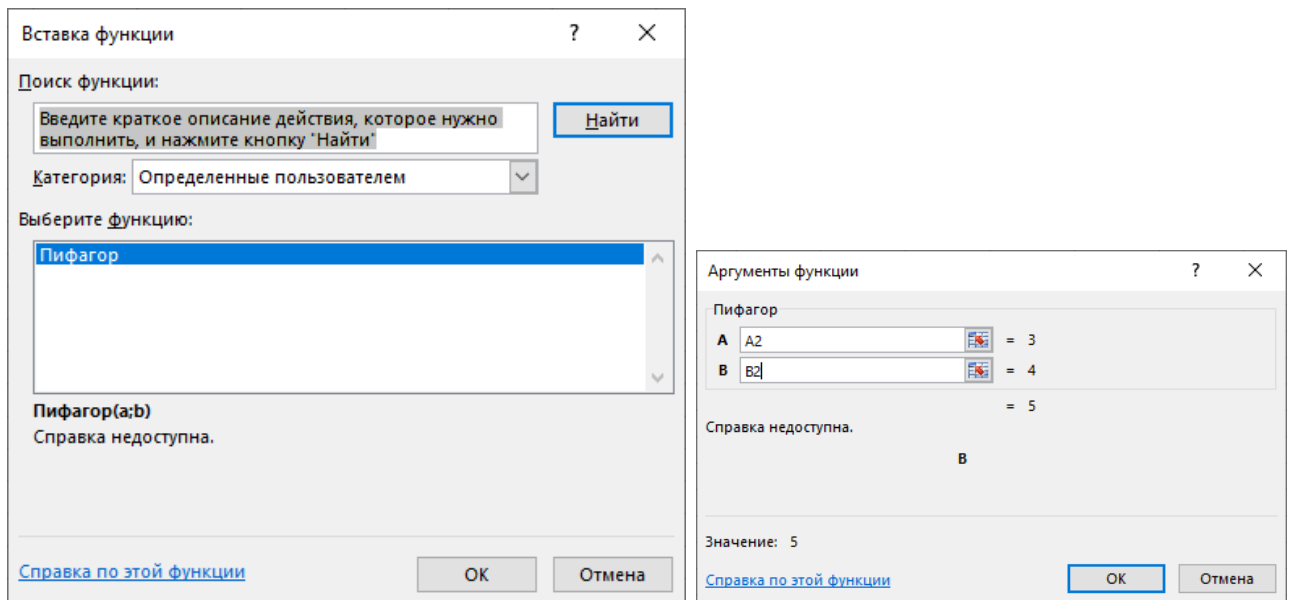


Рисунок 31 – Добавление пользовательской функции

5. Добавить к вновь созданной функции описание, поясняющее ее назначение и автора. Для этого открыть окно создания макросов, набрать в поле *Имя макроса* диалогового окна *Макрос* название функции и ввести описание, нажав кнопку *Параметры* (рисунок 32).

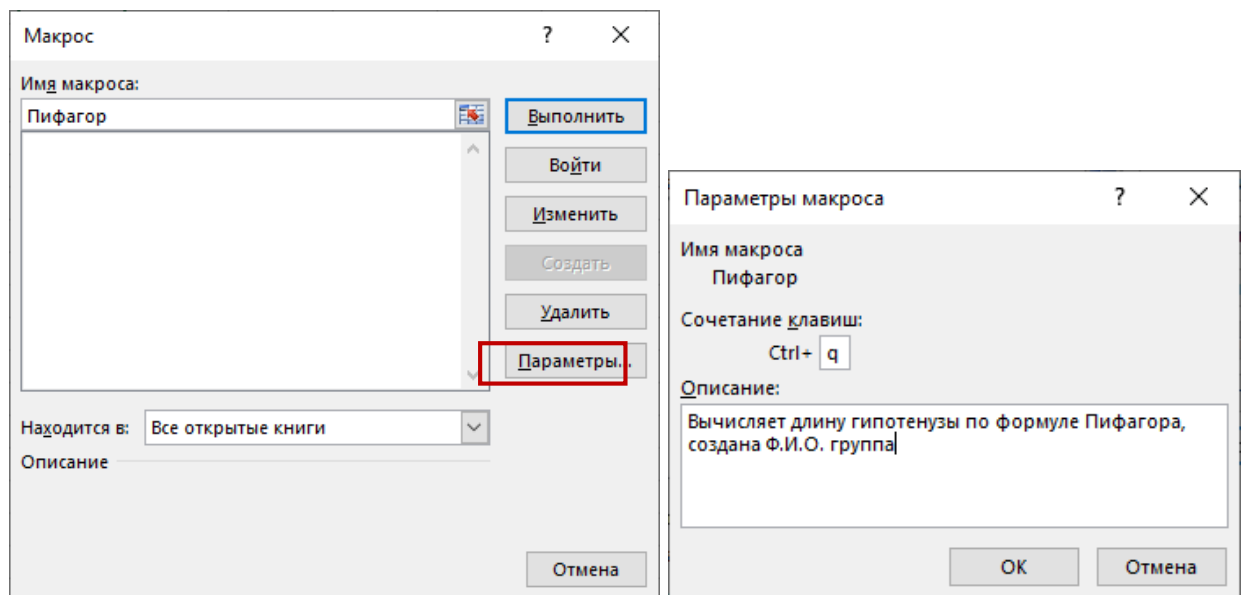
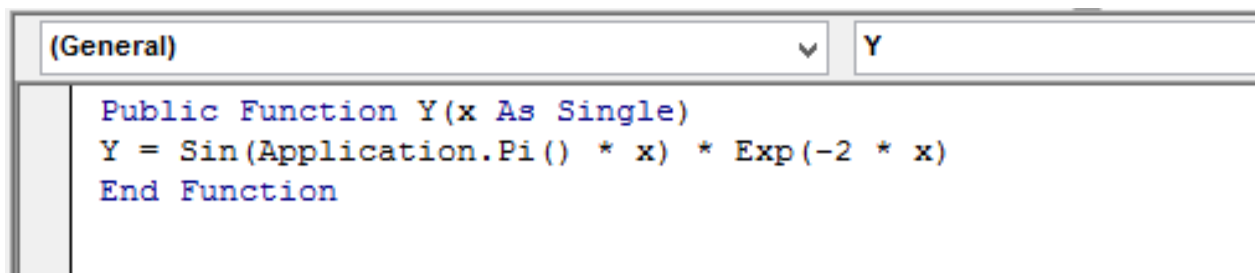


Рисунок 32 – Добавление описания функции

Пример 2. Создать функцию пользователя, математически определенную как $y = \sin(\pi x)e^{-2x}$

Методика выполнения :

1. В редакторе VBA MS Excel в новом модуле через команду **Procedure** меню **Insert** создать функцию с именем «Y» и написать для нее программный код, показанный на рисунке 33.



The image shows a screenshot of the VBA editor's code window. At the top, there is a tab labeled '(General)' and a dropdown menu showing the function name 'Y'. Below this, the code is written in blue text:

```
Public Function Y(x As Single)
Y = Sin(Application.Pi() * x) * Exp(-2 * x)
End Function
```

Рисунок 33 – Программный код функции Y

Данный программный код использует стандартную функцию **Pi()**, которая возвращает значение постоянной π (в Excel – функция пи()). Так как она не является внутренней функцией VBA, то ее необходимо записать в виде **Application.Pi()**.

2. Проверить работу функции для некоторого диапазона переменной x. Задать описание функции.

VBA позволяет создавать и использовать экранные формы, разработанные пользователем. Для создания новой формы пользователя необходимо в MS Excel перейти в редактор VBA и выбрать команду **UserForm** в меню **Insert**. В результате откроется окно конструктора форм (рисунок 34).

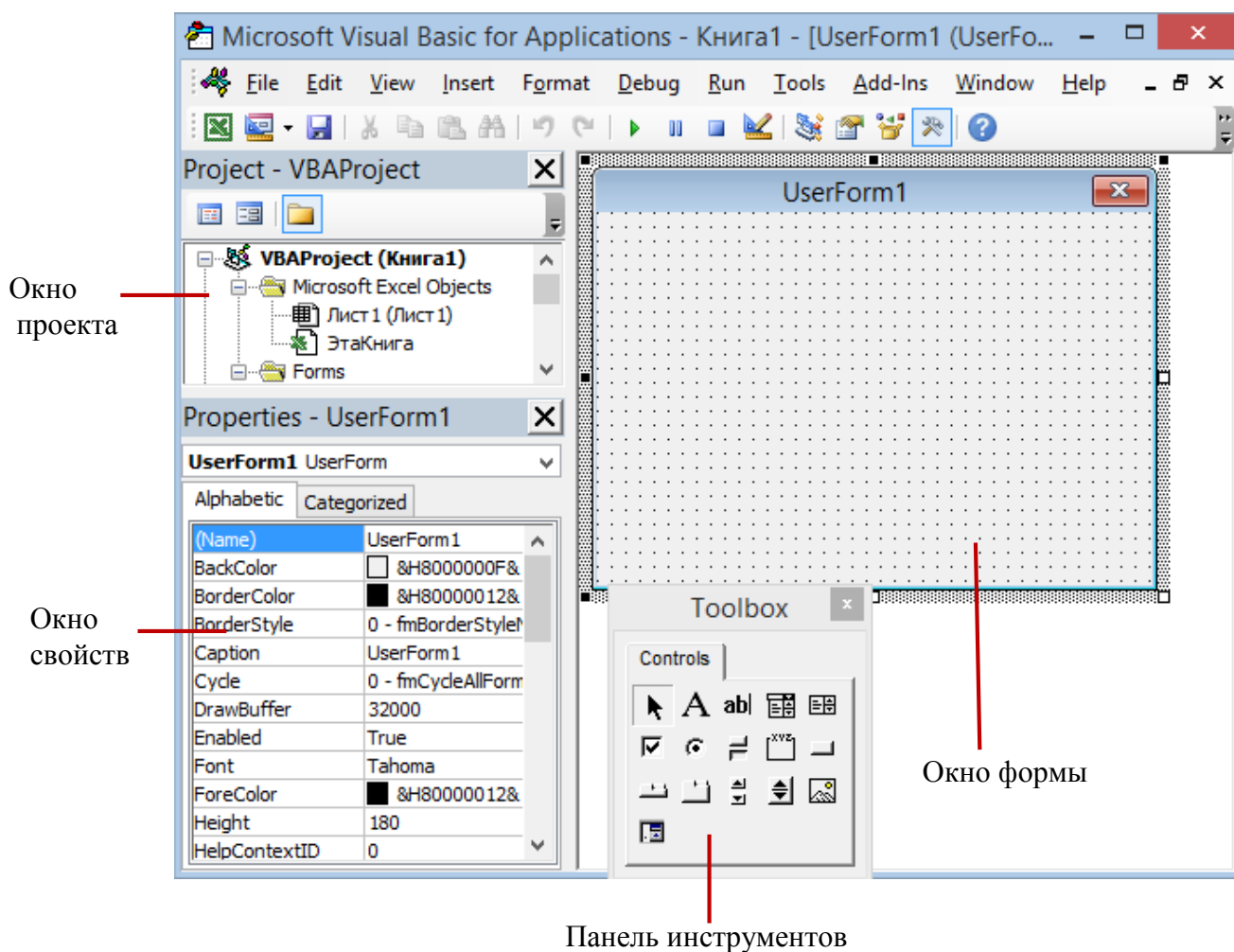


Рисунок 34 – Окно конструктора форм

При создании формы автоматически отображается панель элементов управления *Toolbox (Панель элементов)*, содержащая кнопки, с помощью которых соответствующие элементы управления можно разместить в создаваемой форме. После размещения элемента управления на форме с помощью окна свойств обычно задаются свойства выделенного объекта. Некоторые из свойств для часто используемых элементов управления *Кнопка*, *Поле* и *Надпись* приведены ниже (таблица 8).

Для того чтобы связать выделенный на форме объект с кодом, необходимо дважды щелкнуть по нему мышкой. В открывшемся окне редактирования кода можно выбрать событие, для которого требуется создать процедуру обработки, и ввести текст данной процедуры.

Таблица 8 – Свойства элементов управления *Кнопка, Поле, Надпись*

<i>Название</i>	<i>Назначение</i>
Элемент управления Label (Надпись)	
Name	Определяет имя надписи
Caption	Задаёт текст, отображаемый в надписи
AutoSize	Определяет размер надписи. Допустимые значения: <i>True</i> – режим автоматического подбора размера надписи под размер текстовой информации; <i>False</i> – устанавливает фиксированный размер надписи
TextAlign	Задаёт расположение текста в надписи. Допустимые значения: 1 – <code>fmTextAlignLeft</code> (по левому краю), 2 – <code>fmTextAlignCenter</code> (по центру), 3 – <code>fmTextAlignRight</code> (по правому краю)
Элемент управления TextBox (Поле)	
Name	Задаёт имя поля
Text	Задаёт текст в поле
Locked	Допустимые значения: <i>True</i> (запрещает ввод информации в поле); <i>False</i> (разрешает ввод информации в поле)
Элемент управления CommandButton (Кнопка)	
Name	Задаёт имя кнопки
Caption	Задаёт текст, отображаемый на кнопке
BackColor	Задаёт цвет фона кнопки

Пример 3. Используя VBA в MS Excel, разработать пользовательскую форму «Вычисление функции» для подсчёта значений функций $a = \frac{1}{3}\sqrt{x} + \frac{1}{5}\sqrt[5]{y}$ и $b = e^{(2x+1)}$ для действительных значений x и y .

Методика выполнения:

1. Открыть окно редактора VBA, добавить форму используя команду *UserForm* меню *Insert*, разместить на форме с помощью панели элементов *Toolbox* две надписи (элемент *Label*) и три командных кнопки (элемент *CommandButton*).

2. С помощью окна свойств (*Properties*) установить свойства данных объектов (таблица 9). Если данное окно отсутствует, его можно открыть через контекстное меню любого созданного элемента.

Таблица 9 – Описание свойств элементов формы Примера 3

Объект	Свойство
UserForm1	Caption = Вычисление функции
Label1	Caption = «a =» AutoSize = False
Label2	Caption = «b =» AutoSize = False
CommandButton1	Name = calc Caption = Вычислить
CommandButton2	Name = clean Caption = Очистить
CommandButton3	Name = exitForm Caption =Выход

Вид пользовательской формы в окне редактора VBA приведен на рисунке 35.

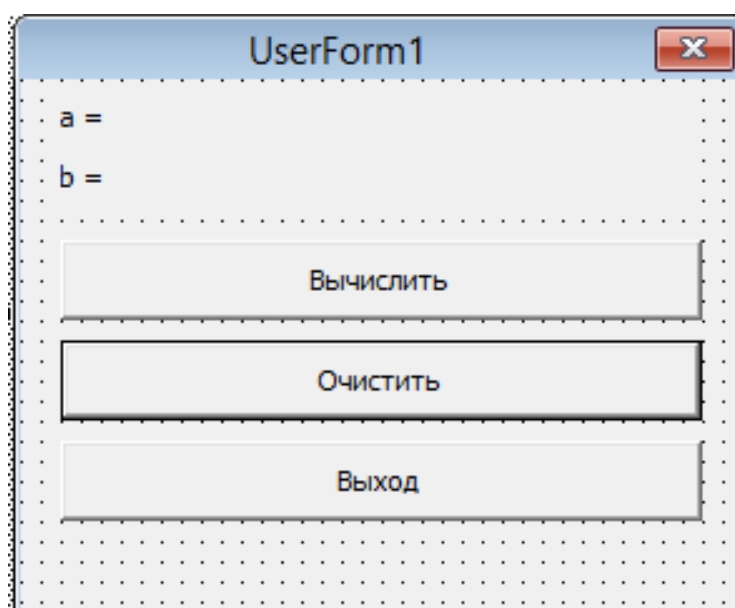
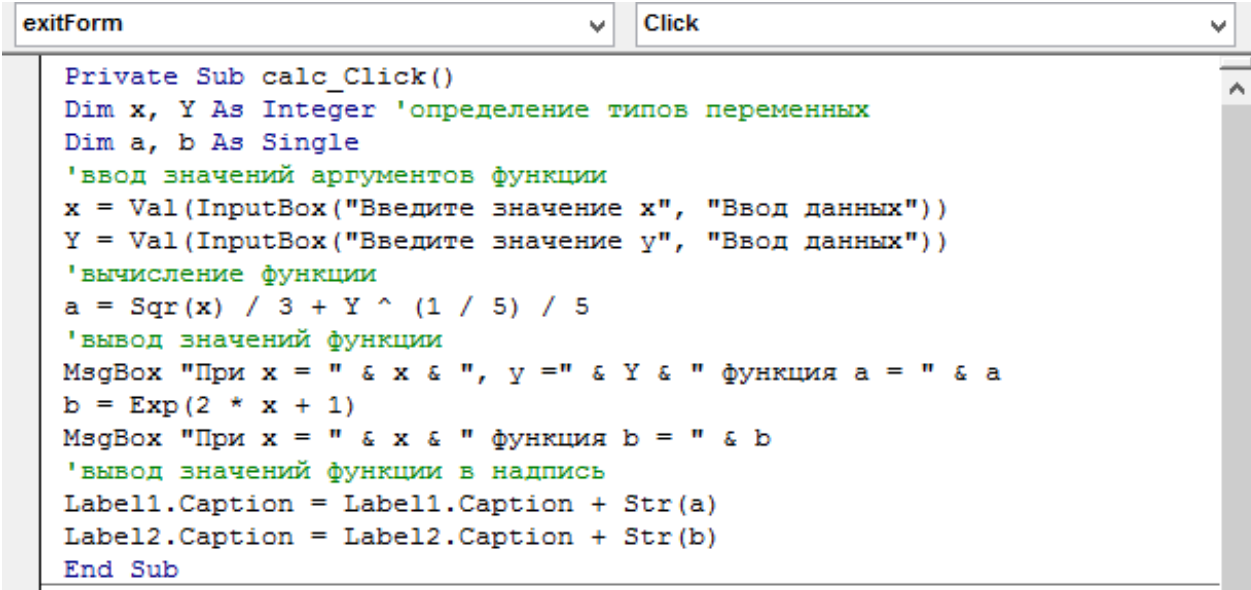


Рисунок 35 – Форма вычисления значения функции


3. Далее нужно перейти к обработке событий, возникающих при нажатии на кнопки. Для этого выполнить двойной щелчок мышью по командной кнопке «Вычислить». Откроется окно редактора кода VBA, в который нужно ввести программный код, как показано на рисунке 36.



```
exitForm Click
Private Sub calc_Click()
Dim x, Y As Integer 'определение типов переменных
Dim a, b As Single
'ввод значений аргументов функции
x = Val(InputBox("Введите значение x", "Ввод данных"))
Y = Val(InputBox("Введите значение y", "Ввод данных"))
'вычисление функции
a = Sqr(x) / 3 + Y ^ (1 / 5) / 5
'вывод значений функции
MsgBox "При x = " & x & ", y = " & Y & " функция a = " & a
b = Exp(2 * x + 1)
MsgBox "При x = " & x & " функция b = " & b
'вывод значений функции в надпись
Label1.Caption = Label1.Caption + Str(a)
Label2.Caption = Label2.Caption + Str(b)
End Sub
```

Рисунок 36 – Программный код для вычисления выражения примера 3

4. Закрыть окно программного кода. Аналогично ввести программный код для оставшихся кнопок «Очистить» и «Выход» (рисунок 37).



```
Private Sub clean_Click()
Label1.Caption = "a = "
Label2.Caption = "b = "
End Sub

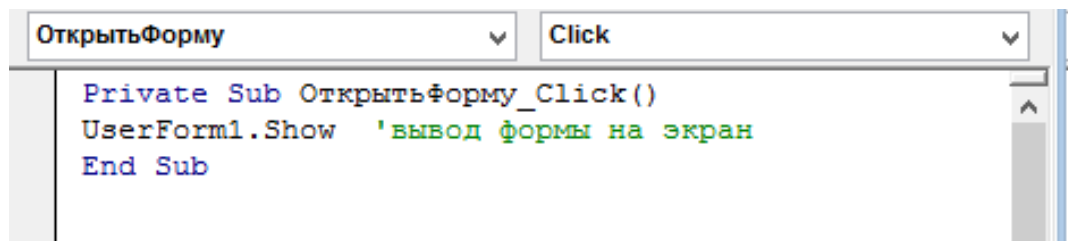
Private Sub exitForm_Click()
End
End Sub
```

Рисунок 37 – Программный код кнопок *Очистить* и *Выход*

5. Далее нужно проверить работу созданной формы. Для удобства запуска формы можно создать на листе Excel кнопку запуска формы. Для этого на панели

Разработчик во вкладке **Элементы управления** активировать **Режим конструктора** и в открывающейся вкладке кнопки **Вставить** выбрать инструмент **Кнопка** в группе **Элементы ActiveX**.

Выделить данную кнопку и, выбрав в контекстном меню пункт **Свойства**, установить в окне свойств значение свойства **Caption** = «Форма для расчета функции», а значение свойства **Name** = ОткрытьФорму. Далее, дважды щелкнув по кнопке, перейти в редактор VBA и ввести программный код, как показано на рисунке 38.



```
ОткрытьФорму Click
Private Sub ОткрытьФорму_Click()
UserForm1.Show 'вывод формы на экран
End Sub
```

Рисунок 38 – Программный код кнопки запуска формы

1. Перейти на лист и используя кнопку проверить работу созданной формы (рисунок 39).

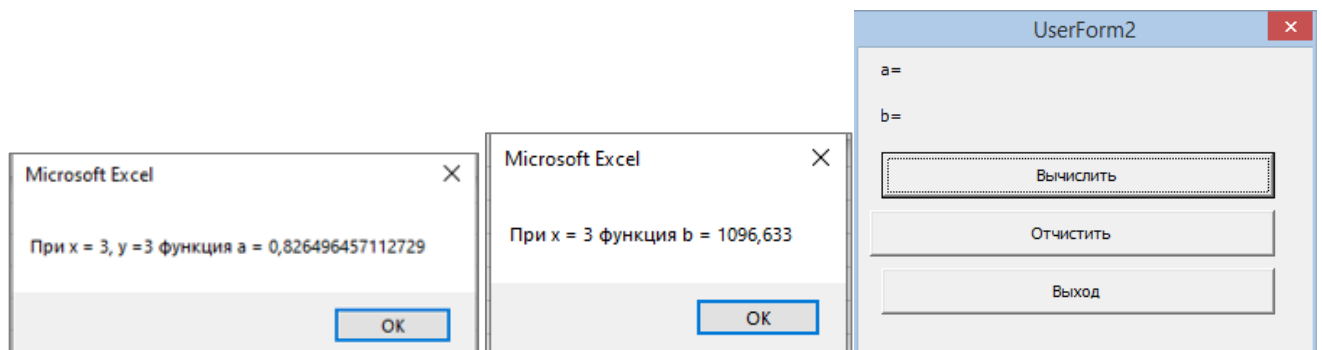


Рисунок 39 – Результат запуска формы

Индивидуальное задание №1:

Создать пользовательскую функцию в соответствии со своим вариантом (таблица 10). При создании функции добавить описание – назначение функции, Ф.И.О., группа.

Индивидуальное задание №2:

Разработать форму вычисления значения функции в соответствии со своим вариантом (таблица 10). Предусмотреть запуск формы через кнопку.

Таблица 10 – Варианты индивидуальных заданий лабораторной работы 2

<i>№ вар.</i>	<i>Описание функции</i>
1	2
1	Вычисление площади кольца S_k , если известны радиусы внешней R_1 и внутренней R_2 окружности: $S_k = \pi(R_1^2 - R_2^2)$
2	Вычисление объема конуса V , если известны его радиус R и высота h : $V = \frac{1}{3}\pi R^2 h$
3	Вычисление объема полого цилиндра V , если известны его наружный радиус R_1 и внутренний радиус R_2 основания и высота h : $V = \pi h(R_1^2 - R_2^2)$
4	Вычисление площади трапеции, если известны основания a и c и боковые стороны b и d , а также полупериметр трапеции p : $S = \frac{a+c}{a-c} \sqrt{(p-a)(p-c)(p-c-b)(p-c-d)}$
5	Вычисление объема полого шара V , если известны наружный радиус R_1 и внутренний радиус R_2 : $V = \frac{4}{3}\pi(R_1^3 - R_2^3)$
6	Вычисление объема усеченного прямого конуса V , если известны радиусы основания R и r , а также высота h : $V = \frac{1}{3}\pi h(R^2 + rR + r^2)$
7	Вычисление длины стороны треугольника a , если известны длины других сторон b и c , а также угол между ними α : $a = \sqrt{(b^2 + c^2 - 2bc \cos \alpha)}$
8	Вычисление объема эллипсоида V , если известны три его полуоси a , b и c : $V = \frac{4}{3}\pi abc$
9	Вычисление объема шарового слоя V , если известны радиусы R_1 и R_2 основания и высота h шарового слоя: $V = \frac{1}{6}\pi h^3 + \frac{1}{2}\pi(R_1^2 - R_2^2)h$

Продолжение таблицы 10

1	2
10	Вычисление площади равностороннего треугольника S , если известна его сторона a : $S = \frac{a^2\sqrt{3}}{4}$
11	Вычисление объема тора V , если известны радиусы внешнего R_1 и внутреннего R_2 круга: $V = 2\pi^2 R_1 R_2^2$
12	Вычисление объема усеченной пирамиды V , если известны площади ее треугольных оснований S_1 и S_2 и высота h : $V = \frac{1}{3}h(S_1 + S_2 + \sqrt{S_1 S_2})$
13	Вычисление шарового сегмента V , если известны радиус его основания R и высота сегмента h : $V = \frac{1}{6}\pi h(3R^2 + h^2)$
14	Вычисление объема пирамиды V , в основании которой лежит прямоугольник, если известны высота пирамиды h и длины сторон прямоугольника a и b : $V = \frac{1}{3}abh$
15	Вычисление площади равнобедренного треугольника S , если известны его боковая сторона a и основание c : $S = \frac{c\sqrt{4a^2 - c^2}}{4}$

3.7 Вопросы для самопроверки

1. Для чего предназначена процедура Function, каковы ее особенности и синтаксис?
2. В чем особенности процедуры Sub, ее синтаксис?
3. Что такое идентификаторы? Для обозначения чего они используются?
4. Что такое данные? Какие виды данных вы знаете?
5. В чем особенности констант, какие служебные слова используются для их объявления?
6. Для обозначения чего используются переменные? Чем отличаются локальные и глобальные переменные?

7. Какие служебные слова используются для объявления локальных и глобальных переменных?
8. Что такое тип данных?
9. Какие типы данных используются для описания числовых данных? Текстовых данных?
10. Какая процедура используется для ввода данных? Ее синтаксис?
11. Какая процедура используется для вывода данных? Ее синтаксис?
12. Что такое оператор? Что может входить в его состав?
13. В чем различие исполняемых и неисполняемых операторов?
14. Для чего используются арифметические выражения, какие операторы и функции они могут содержать?
15. В каких случаях используются логические выражения? Какие операторы и функции они могут содержать?
16. Чем являются строковые выражения, из чего они состоят?
17. Как создать и запустить пользовательскую функцию, в каких случаях они применяются?
18. В чем удобство экранных форм? Как можно создать и запустить форму?
19. Какие элементы управления могут использоваться в формах?

4 Программирование разветвляющихся вычислительных процессов

4.1 Инструкция If...Then

Разветвляющиеся алгоритмы проверяют истинность заданного условия. В зависимости от результата управление передается тому, или иному оператору. Как правило условие является выражением, в котором сравниваются значения некоторых переменных. Условие может быть *простым* – с одним оператором сравнения, или *сложным* – содержать несколько операций сравнения, объединенных логическими функциями (and, or, not).

В VBA используется несколько способов реализации ветвления, выбор способа определяется решаемой задачей. Большинство условных операторов начинается с ключевого слова *If*, далее идут другие ключевые слова, которые являются переключателями и позволяют осуществить переход к одному из операторов.

Наиболее простой условной инструкцией является *безальтернативное* ветвление. Данная инструкция содержит ключевые слова *If...Then*, проверяет только истинность выражения и имеет *однострочную форму* (записывается в одну строку).

Синтаксис оператора:

If Условие *Then* Оператор

Пример1. Ввести переменную x, если введенное меньше 0, то вывести соответствующее текстовое сообщение.

```
Sub Primer1( )  
x= InputBox ( "Введите значение переменной " )  
If x < 0 Then MsgBox "Число x отрицательное "  
End Sub
```

Однострочная форма условного оператора допускает выполнение нескольких операторов в результате проверки одного условия *If...Then*, при этом все операторы должны находиться на одной строке и разделяться двоеточием.

4.2 Инструкция If...Then...Else

Если в зависимости от условия нужно выполнить две или более различных команд, используется инструкция *альтернативного* условного оператора *If...Then...Else*.

Данный оператор можно использовать как в однострочной форме, так и в *многострочной (блоковой)* форме. Альтернативный блочный оператор *If* применяется в тех случаях, когда при выполнении условия нужно осуществить один набор операторов, при невыполнении – другой.

Синтаксис:

If *Условие* **Then** *Оператор1* **Else** *Оператор2*

Работа оператора: проверяется истинность условия, если оно истинно выполняется *Оператор1*, если ложно – *Оператор 2*.

Пример 2. Ввести число *x*, проверить положительное оно или отрицательное и вывести соответствующее сообщение.

Sub Primer2 ()

x = InputBox (" Введите значение переменной ")

If x < 0 Then MsgBox "Число отриц." Else MsgBox "Число положит."

End Sub

Наиболее часто для данного оператора используется блоковая форма записи, которая имеет следующий синтаксис:

If *Условие* **Then**

Блок операторов1

Else

Блок операторов2

End If

Работа оператора: проверяется *Условие*, если оно истинно – выполняется *Блок операторов1*, если ложно – *Блок операторов2*.

Пример 3. Ввести переменные *x* и *y*, если обе они положительные увеличить переменные на 1, иначе заменить на 1.

Sub Primer3 ()

x = InputBox (" Введите значение переменной x ")

y = InputBox (" Введите значение переменной y ")

If (x > 0) and (y > 0) Then

x = x + 1

y = y + 1

Else

x = 1

y = 1

```
End If
MsgBox ("x= " & x)
MsgBox ("y= " & y)
End Sub
```

В VBA также используется инструкция, предназначенная для работы с несколькими операторами If, в тех случаях, когда рассматривается несколько условий в дополнение к исходному *If...Then...ElseIf*. Данная инструкция позволяет проверить дополнительные условия, если исходное принимает значение ложь и имеет следующий синтаксис:

```
If Условие1 Then
Блок операторов 1
ElseIf Условие2 Then
Блок операторов2
ElseIf Условие3 Then
.....
Else
Блок операторов
End If
```

Работа оператора: проверяется *Условие1*, если оно истинно – выполняется *Блок операторов1*, если ложно – то последовательно проверяется каждое условие *Elseif*. Если условие истинно, выполняется блок операторов, следующий за соответствующим предложением *Then*. Если каждое из условий *Elseif* не является истинным выполняются инструкции, следующие за *Else*. После выполнения следующих за *Then* или *Else* операторов работа условного оператора заканчивается, и выполнение программы продолжается со следующего после *End If* оператора.

Пример 4. Ввести переменные x и y, если обе они положительные увеличить переменные на 1, если обе отрицательные уменьшить на 1, если они разного знака приравнять 0.

```
Sub Primer4 ( )
x = InputBox ( " Введите значение переменной x " )
```

y = InputBox (" Введите значение переменной y ")

If (x > 0) and (y > 0) Then

x = x + 1

y = y + 1

ElseIf (x < 0) and (y < 0) Then

x = x - 1

y = y - 1

Else

x = 0

y = 0

End If

MsgBox ("x= " & x)

MsgBox ("y= " & y)

End Sub

Блочная форма записи условного оператора имеет следующие особенности.

- предложение *If* должно стоять первым в строке;
- в блоке *If* допускается любое число вложений *If*, но ни одно такое предложение не может появляться после предложения *Else*;
- предложения *Else*, *ElseIf* и *End If* могут иметь перед собой только номер строки или метку строки;
- блок *If* должен обязательно заканчиваться инструкцией *End If*.
- после *Then* на той же строке может находиться только комментарий, иначе эта инструкция рассматривается как однострочная инструкция *If*.
- предложения *Else* и *ElseIf* являются не обязательными.
- допускаются вложенные блоки *If*.

4.3 Инструкция *Select Case*

В случае, когда необходимо выполнить строго определенные действия в зависимости от некоторого значения параметра, а общее число значений, которые может принимать этот параметр, достаточно велико, то можно использовать инструкцию *Select Case*.

Синтаксис:

Select Case Тестируемое выражение

Case Условие выбора1

Блок операторов1

.....

Case Условие выбораN

Блок операторовN

Case Else

Блок операторов

End Select

Работа оператора: оператор *Select Case* анализирует *Тестируемое значение* и ищет строку *Case*, удовлетворяющую данному значению. Когда такая строка найдена, выполняется *Блок операторов*, расположенный между соседними строками *Case*. Если ни одно условие *Case* не совпадает с *Тестируемым значением*, выполняется Блок операторов, следующий за *Case Else*

В строках *Case* можно указывать как отдельные значения, так и целый диапазон значений с помощью ключевых слов *To* и *Is*. Строка *Case Else* не является обязательной. Она позволяет указать список инструкций, которые будут выполнены в том случае, если не обнаружено ни одного совпадения с тестируемым выражением. Переменная в операторе *Select Case* может принимать также строковые значения.

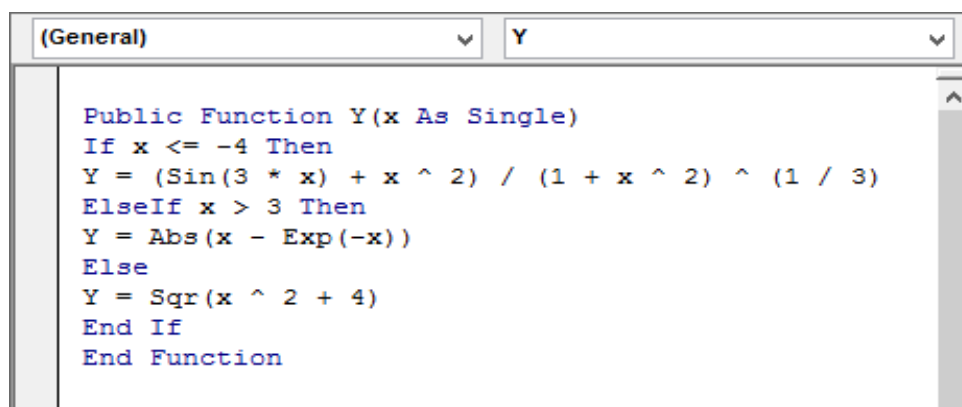
4.4 Лабораторная работа №3 «Программирование алгоритмов разветвляющейся структуры»

Пример 1. Разработать пользовательскую функцию и составить программу, вычисляющую значение функции:

$$y = \begin{cases} \frac{\sin 3x + x^2}{\sqrt[3]{1 + x^2}}, & \text{если } x \leq -4 \\ \sqrt{x^2 + 4}, & \text{если } -4 < x \leq 3 \\ |x - e^{-x}|, & \text{если } x > 3 \end{cases}$$

Методика выполнения:

Перейти в редактор VBA, добавить модуль и функцию командами **Module** и **Function** меню **Insert** задать имя функции – Y и записать программный код, как показано на рисунке 40.



```
(General) Y
Public Function Y(x As Single)
If x <= -4 Then
Y = (Sin(3 * x) + x ^ 2) / (1 + x ^ 2) ^ (1 / 3)
ElseIf x > 3 Then
Y = Abs(x - Exp(-x))
Else
Y = Sqr(x ^ 2 + 4)
End If
End Function
```

Рисунок 40 – Программный код для решения функции в примере 1

Добавить описание функции (Ф.И., группа) и проверить ее работу с различными значениями x.

Пример 2. Архитектор получает процент от суммы сделки при сдаче проекта. Для сделок, сумма которых менее 150 тыс. р., вознаграждение архитектора составит 3,5 %, от 150 до 500 тыс. р. – 4,5 %, а для сделок, сумма которых составляет больше 500 тыс. р. – 5 %. Разработать пользовательскую форму и составить программу для расчета вознаграждения архитектора.

Методика выполнения:

Создать пользовательскую форму вида, представленного на рисунке 41. Свойства объектов формы представлены в таблице 11.

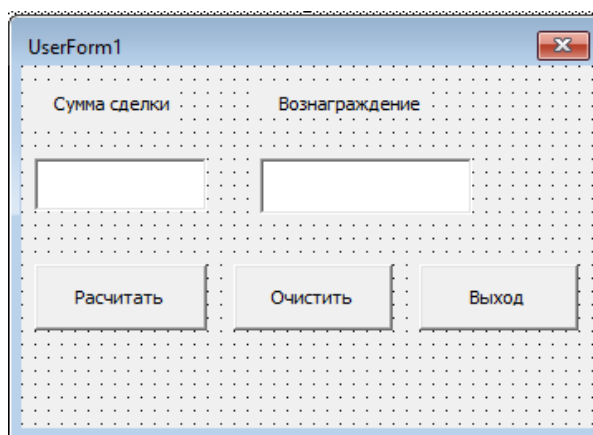


Рисунок 41 – Пользовательская форма к примеру 2

Таблица 11 – Свойства объектов формы примера 2

Объект	Свойство	Объект	Свойство
UserForm1	Caption= Вознаграждение архитектора	Label1	Caption = Сумма сделки AutoSize = True
CommandButton1	Name = calc Caption = Рассчитать	Label2	Caption = Вознаграждение AutoSize = True
CommandButton2	Name = clean Caption = Очистить	TextBox1	Name = txtSumma Locked = True
CommandButton3	Name = exitForm Caption = Выход	TextBox1	Name = txtVzngr Locked = True

Для обработки события нажатия кнопки «Рассчитать» ввести программный код, отображенный на рисунке 42:

```
exitForm Click
Private Sub calc_Click()
Dim S, P As Single
S = Val(InputBox("Введите сумму сделки", "Ввод данных"))
If S < 150000 Then
P = S * 0.035
ElseIf (S >= 150000) And (S < 500000) Then P = S * 0.045
Else
P = S * 0.05
End If
txtSumma.Text = Str(S) + " руб"
txtVzngr.Text = Str(P) + "руб"
End Sub
```

Рисунок 42 – Программный код для расчета значений примера 2

Ввести программный код для кнопок «Очистить» и «Выход», как показано на рисунке 43 и создать для запуска формы кнопку.

```
Private Sub clean_Click()
txtSumma.Text = " "
txtVzngr.Text = " "
End Sub

Private Sub exitForm_Click()
End
End Sub
```

Рисунок 43 – Программный код обработки событий кнопок *Очистить* и *Выход*

Пример 3. По введенной дате рождения сообщить пользователю его знак зодиака.

Методика выполнения:

Создать пользовательскую форму вида, представленного на рисунке 44. Свойства объектов формы представлены в таблице 12.

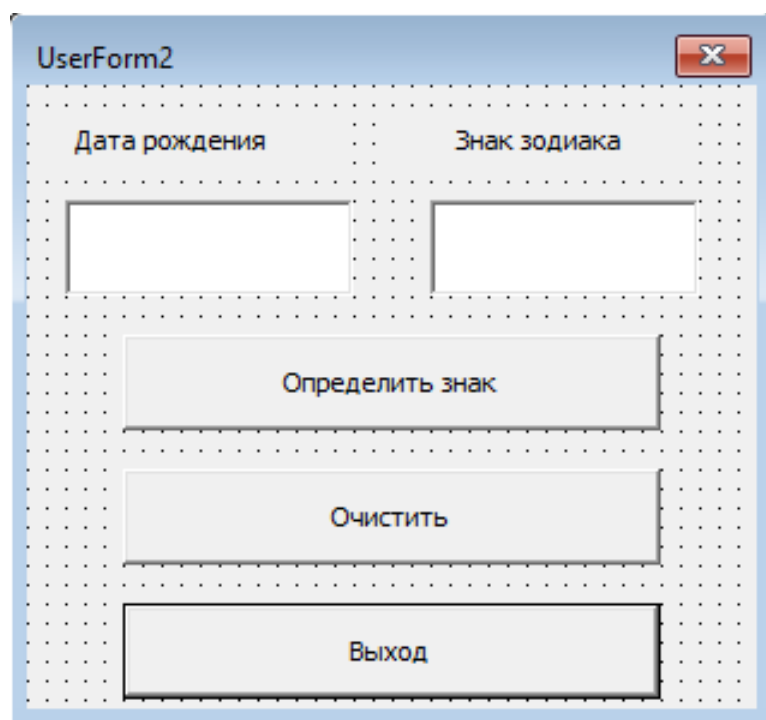


Рисунок 44 – Пользовательская форма к примеру 3

Таблица 12 – Свойства объектов формы примера 3

<i>Объект</i>	<i>Свойство</i>	<i>Объект</i>	<i>Свойство</i>
UserForm2	Caption= Знака зодиака	Label1	Caption Дата рождения AutoSize = True
CommandButton1	Name = calc Caption = Определить знак	Label2	Caption = Знак зодиака AutoSize = True
CommandButton2	Name = clean Caption = Очистить	Text-Box1	Name = txtSumma Locked = True
CommandButton3	Name = exitForm Caption = Выход	Text-Box1	Name = txtVzngnr Locked = True

Чтобы определить знак зодиака по введенной пользователем дате рождения, необходимо воспользоваться оператором выбора *Select Case*. Кроме этого для выделения из даты месяца и числа потребуются функции *Month* (Дата) и *Day* (Дата), которые возвращают соответственно месяц как целое число от 1 до 12 и день как целое число от 1 до 31.

Таким образом, код обработки события нажатия кнопки «Определить знак» будет иметь вид, отображенный на рисунке 45. Ввести программный код для кнопок «Очистить» и «Выход» и создать для запуска формы кнопку.

```
clean Click
Private Sub calc_Click()
Dim DR As Date
DR = (txtData.Text)
Select Case Month(DR) ' выбор знака зодиака по номеру месяца
Case 1
If Day(DR) <= 20 Then txtZodiak.Text = "Козерог" Else txtZodiak.Text = "Водолей"
Case 2
If Day(DR) <= 19 Then txtZodiak.Text = "Водолей" Else txtZodiak.Text = "Рыбы"
Case 3
If Day(DR) <= 20 Then txtZodiak.Text = "Рыбы" Else txtZodiak.Text = "Овен"
Case 4
If Day(DR) <= 20 Then txtZodiak.Text = "Овен" Else txtZodiak.Text = "Телец"
Case 5
If Day(DR) <= 20 Then txtZodiak.Text = "Телец" Else txtZodiak.Text = "Близнецы"
Case 6
If Day(DR) <= 21 Then txtZodiak.Text = "Близнецы" Else txtZodiak.Text = "Рак"
Case 7
If Day(DR) <= 22 Then txtZodiak.Text = "Рак" Else txtZodiak.Text = "Лев"
Case 8
If Day(DR) <= 23 Then txtZodiak.Text = "Лев" Else txtZodiak.Text = "Дева"
Case 9
If Day(DR) <= 22 Then txtZodiak.Text = "Дева" Else txtZodiak.Text = "Весы"
Case 10
If Day(DR) <= 23 Then txtZodiak.Text = "Весы" Else txtZodiak.Text = "Скорпион"
Case 11
If Day(DR) <= 22 Then txtZodiak.Text = "Скорпион" Else txtZodiak.Text = "Стрелец"
Case Else
If Day(DR) <= 21 Then txtZodiak.Text = "Стрелец" Else
txtZodiak.Text = "Козерог"
End Select
End Sub
```

Рисунок 45 – Программный код обработки события определения знака зодиака

Индивидуальное задание №1:

Используя редактор VBA MS Excel, разработать пользовательскую функцию и составить программу, вычисляющую значение функции в соответствии с вариантом задания (таблица 13), добавить описание функции (Ф.И., группа) и проверить ее работу с различными значениями х.

Таблица 13 – Варианты индивидуального задания 1 лабораторной работы 3

<i>№ вар.</i>	<i>Функция</i>
1	2
1	$y = \begin{cases} \cos^2(x + 1), & \text{если } 1 < x \leq 3 \\ 2 \sin(x - 3), & \text{если } -3 < x \leq 1 \\ x, & \text{в остальных случаях} \end{cases}$
2	$y = \begin{cases} \frac{\cos^2 x}{2}, & \text{если } -1 < x \leq 3 \\ \sin(x + 1), & \text{если } 3 < x \leq 10 \\ \sqrt{x} + 1, & \text{в остальных случаях} \end{cases}$
3	$y = \begin{cases} \frac{1}{2}x^2 - \sin x, & \text{если } x \leq 10 \\ e^{x+1}, & \text{если } 10 < x \leq 15 \\ \sin(x^2 + 1), & \text{если } x > 15 \end{cases}$
4	$y = \begin{cases} \frac{\cos 3x + x^2}{\sin^3 x - 3x}, & \text{если } x = 2 \\ \sin^2 x + \sqrt{x^2 + 2}, & \text{если } x > 2 \\ x^3 \log_2^3 \arctg x - e^{-x} , & \text{если } x < 2 \end{cases}$
5	$y = \begin{cases} x^2 - \operatorname{tg} x, & \text{если } -2 \leq x \leq 2 \\ e^{x+1} + \sqrt{x^3}, & \text{если } x > 12 \\ \sin x^2 + \ln x , & \text{если } x < 12 \end{cases}$
6	$y = \begin{cases} \sin x^2 - \sqrt{\ln \sin x }, & \text{если } 0 < x \leq 5 \\ \frac{3}{4}e^{x+1}, & \text{если } 5 < x \leq 10 \\ 3 \cos x + \sin(x^2 + 1), & \text{если } x > 10 \end{cases}$
7	$y = \begin{cases} \sin^2(x + 1), & \text{если } -2 < x \leq 0 \\ \cos(x - 3), & \text{если } x > 1 \\ \frac{x}{2} + e^{x+1}, & \text{если } 0 < x \leq 1 \end{cases}$

Продолжение таблицы 13

1	2
8	$y = \begin{cases} 2 \sin^2 x, & \text{если } x > 1 \\ \sqrt{1 + \frac{x^2}{1+x^2}}, & \text{если } x < 0 \\ e^{x+1}, & \text{если } 0 < x \leq 1 \end{cases}$
9	$y = \begin{cases} \frac{1+2x}{1+x^2}, & \text{если } x < 0 \\ \sin^2 x \sqrt{1+x}, & \text{если } 0 \leq x < 1 \\ \sin^2 x e^{0.2x}, & \text{если } x \geq 1 \end{cases}$
10	$y = \begin{cases} x ^{1/3}, & \text{если } x < 0 \\ \frac{x}{1+x} - 2x, & \text{если } 0 < x \leq 1 \\ \frac{ 3-x }{1+x}, & \text{если } x > 1 \end{cases}$
11	$y = \begin{cases} \frac{1+x}{1+x^2}, & \text{если } x < 0 \\ \sqrt{1 + \frac{x}{1+x}}, & \text{если } 0 \leq x < 1 \\ 2 \sin 3x , & \text{если } x \geq 1 \end{cases}$
12	$y = \begin{cases} 1 + \frac{3+x}{1+x^2}, & \text{если } x < 0 \\ \sqrt{1 + (1-x)^2}, & \text{если } 0 \leq x < 1 \\ \frac{1+x}{1+\cos^2 x}, & \text{если } x \geq 1 \end{cases}$
13	$y = \begin{cases} \frac{1+x}{\sqrt[3]{1+x^2}}, & \text{если } x < 0 \\ 2e^{-2x} - x, & \text{если } 0 \leq x < 1 \\ 2-x ^{1/3}, & \text{если } x \geq 1 \end{cases}$

Продолжение таблицы 13

1	2
14	$y = \begin{cases} \frac{ x }{x + x^2}, & \text{если } x < 0 \\ \sqrt{1 + x^3}, & \text{если } 0 \leq x < 2 \\ \frac{1 + \sin x}{1 + x} + 3x, & \text{если } x \geq 2 \end{cases}$
15	$y = \begin{cases} \frac{1 + x + x^2}{x + x^2}, & \text{если } x < 0 \\ \sqrt{1 + \frac{x^3}{1 + x}}, & \text{если } 0 \leq x < 3 \\ 2 1 + \sin x , & \text{если } x \geq 3 \end{cases}$

Индивидуальное задание №2:

В соответствии с вариантом задания (таблица 14) разработать пользовательскую форму и составить программу в редакторе VB MS Excel. Создать для запуска формы кнопку.

Таблица 14 – Варианты индивидуального задания 2 лабораторной работы 3

№ вар.	Задание
1	2
1	Работник строительной фирмы получает премию в размере некоторой доли от суммы проделанной работы: если объем работы до 6000 р., то в размере 5 %; если объем до 15 0000 р. – 7 %; если выше – 10 %. Определить размер премии работника.
2	Рабочий на стройке получает доплату за работу свыше нормы (8 часов в день). При дополнительной работе до 2 часов – 500 руб., от 2 до 4 часов – 700 р. свыше 4 ч. – 1000 р. Определить размер доплаты рабочего .

Продолжение таблицы 14

1	2
3	Рассчитать заработную плату сотруднику строительной фирмы с учётом подоходного налога. Подоходный налог высчитывается по следующей схеме: если зарплата составляет менее 8200 р., то налог не взимается, если от 8200 р. до 100 тыс. р., то он составляет 13 %, а если зарплата составляет более 100 тыс. р., подоходный налог составит 15% от начисленной суммы.
4	Менеджер по продажи недвижимости получает премию по итогам работы за месяц по следующей схеме: если заключено сделок менее чем на 50 000 р., то премия составляет 5 % от общей стоимости сделок, если не меньше чем на 50 000 р. но менее чем на 150 000 р. – 7,5 %, если же не менее 150 000 р. – 11,3 %. Определить размер премии сотрудника.
5	Рассчитываясь в строительном магазине, покупатель получает скидку: если количество покупаемого им товара не менее 100, то скидка составляет 10 %, если от 50 до 100, то 5 %, в остальных случаях скидка не предоставляется.
6	Определить стоимость квартиры с учетом следующих расценок: площади до 20 м ² – 350 руб за м ² , от 20 до 40 м ² – 340 руб за м ² , остальные площади – 320 руб за м ² .
7	Рассчитать стоимость заказа в магазине, если действуют следующие расценки: покупка до 100 мешков строительной смеси – 250 р. за мешок; от 100 до 1000 мешков – 240 р.; свыше 1 000. – 225 р. за мешок.
8	Исходя из длительности и тарифа S р. за минуту разговора по телефону определить стоимость разговора по телефону с учетом скидки: по субботам предоставляется скидка в 15 %, по воскресеньям в 20 %.
9	Рассчитать стоимость ремонтных работ помещения в зависимости от площади до 50 м ² – 95 руб за м ² , от 20 до 40 м ² – 93 руб за м ² , остальные площади – 90 руб за м ²
10	Билет на пригородном поезде стоит 350 р., если расстояние до станции не больше 20 км; 530 р., если расстояние более 20 км, но не превышает 75 км; 600 р., если расстояние больше 75 км. Определить стоимость билета.
11	Рассчитать стоимость заказа в типографии, если действуют следующие расценки: печать до 100 экземпляров – 10 р. за лист; от 100 до 1000 экземпляров – 7 р. за лист; свыше 1 000. – 5 р. за лист.

Продолжение таблицы 14

1	2
12	Продавец-консультант в строительном магазине получает комиссионные, начисляемые по следующей схеме: если продукции продано не меньше чем на 10 000 р., то процент комиссионных составляет 2% от стоимости реализованной продукции, если продано от 10 000р. до 30 000р. – 5%, если свыше 30 000 р. – 7 %.
13	Определить стоимость аренды для площади с учетом следующих расценок: аренда площади до 20 м ² – 150 руб за м ² , аренда площади свыше 150 м ² – 70 руб за м ² , остальные площади – 100 руб за м ²
14	На банковский вклад начисляются проценты в сумме 15 % годовых, если сумма вклада. превышает 100 000 р., 10 % годовых на суммы до 50 000 р., 12 % годовых на остальные суммы. Рассчитать сумму полученных вкладчиком процентов по истечении срока.
15	Менеджер по продаже недвижимости получает комиссионные в зависимости от объема продаж за месяц: если объем продаж менее 100 000 р., то ставка комиссионных составляет 8 %, если от 100 000 до 200 000 р. – 10,5 %, более 200 000 р. – 14 %. Определить сумму комиссионных менеджера.

4.5 Вопросы для самопроверки

1. В каких случаях используются операторы ветвления?
2. Чем отличается альтернативное ветвление от безальтернативного?
3. Какой синтаксис имеет безальтернативное ветвление?
4. Какой синтаксис альтернативного ветвления?
5. В чем отличие блочной и однострочной формы записи условного оператора?
6. Какая инструкция условного оператора используется при вложенных условиях?
7. Какое число вложений *If* допускается в блоке *If*?
8. Какое служебное слово используется для завершения условного оператора?

9. В чем особенность инструкции *Select Case* в каких случаях она применяется?

5 Программирование циклических вычислительных процессов

5.1 Команды организации циклов

Какие-либо действия процедуры, повторяющиеся заданное количество раз, или пока выполняется, или не выполняется некоторое условие называют **циклом**.

Процесс выполнения всех операторов, заключенных в структуру цикла один раз, называется *итерацией цикла*.

Структуры цикла, всегда выполняющиеся заданное количество раз, называются *циклами с фиксированным числом итераций*. Другие типы структур цикла повторяются переменное количество раз в зависимости от некоторого набора условий. Такие циклы называются *неопределенными циклами*.

Блок операторов, находящийся между началом и концом цикла, называется *"тело цикла"*.

5.2 Циклы с фиксированным числом итераций

Цикл **For..Next**

Синтаксис цикла *For..Next*:

For Счетчик = Начало **To** Конец [*Step Шаг*]

Операторы

Next [Счетчик]

В квадратных скобках указаны необязательные атрибуты цикла *For..Next*.

Счетчик - любая численная переменная VBA

Начало - любое численное выражение, определяет начальное значение для переменной Счетчик

Конец - численное выражение, определяет конечное значение для переменной Счетчик

Операторы - один, несколько или ни одного оператора VBA (тело цикла).

Step - необязательный атрибут, числовое выражение, задающее шаг цикла. Может быть как положительным, так и отрицательным. Если атрибут *Step* отсутствует, цикл *For..Next* выполняется с шагом по умолчанию, равному 1

По умолчанию VBA увеличивает переменную *Счетчик* на 1 каждый раз при выполнении операторов в цикле. Можно задать другое значение (*Step Шаг* - любое численное выражение), на которое будет изменяться *Счетчик*.

Ключевое слово *Next* сообщает VBA о том, что достигнут конец цикла. Необязательная переменная *Счетчик* после ключевого слова *Next* должна быть той же самой переменной *Счетчик*, которая была задана после ключевого слова *For* в начале структуры цикла.

Пример 1. Посчитать сумму цифр от 1 до 10:

```
Sub example_01()
```

```
Dim i As Integer, SUM As Integer
```

```
SUM = 0
```

```
For i = 1 To 10
```

```
    SUM = SUM + i
```

```
Next i
```

```
    MsgBox "Сумма чисел от 1 до 10 равна " & SUM
```

```
End Sub
```

Окно сообщений, сгенерированное функцией `MsgBox` для примера 1 представлено на рисунке 46.

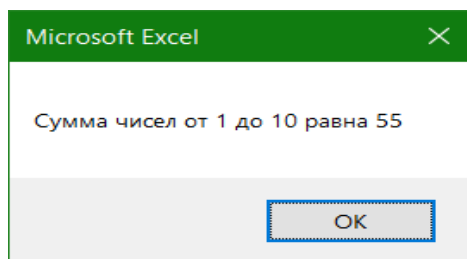


Рисунок 46 – Окно сообщений, сгенерированное функцией MsgBox для примера 1

Пример 2. Посчитать сумму четных чисел от 1 до 10 и сумму нечетных чисел от 1 до 10 (варианты цикла *For..Next* с использованием шага цикла отличного от единицы):

```
Sub example_02()  
    Dim i As Integer, SUM As Integer, SUM2 As Integer  
    SUM = 0: SUM2 = 0  
    For i = 1 To 10 Step 2  
        SUM = SUM + i  
    Next i  
    For i = 10 To 1 Step -2  
        SUM2 = SUM2 + i  
    Next i  
    MsgBox "Сумма чисел от 1 до 10 равна " & SUM1 & vbCr _  
        & "Сумма четных чисел от 1 до 10 равна " & SUM2  
End Sub
```

Окно сообщений, сгенерированное функцией MsgBox для примера 2, представлено на рисунке 47.

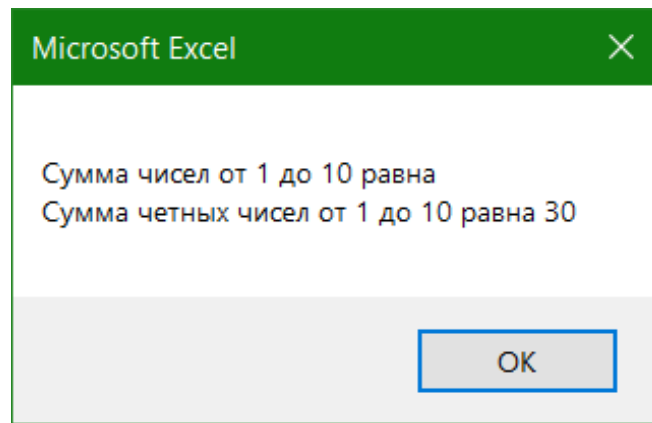


Рисунок 47 – Окно сообщений, сгенерированное функцией MsgBox для примера 2

Обратите внимание! При уменьшении счетчика цикла *For..Next* цикл выполняется, пока переменная счетчика больше или равна конечному значению, а когда счетчик цикла увеличивается, цикл выполняется, пока переменная счетчика меньше или равна конечному значению.

Цикл **For Each..Next**

Цикл *For Each..Next* не использует счетчик цикла. Циклы *For Each..Next* выполняются столько раз, сколько имеется элементов в определенной группе, такой как коллекция объектов или массив (которые будут рассматриваться позже). Проще говоря, цикл *For Each..Next* выполняется один раз для каждого элемента в группе.

Синтаксис цикла *For Each..Next*:

For Each Элемент ***In*** Массив

Операторы

Next [Элемент]

Элемент - переменная, используемая для итерации по всем элементам в определенной группе

Массив - это объект коллекции или массив

Операторы - один, несколько или ни одного оператора VBA (тело цикла).

Цикл *For Each..Next* всегда выполняется столько раз, сколько имеется элементов в определенной группе.

Пример 3. С помощью функции SheetExists, использующей цикл *For Each..Next* определить, существует ли определенный лист в рабочей книге Excel:

```
Function SheetExists(sName As String) As Boolean
```

```
Dim aSheet As Object
```

```
SheetExists = False
```

```
'цикл по всем листам со сравнением имен с sName
```

```
For Each aSheet In ActiveWorkbook.Sheets
```

```
    If (StrComp(aSheet.Name, sName, 1) = 0) Then
```

```
        SheetExists = True
```

```
        Exit For
```

```
    End If
```

```
Next aSheet
```

```
End Function
```

```
Sub Test_SheetExists()
```

```
'тестирование функции
```

```
Dim Stroka As String
```

```
Stroka = InputBox("Введите имя листа " & " текущей книги: ")
```

```
If SheetExists(Stroka) Then
```

```
    MsgBox "Лист" & Stroka & " в текущей книге имеется"
```

```
Else
```

```
    MsgBox "Лист" & Stroka & " в текущей книге отсутствует"
```

```
End If
```

```
End Sub
```

5.3 Неопределенные циклы

Существуют два основных способа создания неопределенного цикла. Можно построить цикл так, что VBA будет тестировать некоторое условие (*детерминант цикла*) либо *перед выполнением цикла*, либо - *после выполнения цикла*.

Цикл Do .. While - тестирует свое условие детерминанта до выполнения цикла.

Синтаксис цикла *Do .. While*:

Do While Условие

Операторы

Loop

Условие - логическое выражение для детерминанта цикла

Операторы - один, ни одного или несколько операторов, которые составляют тело цикла

Loop - ключевое слово, указывает на окончание тела цикла и обозначает место, из которого VBA возвращается в начало цикла для проверки условия

VBA выполняет цикл пока логическое выражение, представленное с помощью *Условия*, равно *True*.

При выполнении цикла *Do While* сначала тестируется логическое выражение (*Условие*); если оно равно *True* - выполняется тело цикла. При достижении ключевого слова *Loop* управление опять передается в начало цикла и снова проверяется логическое выражение. Так происходит до тех пор, пока логическое выражение не станет *False*. Когда логическое выражение становится *False* - управление передается оператору, следующему за ключевым словом *Loop*.

Обратите внимание! Если логическое выражение равно *False* при первом выполнении цикла *Do While*, то управление сразу передается оператору, следующему за *Loop*, а операторы, находящиеся в теле цикла соответственно пропускаются. Другими словами говоря, цикл *Do While* позволяет ни разу не выполнять операторы внутри него.

Пример 4. Подсчитать сумму цифр от 1 до 10, используя цикл *Do While*:

Sub example_04()

Dim i As Integer, SUM As Integer

i = 1: SUM = 0

Do While i <= 10

SUM = SUM + i

i = i + 1

Loop

MsgBox ("Цикл Do While - Сумма цифр от 1 до 10 = " & SUM)

End Sub

Окно сообщений, сгенерированное функцией `MsgBox` для примера 4, представлено на рисунке 48.

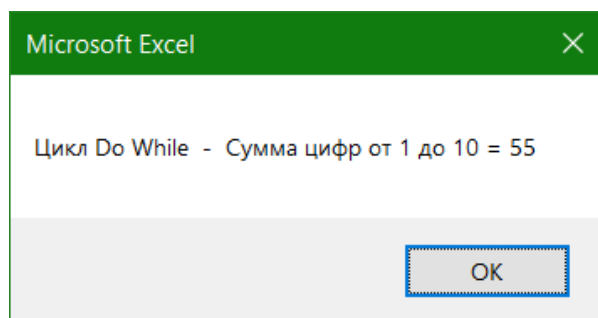


Рисунок 48 – Окно сообщений, сгенерированное функцией `MsgBox` для примера 4

Цикл Do .. Until - еще один цикл, тестирующий условие детерминанта до выполнения цикла.

Синтаксис цикла *Do .. Until*:

Do Until Условие

Операторы

Loop

Условие - логическое выражение для детерминанта цикла

Операторы - один, ни одного или несколько операторов, которые составляют тело цикла

Loop - ключевое слово, указывает на окончание тела цикла и обозначает место, из которого VBA возвращается в начало цикла для проверки условия

VBA выполняет цикл пока логическое выражение, представленное с помощью *Condition*, равно *False*. В остальном цикл *Do Until* полностью аналогичен циклу *Do While*.

Пример 5. Подсчитать сумму цифр от 1 до 10, используя цикл *Do Until*

```
Sub example_05()
```

```
Dim i As Integer, SUM As Integer
```

```
    i = 1: SUM = 0
```

```
    Do Until i > 10
```

```
        SUM = SUM + i
```

```
        i = i + 1
```

```
    Loop
```

```
    MsgBox ("Цикл Do Until - Сумма цифр от 1 до 10 = " & SUM)
```

```
End Sub
```

Для тестирования условий после выполнения тела цикла надо поместить логическое выражение в конец блока операторов, составляющих тело цикла, после ключевого слова *Loop*, которое сообщает о конце цикла.

Цикл Do .. Loop While - конструкция цикла, тестирующая свое условие детерминанта после выполнения цикла.

Синтаксис цикла *Do .. Loop While*:

Do

Операторы

Loop While Условие

Условие - логическое выражение для детерминанта цикла

Операторы - один, ни одного или несколько операторов, которые составляют тело цикла

Loop - ключевое слово, указывает на окончание тела цикла и обозначает место, из которого VBA возвращается в начало цикла после проверки условия

VBA выполняет цикл пока логическое выражение, представленное с помощью *Условия*, равно *True*. При выполнении цикла *Do Loop While* сначала выполняются операторы тела цикла, затем по достижении ключевого слова *Loop* тестируется логическое выражение (*Условие*); если оно равно *True* - управление передается в начало тела цикла и цикл повторяется снова. Так происходит до тех пор, пока логическое выражение не станет *False*. Когда логическое выражение становится *False* - управление передается оператору, следующему за строкой *Loop While*...

Обратите внимание! Даже если при первом выполнении цикла *Do Loop While* логическое выражение равно *False* тело цикла всё равно будет выполнено. То есть, независимо от значения логического выражения, представленного с помощью *Условия*, этот цикл всегда выполняется, как минимум один раз.

Пример 6. Подсчитать сумму цифр от 1 до 10, используя цикл *Do Loop While*

Sub example_06()

Dim i As Integer, SUM As Integer

i = 1: SUM = 0

Do

SUM = SUM + i

i = i + 1

Loop While i <= 10

MsgBox ("Цикл Do Loop While - Сумма цифр от 1 до 10 = " & SUM)

End Sub

Цикл Do .. Loop Until - еще один цикл, тестирующий условие детерминанта после выполнения цикла.

Синтаксис цикла *Do .. Loop Until*:

Do

Операторы

Loop Until Условие

Условие - логическое выражение для детерминанта цикла

Операторы - один, ни одного или несколько операторов, которые составляют тело цикла

Loop - ключевое слово, указывает на окончание тела цикла и обозначает место, из которого VBA возвращается в начало цикла после проверки условия

VBA выполняет цикл пока логическое выражение, представленное с помощью *Условия*, равно *False*. В остальном цикл *Do Loop Until* полностью аналогичен циклу *Do Loop While*.

Пример 7. Подсчитать сумму цифр от 1 до 10, используя цикл *Do Loop Until*

```
Sub example_07()
```

```
Dim i As Integer, SUM As Integer
```

```
    i = 1: SUM = 0
```

```
    Do
```

```
        SUM = SUM + i
```

```
        i = i + 1
```

```
    Loop Until i > 10
```

```
    MsgBox ("Цикл Do Loop Until - Сумма цифр от 1 до 10 = " & SUM)
```

```
End Sub
```

5.4 Вложенные циклы

Циклы можно помещать внутрь друг друга. Помещение одной структуры цикла в другую называется *вложением циклов*. Можно делать вложение структур циклов разного типа.

При вложении циклов надо соблюдать определенные правила:

- При вложении циклов *For..Next* каждый цикл должен иметь свою уникальную переменную счетчика;
- При вложении циклов *For Each..Next* каждый цикл должен иметь свою уникальную *element*-переменную;
- Если используется оператор *Exit For* или *Exit Do* во вложенном цикле, этим оператором заканчивается только выполняемый в данный момент цикл. VBA продолжает выполнение следующего цикла более высокого уровня.

Пример 8. Построить изображение шахматной доски произвольной размерности (заданной пользователем) и пронумеровать клетки – слева направо, сверху вниз.

```
Sub example_08 ()
  Dim N As Integer
  N = InputBox("Введите размерность шахматной доски", _
              "Ввод N", "10")
  For i = 1 To N
    For j = 1 To N
      k = N * (i - 1) + j
      Cells(i + 1, j + 1) = k
      Cells(i + 1, j + 1).Borders.LineStyle = 1
      If (i + j) Mod 2 = 0 Then
        Cells(i + 1, j + 1).Interior.Color = _
          RGB(250, 250, 50)
        Cells(i + 1, j + 1).Font.Color = _
          RGB(50, 50, 50)
      Else
        Cells(i + 1, j + 1).Interior.Color = _
          RGB(50, 50, 50)
        Cells(i + 1, j + 1).Font.Color = _
          RGB(250, 250, 50)
      End If
    Next
  Next
End Sub
```

Результат выполнения программы представлен на рисунке 49.

	A	B	C	D	E	F	G	H	I	J	K
1											
2		1	2	3	4	5	6	7	8	9	10
3		11	12	13	14	15	16	17	18	19	20
4		21	22	23	24	25	26	27	28	29	30
5		31	32	33	34	35	36	37	38	39	40
6		41	42	43	44	45	46	47	48	49	50
7		51	52	53	54	55	56	57	58	59	60
8		61	62	63	64	65	66	67	68	69	70
9		71	72	73	74	75	76	77	78	79	80
10		81	82	83	84	85	86	87	88	89	90
11		91	92	93	94	95	96	97	98	99	100

Рисунок 49 – Сквозная нумерация клеток шахматной доски

В этой программе используются средства языка VBA for Excel:

- 1) для обрамления клеток рамкой используется свойство клетки **Borders.LineStyle**;
- 2) для закрашивания клеток определёнными цветами с помощью функции **RGB**, значение которой присваивается свойству клетки **Interior.Color**;
- 3) для выбора цвета шрифта надписи на клетке с помощью функции **RGB**, значение которой присваивается свойству клетки **Font.Color**.

Функция **RGB**, как нетрудно догадаться, возвращает цвет в соответствии со значениями трёх своих аргументов (соответственно, *красного*, *зелёного* и *синего*). Например, функция **RGB(250, 250, 50)** возвращает цвет, близкий к жёлтому (равные доли красного и зелёного дают жёлтый цвет), а функция **RGB(50, 50, 50)** возвращает тёмно-серый цвет.

Прерывание выполнения макроса или процедуры

Зачастую программисты допускают ошибки в написании кода, которые приводят к закликиванию программы (бесконечный цикл). Для прерывания выполнения

VBA необходимо нажать клавишу Esc или комбинацию клавиш Ctrl+Break. VBA заканчивает выполнение текущего оператора, переходит в состояние ожидания и отображает окно сообщения о runtime-ошибке.

5.5 Лабораторная работа №4 «Программирование алгоритмов циклической структуры»

Пример 1

Разработать пользовательскую форму и составить программу для вычисления

суммы ряда $\sum_{i=1}^n \frac{i!}{i+1}$

Методика выполнения:

Перейти в редактор VBA добавить, создать форму, как показано на рисунке 50, в названии формы указать № примера и Ф.И.

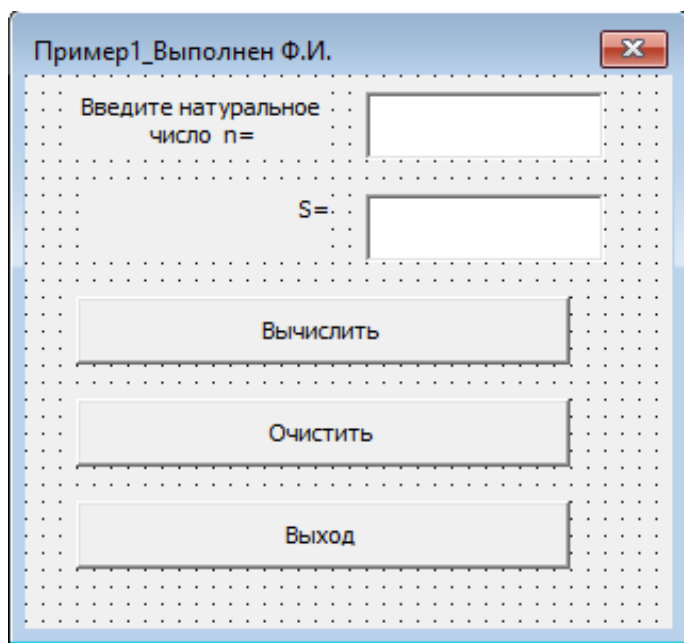


Рисунок 50 – Пользовательская форма

Таблица 15 – Свойства объектов формы примера 1

Объект	Свойство	Объект	Свойство
UserForm1	Caption = Пример1_Выполнен Ф.И.	TextBox2	Name = txtS
Label1	Caption= Введите натуральное число n AutoSize = True TextAlign = 2-fmTextAlignCenter	Command Button1	Name = calc Caption=Вычислить
TextBox1	Name = txtN	Command Button2	Name = clean Caption= Очистить
Label2	Caption= S= AutoSize = True TextAlign =3-fmTextAlignRight	Command Button3	Name = ExitForm Caption=Выход

Для обработки события нажатия кнопки «Рассчитать» введем следующий код:

```
Private Sub calc_Click()
```

```
Dim s, f As Double
```

```
Dim i, n As Integer
```

```
n = Val(txtN.Text) ' ввод значения n
```

```
s = 0 ' начальное значение суммы
```

```
f = 1 ' начальное значение произведения (факториала)
```

```
For i = 1 To n
```

```
f = f * i
```

```
s = s + f / (i + 1)
```

```
Next i
```

```
txtS.Text = s
```

```
End Sub
```


Ввести код для обработки событий при нажатии кнопок «Очистить» и «Выход». Предусмотреть для запуска формы кнопку.

Пример 2. Разработать пользовательский интерфейс и составить программу для вычисления задачи: за сколько лет в банке при начальном вкладе 10 000 р. и годовой процентной ставке 12 % будет накоплена сумма более чем N р.

Методика выполнения:

Перейти на чистый лист, ввести описание условия задачи. На вкладке **Разработчик**, панели инструментов **Элементы управления** включить режим конструктора и добавить Элементы ActiveX «Текстовое поле» и 2 элемента «Кнопка» как показано на рисунке 51.

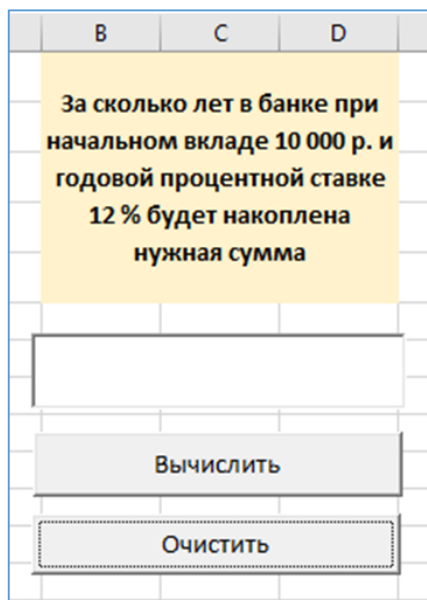


Рисунок 51 – Пользовательский интерфейс примера 2

Таблица 15 – Свойства объектов листа

Объект	Свойство
TextBox1	Name = txtS
CommandButton1	Name = calc Caption=Вычислить
CommandButton2	Name = clean Caption= Очистить

Для обработки событий нажатия кнопок «Рассчитать» и «Очистить» перейти в режим конструктора, дважды щелкнуть по кнопке и ввести в модуле следующий программный код:

```
Private Sub calc_Click()
    Dim s, n As Double
    Dim k As Integer
    n = Val(InputBox("Задайте нужную сумму ", "Ввод данных"))
    k = 0      ' начальное значение количества лет
    s = 10000 ' начальное значение суммы
    Do While s < n
        s = s + s * 0.12
        k = k + 1
    Loop
    txtS.Text = k
End Sub
```

```
Private Sub clean_Click()
    txtS.Text = ""
End Sub
```

Индивидуальное задание №1

В соответствии с вариантом задания разработать пользовательскую форму и составить программу в редакторе VBA MS Excel. Для запуска формы использовать кнопку, в имени формы указать № задания и свою Ф.И.

Таблица 16 – Варианты индивидуального задания1 лабораторной работы 4

№ вар.	Задание
1	2
1	Дано натуральное число n, действительное x. Вычислить произведение: $\prod_{k=1}^n \left(1 + \frac{\sin(kx)}{k!} \right)$

Продолжение таблицы 16

1	2
2	Даны натуральное n , действительное x . Вычислить сумму: $\sin x + \sin x^2 + \dots + \sin x^n$
3	Даны натуральное число n , действительное число x . Вычислить сумму: $\sum_{i=1}^n \frac{x + \cos(ix)}{2^i}$
4	Даны натуральное n , действительное x . Вычислить сумму: $\sum_{i=1}^n \left(\frac{1}{i!} + \sqrt{ x } \right)$
5	Дано натуральное число n . Вычислить сумму: $\sum_{k=1}^n \frac{1}{(2k+1)^2}$
6	Даны натуральное n , действительное x . Вычислить сумму: $\sin x + \sin^2 x + \dots + \sin^n x$
7	Дано действительное число a , натуральное число n . Вычислить произведение: $a(a+1)\dots(a+n-1)$
8	Даны натуральное число n , действительное число x . Вычислить произведение: $\prod_{k=1}^n \left(\frac{1}{k+1} - \cos^k(x) \right)$
9	Даны натуральное число n , действительное число x . Вычислить произведение: $\prod_{k=1}^n \left(1 + \frac{\sin(kx)}{k!} \right)$
10	Дано натуральное число n . Вычислить произведение: $\prod_{i=1}^n \left(2 + \frac{1}{i!} \right)$
11	Дано действительное число x . Вычислить: $x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!}$
12	Дано действительное число a , натуральное число n . Вычислить: $\frac{1}{a} + \frac{1}{a(a+1)} + \dots + \frac{1}{a(a+1)\dots(a+n)}$
13	Вычислить произведение: $\prod_{i=2}^{30} \frac{i^2 + 1}{i!}$
14	Дано натуральное число n . Вычислить: $\prod_{i=2}^n \frac{i+1}{i+2}$

Индивидуальное задание №2

В соответствии с вариантом задания разработать пользовательский интерфейс и составить программу в редакторе VBA MS Excel.

Таблица 17 – Варианты индивидуального задания 2 лабораторной работы 4

№ вар.	Задание
1	2
1	Владелец строительной фирмы закупил оборудование на n тыс. руб. Каждый месяц он получает прибыль в размере 20 % от потраченной суммы, через сколько месяцев он полностью возместит потраченные деньги.
2	Гражданин открыл счет в банке, вложив 60 000 р. Через каждый месяц размер вклада увеличивается на 1,7 % от имеющейся суммы. Определить сумму вклада через n месяцев (n – некоторое натуральное число).
3	Дано натуральное число N . Используя цикл While найти сумму первых N натуральных чисел кратных 3
4	Одноклеточная амёба каждые 3 часа делится пополам. Определить, сколько амёб будет через 24 часа, если первоначально было n амёб. (n – некоторое натуральное число).
5	Издательство планирует выпустить: первую книгу тиражом в 55 экземпляров, а каждую следующую тиражом, в 2 раза превышающим тираж предыдущей книги. Каким тиражом будет выпущена n -я книга.
6	Дано натуральное число N . Используя цикл While найти произведение четных чисел от 1 до N .
7	Вкладчик разместил в банке 45 000 р. Через каждый месяц размер вклада увеличивается на 1,5 % от имеющейся суммы. Определить, за какой месяц величина ежемесячного увеличения вклада превысит некоторое заданное число N .
8	Вкладчик разместил в банке N тыс. р. Через каждый месяц размер вклада увеличивается на 1,6 % от имеющейся суммы. Определить сумму вклада через год.
9	Завод выпустил в первый год N тыс изделий. В течении 5 лет они планируют каждый год увеличивать производительность выпуска изделий на 15%. Определить количество изделий, которое выпустит завод через 5 лет.
10	Гражданин открыл счет в банке, вложив 50 000 р. Через каждый месяц размер вклада увеличивается на 1,7 % от имеющейся суммы. Определить через сколько месяцев сумма вклада достигнет N тыс. р ($N > 50$).

Продолжение таблицы 17

1	2
11	Вкладчик разместил в банке N тыс. р. Через каждый месяц размер вклада увеличивается на 1,5 % от имеющейся суммы. Определить, за какой месяц величина ежемесячного увеличения вклада превысит 1000.
12	Завод выпустил в первый год 100 тыс изделий. Каждый год они планируют увеличивать производительность выпуска изделий на 9%. Определить количество изделий, которое выпустит завод через n лет. (n – некоторое натуральное число).
13	Определить, за сколько лет в банке при начальном вкладе 55 000 р. и годовой процентной ставке 17,5 % будет накоплена некоторая сумма

5.6 Вопросы для самопроверки

1. Что такое циклический оператор (оператор цикла)?
2. Какие два типа оператора цикла используются в языке Basic?
3. Как формально представляется синтаксис оператора цикла со счётчиком *For ... Next*?
4. Какие элементы синтаксиса оператора *For ... Next* являются «необязательными»?
5. Как формально представляется синтаксис оператора цикла с условием *Do ... Loop*?
6. Какие элементы синтаксиса оператора *Do ... Loop* являются «необязательными»?
7. В чём различие между смыслом проверки условий после ключевых слов *While* и *Until* в циклах с условием?
8. Как можно, при необходимости, обеспечить досрочный выход из цикла любого вида (как *For ... Next*, так и *Do ... Loop*)?

6 Одномерные массивы в среде VBA

6.1 Общие сведения

Массив – это упорядоченная последовательность данных, состоящая из фиксированного числа элементов, имеющих один и тот же тип, и обозначаемая одним именем.

Массиву присваивается имя, посредством которого можно ссылаться на него, как на единое целое. Элементы, образующие массив, упорядочены так, что каждому элементу соответствует совокупность номеров (индексов), определяющих его место в общей последовательности.

Доступ к каждому отдельному элементу осуществляется обращением к имени массива с указанием индекса нужного элемента: **<имя массива> (<индекс>)**.

Описание массива определяет его имя, размер массива и тип данных.

Одномерный массив – массив, у которого в описании задан только один индекс, если заданы два индекса – то это двумерный массив и т.д.

Одномерные массивы часто называют векторами, т.е. они представляют собой конечную последовательность пронумерованных элементов.

Общий вид описания одномерного массива:

Dim Имя_Массива (***Index1 To Index2***) ***As*** Тип_Элементов,

где *Index1* – нижний предел индексов массива, *Index2* – верхний предел индексов массива. Если в описании массива будет указан только один индекс, то нумерация элементов, по умолчанию, начинается с нуля.

Примеры описания одномерного массива:

1) **Dim x(8) As Integer** - в данном случае индексы элементов массива нумеруются, начиная с нуля, а количество элементов в массиве будет равно 9;

2) **Dim x(1 To 10) As Single** - объявляется одномерный массив вещественных значений, состоящий из 10 элементов, индексы элементов начинаются с 1;

3) **Dim x(2 To 11) As Integer** - при таком описании, объявляется одномерный массив целочисленных значений, состоящий из 10 элементов, индексы его элементов начинаются с 2.

Элемент одномерного массива обозначается путем указания имени массива и индекса, заключенного в круглые скобки, например, $Z(6)$.

Рабочий лист MS Excel имеет свойства **Range** и **Cells**, которые можно использовать для доступа к ячейкам в VBA. С помощью данных свойств можно выполнять запись элементов массива, как одномерных, так и многомерных в ячейки листа MS Excel, а также считывание этих элементов из ячеек.

В следующих примерах показано, как можно использовать свойство **Range** в кодах VBA.

1) $\text{Range}("A" \& i) = X(i)$ - позволяет поместить элементы массива X в ячейки столбца A , количество ячеек определяет переменная i .

2) $\text{Range}("C2") = k$ - выводит найденное значение k в ячейку $C2$.

3) $\text{Worksheets}("Лист1").\text{Range}("A" \& I).\text{ClearContents}$ – позволяет удалить содержимое I ячеек столбца A .

В следующих примерах показано, как можно использовать свойство **Cells**:

1) $X(i) = \text{Cells}(i, "A")$ 'i-ому элементу массива X присвоить значение ячейки расположенной на пересечении I -ой строки и столбца "A"

2) $\text{Cells}(k, 2) = Z(i)$ – в ячейку с номером строки k и номером столбца - 2 записать i элемент массива Z

3) $\text{Cells}(2, I) = x(I)$ - в ячейку с номером строки 2 и номером столбца i записать i элемент массива X .

Пример 1. Организовать ввод элементов одномерного массива с помощью оператора **Inputbox** и вывод элементов в виде таблицы на рабочий лист MS Excel. Отсортировать полученный массив по возрастанию значений элементов.

```
Private Sub Primer1()
```

```
Dim x(1 To 11) As Integer
```

```
Dim i, j, d, k As Integer
```

```
k = 10 'количество элементов массива
```

```

For i = 1 To k
x(i) = Val(InputBox("Введите" & i & "элемент массива", "Ввод данных"))
Range("A" & i + 1) = x(i) 'вывод элементов массива в столбец A
Next i

For i = 1 To 10 'сортировка по возрастанию
For j = 1 To 11
If x(i) < x(j) Then
d = x(i) 'значение промежуточной переменной
x(i) = x(j)
x(j) = d
End If
Next j
Next I

For I = 1 To 10
Cells(i + 1, "B") = x(i) 'вывод отсортированного массива в столбец B
Next i
End Sub

```

Пример 2. Указать индекс первого нулевого элемента в массиве.

```

Private Sub Primer2_Click()
Dim x(1 to 10) as Integer, I as Integer
For i=1 to 10
x(i)=Cells(I, "A") Next i
i=1
Do while x(i)>0
i=i+1
Loop
Range("C") = i
End sub

```


Пример 3. Известны данные о количестве купленных квартир в пяти строящихся домах на текущий момент строительства. Определить дом, в котором куплено больше всего квартир.

```
Private Sub Primer3_Click()  
Dim max, Nomer As Single  
Dim Kol(1 To 5) As Integer, i As Integer  
For i = 1 To 5  
Kol(i) = Val(InputBox("Введите количество квартир " & i & " дома", "Ввод  
данных"))  
Range("A" & i) = Kol(i)  
Next i  
max = Kol(1)  
k = 0  
For i = 1 To 5  
If Kol(i) > max Then  
max = Kol(i)  
Nomer = i  
End If  
Next i  
Range("C2") = max  
Range("C3") = Nomer  
End Sub
```

6.2 Лабораторная работа № 5 «Обработка одномерных массивов в среде VBA»

Пример 1. Посчитать количество отрицательных элементов в заданном массиве. Вывести отрицательные элементы массива в отдельный столбец.

Методика выполнения:

Создать шаблон с данными на листе Excel (рисунок 52). Исходный массив задать в столбце А, заполнив произвольными числами. Создать шапку, в которой будут следующие заголовки: *Массив*, *Отрицательные элементы*, *Количество отрицательных элементов*.

Добавить управляющие кнопки на лист (рисунок 52).

Примечание: Добавить управляющую кнопку: на вкладке **Разработчик** в группе **Элементы управления** в режиме **Конструктора**, вставить элемент **Кнопка** из группы **ActiveX**.

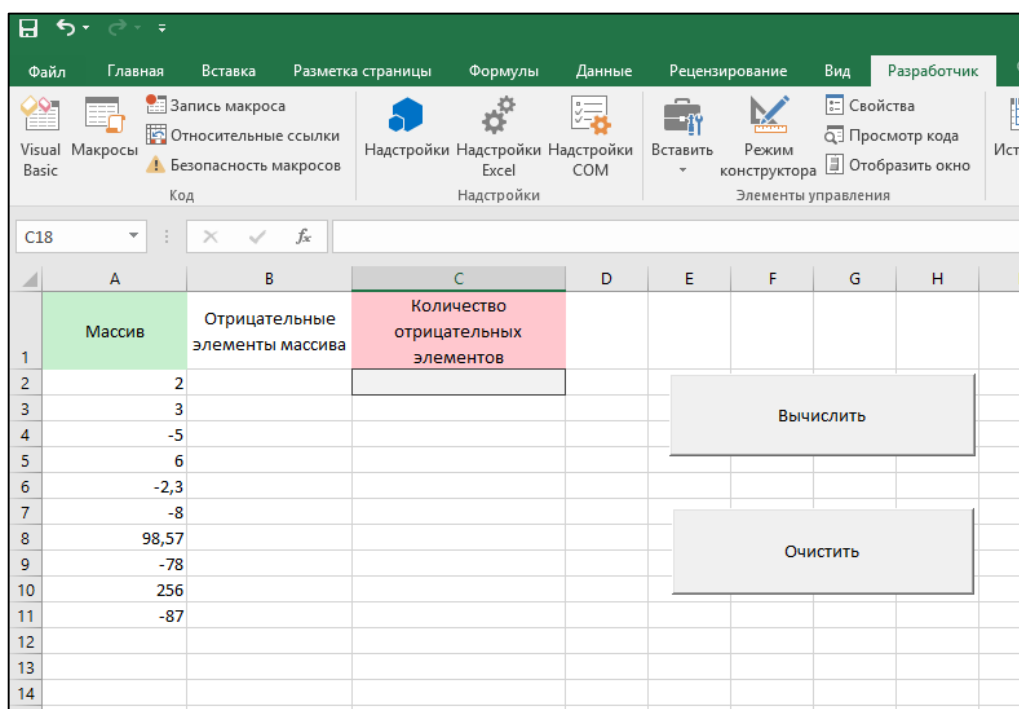


Рисунок 52 – Пользовательский интерфейс примера 1

Свойства объектов *Кнопка* представлены в таблице.

Таблица 18 – Свойства объектов интерфейса к примеру 1

Объект	Свойство
CommandButton1	Name = calc
	Caption = Вычислить
CommandButton2	Name = clean
	Caption = Очистить

В режиме *Конструктор* выполнить двойной щелчок мышью по кнопке *Вычислить* и в открывшемся окне записать код программы (рисунок).

```
clean

Private Sub calc_Click()
Dim x(1 To 10) As Integer, I, k, n As Integer
k = 0
For I = 1 To 10
x(I) = Cells(I + 1, "A")
If x(I) < 0 Then k = k + 1: Range("B" & I) = x(I)
Next I
Range("C2") = k
End Sub
```

Рисунок 53 – окно кода кнопки *Вычислить*

Код кнопки *Очистить* (рисунок 54):

```
Private Sub clean_Click()
For I = 1 To 10
Worksheets("Лист1").Range("A" & I + 1).ClearContents
Worksheets("Лист1").Range("B" & I + 1).ClearContents
Worksheets("Лист1").Range("C2").ClearContents
Next I
End Sub
```

Рисунок 54 – Код кнопки *Очистить*

Результат работы программы (рисунок 55):

	A	B	C	D	E	F	G	H
	Массив	Отрицательные элементы массива	Количество отрицательных элементов					
1								
2	2		5					
3	3							
4	-5	-5						
5	6							
6	-2,3	-2						
7	-8	-8						
8	98,57							
9	-78	-78						
10	256							
11	-87	-87						
12								

Рисунок 55 – Результат работы программы

Индивидуальное задание:

Используя редактор VBA MS Excel, сформировать элементы одномерного массива в соответствии с вариантом задания (таблица 19) и вывести результат на лист MS Excel. Создать все нужные заголовки. Результаты вычислений отобразить на листе MS Excel. Создать управляющие кнопки **Вычислить**, **Очистить** и, если требуется по заданию, кнопку **Сортировка**.

Таблица 19 – Варианты индивидуального задания лабораторной работы 5

№ вар.	Задание
1	2
1	Сформировать элементы массива по правилу $x_i = 2\cos(2i) - 1$. Определить максимальный элемент массива и его порядковый номер. Отсортировать элементы исходного массива по возрастанию, вывести отсортированный массив в отдельный столбец.
2	Сформировать элементы массива по правилу $x_i = 3\sin(2i)$. Посчитать сумму положительных элементов массива. Вывести положительные элементы в отдельный столбец.
3	Сформировать элементы массива, организовав ввод данных с помощью функции <i>Inputbox</i> . Отсортировать элементы исходного массива по возрастанию. Из полученного массива положительные элементы заменить их квадратным корнем, а отрицательные возвести в квадрат, вывести измененный массив в отдельный столбец.
4	Сформировать элементы массива, организовав ввод данных с помощью функции <i>Inputbox</i> . Переписать в массив Z элементы $-5 < x_i < 5$ и посчитать их количество.

Продолжение таблицы 19

5	Сформировать элементы массива по правилу $x_i = (2 + i) \ln(i + 1)$. Найти среднее арифметическое элементов массива. Элементы, имеющие значение больше среднеарифметического, вывести в отдельный столбец.
6	Сформировать элементы массива по правилу $x_i = 3 + (2 + i)(i - 1)$. Отсортировать элементы исходного массива по убыванию. Определить количество элементов массива кратных 3, вывести эти элементы в отдельный столбец.
7	Сформировать элементы массива, организовав ввод данных с помощью функции <i>Inputbox</i> . Посчитать сумму элементов, стоящих в массиве на четных местах, вывести эти элементы в отдельный столбец.
8	Сформировать элементы массива, организовав ввод данных с помощью функции <i>Inputbox</i> . Отсортировать элементы исходного массива по убыванию. Заменить все положительные элементы массива на 1, вывести измененный массив в отдельный столбец.
9	Сформировать элементы массива по правилу $x_i = 2 + (i - 1)^2$. Найти сумму максимального и минимального элементов. Отсортировать элементы исходного массива по возрастанию, вывести отсортированный массив в отдельный столбец.
10	Сформировать элементы массива по правилу $x_i = 5 \cos(i^2 - 1)$. Отсортировать элементы исходного массива по убыванию. Заменить все отрицательные элементы массива на 0 и вывести измененный массив в отдельный столбец.
11	Сформировать элементы массива, организовав ввод данных с помощью функции <i>Inputbox</i> . Определить минимальный элемент массива и его порядковый номер. Отсортировать элементы исходного массива по убыванию, вывести отсортированный массив в отдельный столбец.
12	Сформировать элементы массива, организовав ввод данных с помощью функции <i>Inputbox</i> . Отсортировать элементы исходного массива по возрастанию, вывести отсортированный массив в отдельный столбец. Посчитать сумму отрицательных элементов и сумму положительных элементов массива.
13	Сформировать элементы массива по правилу $x_i = 2 + (i + 2)^2$. Найти сумму элементов, меньших среднего значения. Вывести эти элементы в отдельный столбец.
14	Сформировать элементы массива по правилу $x_i = 5 - (i - 3)^2$. Определить количество четных элементов массива, переписать четные элементы в отдельный массив.
15	Сформировать элементы массива, организовав ввод данных с помощью функции <i>Inputbox</i> . Посчитать сумму элементов, стоящих в массиве на нечетных местах. Вывести элементы, стоящие на нечетных местах в отдельный столбец.

6.3 Вопросы для самопроверки

1. Какие массивы называются одномерными?
2. Как выполняется описание одномерного массива?
3. Как выполняется нумерация элементов одномерного массива?
4. Как обратиться к элементу одномерного массива?
5. Какие алгоритмические конструкции используются для задания одномерного массива?
6. Какие функции можно использовать для обращения к ячейкам листа MS Excel в VBA?
7. Как можно использовать функцию Range при работе с одномерным массивом?
8. Как можно использовать функцию Cells при работе с одномерным массивом?
9. Что такое сортировка массива и как она выполняется?

7 Двумерные массивы в среде VBA

7.1 Общие сведения о двумерных массивах

Двумерный массив - структура данных, хранящая прямоугольную матрицу. В матрице каждый элемент определяется номером строки и номером столбца, на пересечении которых он расположен. Например, адрес каждой ячейки листа MS Excel состоит из двух чисел, одно из которых (номер строки) является первым индексом, а второе (номер столбца) - вторым индексом массива.

Общий вид описания двумерного массива в редакторе VBA:

Dim Имя_Массива (X, Y) As Тип_Элементов,

где X и Y – целые числа, а нумерация элементов в этом случае начинается с нуля.

Примеры описания двумерных массивов:

Dim Mass (5,7) **As Integer** - двумерный статический массив целых чисел, включающий 48 элементов.

Dim Mass (2 To 8, 1 To 6) **As Integer** - двумерный статический массив целых чисел, включающий $7*6=42$ элемента.

Обращение к элементу массива происходит путем указания имени массива и двух индексов, заключенных в круглые скобки, например, Z(5, 7). При этом первый индекс указывает на номер строки, а второй на номер столбца.

Для записи элементов двумерного массива в ячейки MS Excel, а также считывания данных из ячеек можно использовать свойства *Range* и *Cells* аналогично тому, как данные свойства используются при обработки одномерных массивов.

Пример 1. Организовать ввод целочисленных значений элементов двумерного массива с помощью оператора *Inputbox* и вывод этих элементов в виде таблицы на рабочий лист MS Excel.

```
Private Sub Primer1_Click()  
Dim Z(1 To 5, 1 To 5) As Integer  
Dim i, j As Integer  
For i = 1 To 5  
For j = 1 To 5  
Z(i, j) = Val(InputBox("введите элемент " & i & j))  
Cells(i, j) = Z(i, j) 'вывод элементов массива на лист  
Next j  
Next i  
End Sub
```

Пример 2. Определить наибольший элемент матрицы, номер строки и столбца, на пересечении которых он расположен. Элементы матрицы ввести непосредственно в ячейки листа MS Excel.

```
Private Sub Primer2_Click()  
Dim Z(1 To 3, 1 To 3) As Integer  
Dim i, j, max, stroka, stolb As Integer  
Z(1, 1) = Cells(1, 1)
```

```

max = Z(1, 1)
For i = 1 To 3
For j = 1 To 3
Z(i, j) = Cells(i, j)
If Z(i, j) > max Then
max = Z(i, j)
stroka = i
stolb = j
End If
Next j
Next i
Range("A5") = max 'вывод максимального значения в ячейку A5
Range("A6") = stroka 'вывод найденного номера строки в ячейку A6
Range("A7") = stolb 'вывод найденного номера столбца в ячейку A7
End Sub

```

7.2 Лабораторная работа № 6 «Обработка двумерных массивов в среде VBA»

Пример 1. Строительная фирма арендует автомобильную технику, необходимую для использования на трех строительных участках. Известны данные о количестве часов аренды каждой единицы техники.

Необходимо разработать пользовательский интерфейс и составить программу в редакторе VB MS Excel, с помощью которой выполнить:

- 1) ввод исходных данных двумерного массива и их вывод в виде таблицы на рабочий лист;
- 2) определить:
 - аренду в часах каждого вида технического средства на трех участках;

- вид техники, которая по общим часам аренды на трех участках уступает другим видам техники;
- среднее количество часов аренды техники на 1 участке
- количество технических средств, используемых на 1 участке, для которых количество часов аренды не превышало найденного среднего показателя, и виды таких технических средств.

4) вывод полученных результатов на тот же рабочий лист.

Методика выполнения:

На первом рабочем листе Excel сформировать таблицу и ввести названия пяти видов автомобильных технических средств (рисунок 56).

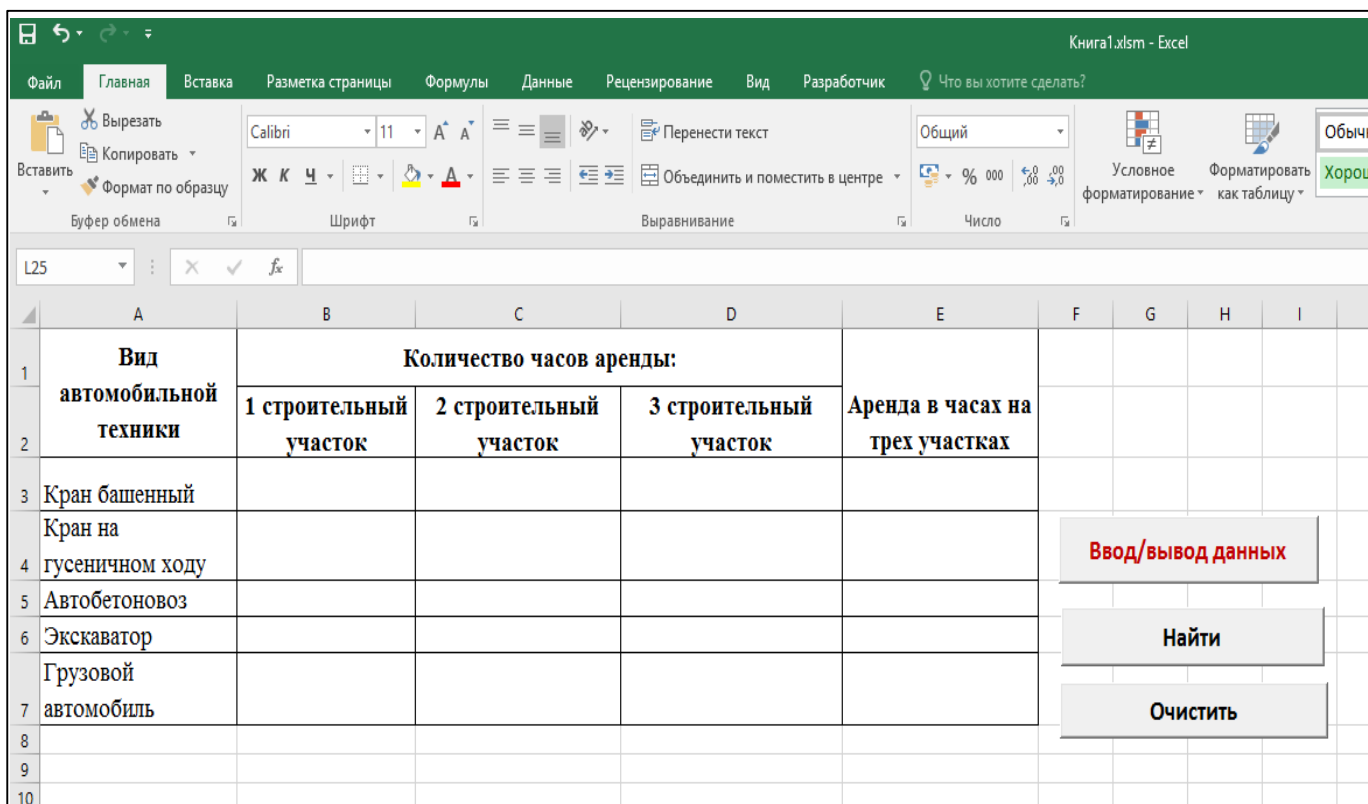


Рисунок 56 – Пользовательский интерфейс примера 1

Ниже таблицы разместить три кнопки с помощью соответствующего инструмента панели *Элементы управления*. Свойства размещенных на первом листе кнопок представлены в таблице 20.

Таблица 20 – Свойства объектов интерфейса примера 1

Объект	Свойства
CommandButton1	Name = inputOutput Caption = Ввод/вывод данных
CommandButton2	Name = calc Caption = Найти
CommandButton3	Name = clean Caption = Очистить

Значения количества часов проката будем вводить с помощью функции *InputBox* и присваивать соответствующим элементам двумерного массива.

Для обработки события нажатия кнопки **Ввод/вывод** ввести программный код:

Dim Chas(1 To 10, 1 To 10) As Integer 'описание массива (количество часов аренды)

Dim Tehn, Uchas, i, j As Integer

Private Sub InputOutput_Click()

Tehn = Val(InputBox("Введите количество единиц техники", "Ввод данных"))

Uchas = Val(InputBox("Введите число строительных участков", "Ввод данных"))

For i = 1 To Tehn

For j = 1 To Uchas

Chas(i, j) = Val(InputBox("Введите количество часов аренды " & Cells(i + 2, 1) & " " & Cells(2, j + 1), "Ввод данных"))

Cells(i + 2, j + 1) = Chas(i, j) 'вывод элементов массива на лист

Next j

Next i

End Sub

Для обработки события нажатия кнопки **Найти** ввести программный код:

Private Sub calc_Click()

Dim s, min, minTehn, k As Integer

Dim marka As String

Tehn = 5 'количество технических средств

Uchas = 3 'количество участков

For i = 1 To Tehn

For j = 1 To Uchas

Chas(i, j) = Cells(i + 2, j + 1)

Next j

Next i

min = 45000 'число произвольное, заведомо большее любого значения элементов

массива

For i = 1 To Tehn

s = 0

For j = 1 To Uchas

s = s + Chas(i, j) 'нахождение суммы часов проката каждой единицы техники

Next j

Cells(i + 2, "E") = s

If s < min Then min = s: minTehn = i 'поиск минимальной из этих сумм

Next i

MsgBox "Вид техники, уступающей остальным автомобильным средствам по часам аренды на 3 участках, - " & Cells(minTehn + 2, 1)

s = 0

For i = 1 To 5

s = s + Chas(i, 1)

Next i

MsgBox "Среднее количество часов аренды одного технического средства на 1 участке составляет: " & CStr(Round(s / Tehn, 2))

k = 0

marka = ""

For i = 1 To Tehn

If Chas(i, 1) <= s / Tehn Then

k = k + 1

marka = marka + Cells(i + 2, 1) + " "

End If

Next i

MsgBox "Количество технических средств 1 участка, у которых количество часов аренды не превышает среднее, " & Str(k) & " Техника 1 участка (кол-во часов не превышает среднее) " & marka

Cells(Tehn + 3, 1) = "Вид техники, имеющей минимальное кол-во часов аренды"

'вывод результатов на лист

Cells(Tehn + 3, 2) = Cells(minTehn + 2, 1)

Cells(Tehn + 4, 1) = "Среднее кол-во часов аренды на 1 участке"

Cells(Tehn + 4, 2) = s / Tehn

Cells(Tehn + 5, 1) = "Количество технических средств 1 участка (кол-во часов не превышает среднее)"

Cells(Tehn + 5, 2) = k

Cells(Tehn + 6, 1) = "Техника 1 участка (кол-во часов не превышает среднее)"

Cells(Tehn + 6, 2) = marka

End Sub

Ввести программный код для кнопки **«Очистить»** как показано на рисунке 57.

```
Private Sub clean_Click()  
Tehn = 5  
Uchas = 3  
For i = 1 To Tehn + 4  
For j = 1 To Uchas + 1  
Cells(i + 2, j + 1).ClearContents  
Next j  
Cells(i + Tehn + 2, 1).ClearContents  
Next i  
End Sub
```

Рисунок 57 – Код кнопки *Очистить*

Результат выполнения программы представлен на рисунке 58:

The screenshot shows an Excel spreadsheet with the following data:

1	Вид автомобильной техники	Количество часов аренды:			Аренда в часах на трех участках
		1 строительный участок	2 строительный участок	3 строительный участок	
3	Кран башенный	42	54	32	128
4	Кран на гусеничном ходу	63	74	52	189
5	Автобетоновоз	66	35	45	146
6	Экскаватор	15	52	37	104
7	Грузовой автомобиль	66	25	35	126
8	Вид техники, имеющей минимальное кол-во часов аренды	Экскаватор			
9	Среднее кол-во часов аренды на 1 участке	50,4			
10	Количество технических средств 1 участка (кол-во часов не превышает среднее)	2			
11	Техника 1 участка (кол-во часов не превышает среднее)	Кран башенный Экскаватор			

Buttons on the right side of the spreadsheet:

- Ввод/вывод данных
- Найти
- Очистить

Рисунок 58 – Результат выполнения программы примера 1

Индивидуальное задание:

В соответствии с вариантом задания (таблица 21) разработать пользовательский интерфейс и составить программу в редакторе VB MS Excel, с помощью которой осуществляется:

- 1) ввод исходных данных двумерного массива (ввод наименований в ячейки первого столбца таблицы осуществляется вручную, численные данные массива вводятся через функцию InputBox) и их вывод в виде таблицы на рабочий лист;
- 2) обработка массива (нахождение суммы, количества элементов и т. п. в зависимости от варианта задания) и вывод ее результатов на тот же рабочий лист.

Таблица 21 – Варианты индивидуального задания лабораторной работы 7

№В	Задание												
1	2												
1	<p>Известны данные о доходе строительного магазина от продажи каждого из 6 наименований стройматериалов за осень:</p> <table border="1" data-bbox="260 454 1503 586"> <thead> <tr> <th data-bbox="260 454 571 499">Стройматериалы</th> <th colspan="3" data-bbox="571 454 1503 499">Доход (руб)</th> </tr> <tr> <td data-bbox="260 499 571 544"></td> <th data-bbox="571 499 882 544">сентябрь</th> <th data-bbox="882 499 1193 544">октябрь</th> <th data-bbox="1193 499 1503 544">ноябрь</th> </tr> </thead> <tbody> <tr> <td data-bbox="260 544 571 586">...</td> <td data-bbox="571 544 882 586">...</td> <td data-bbox="882 544 1193 586">...</td> <td data-bbox="1193 544 1503 586">...</td> </tr> </tbody> </table> <p>Определить:</p> <ul style="list-style-type: none"> – доход за осенний период, полученный от продажи каждого наименования стройматериалов; – название стройматериала, от продажи которого получен максимальный доход за три месяца и долю (в %) этого дохода в общем доходе, полученном за осень. 	Стройматериалы	Доход (руб)				сентябрь	октябрь	ноябрь
Стройматериалы	Доход (руб)												
	сентябрь	октябрь	ноябрь										
...										
2	<p>Известны данные о количестве часов проката на каждом из 5 строительных участков трех видов спецтехники:</p> <table border="1" data-bbox="260 938 1503 1115"> <thead> <tr> <th data-bbox="260 938 571 983">Название участка</th> <th colspan="3" data-bbox="571 938 1503 983">Количество часов проката</th> </tr> <tr> <td data-bbox="260 983 571 1028"></td> <th data-bbox="571 983 882 1028">Экскаваторы</th> <th data-bbox="882 983 1193 1028">Бульдозеры</th> <th data-bbox="1193 983 1503 1028">Автобетоновоз</th> </tr> </thead> <tbody> <tr> <td data-bbox="260 1028 571 1115">...</td> <td data-bbox="571 1028 882 1115">...</td> <td data-bbox="882 1028 1193 1115">...</td> <td data-bbox="1193 1028 1503 1115">...</td> </tr> </tbody> </table> <p>Определить:</p> <ul style="list-style-type: none"> – общее количество часов проката по каждому виду спецтехники; – название участка, у которого наибольшее количество часов проката из указанных в таблице и название соответствующей спецтехники. 	Название участка	Количество часов проката				Экскаваторы	Бульдозеры	Автобетоновоз
Название участка	Количество часов проката												
	Экскаваторы	Бульдозеры	Автобетоновоз										
...										
3	<p>Известны данные о зарплате 7 сотрудников строительной фирмы за каждый месяц второго квартала:</p> <table border="1" data-bbox="260 1377 1503 1509"> <thead> <tr> <th data-bbox="260 1377 604 1422">ФИО сотрудника</th> <th colspan="3" data-bbox="604 1377 1503 1422">Зарплата</th> </tr> <tr> <td data-bbox="260 1422 604 1467"></td> <th data-bbox="604 1422 890 1467">апрель</th> <th data-bbox="890 1422 1147 1467">май</th> <th data-bbox="1147 1422 1503 1467">июнь</th> </tr> </thead> <tbody> <tr> <td data-bbox="260 1467 604 1509">...</td> <td data-bbox="604 1467 890 1509">...</td> <td data-bbox="890 1467 1147 1509">...</td> <td data-bbox="1147 1467 1503 1509">...</td> </tr> </tbody> </table> <p>Определить:</p> <ul style="list-style-type: none"> – общую зарплату всех сотрудников за каждый месяц; – среднюю зарплату сотрудников фирмы, полученную за первый месяц квартала, число и фамилии сотрудников, у которых зарплата за первый месяц квартала превысила найденный средний показатель 	ФИО сотрудника	Зарплата				апрель	май	июнь
ФИО сотрудника	Зарплата												
	апрель	май	июнь										
...										

Продолжение таблицы 21

1	2																									
4	<p>Известны данные о количестве жильцов, проживающих в каждой из четырех квартир каждого этажа 6-этажного дома:</p> <table border="1" data-bbox="261 389 1495 609"> <thead> <tr> <th data-bbox="261 389 558 474">Номер этажа</th> <th colspan="4" data-bbox="558 389 1495 434">Число жителей</th> </tr> <tr> <th data-bbox="261 434 558 474"></th> <th data-bbox="558 434 794 474">1</th> <th data-bbox="794 434 1031 474">2</th> <th data-bbox="1031 434 1267 474">3</th> <th data-bbox="1267 434 1495 474">4</th> </tr> </thead> <tbody> <tr> <td data-bbox="261 474 558 519">1</td> <td data-bbox="558 474 794 519">...</td> <td data-bbox="794 474 1031 519">...</td> <td data-bbox="1031 474 1267 519">...</td> <td data-bbox="1267 474 1495 519">...</td> </tr> <tr> <td data-bbox="261 519 558 564">2</td> <td data-bbox="558 519 794 564">...</td> <td data-bbox="794 519 1031 564">...</td> <td data-bbox="1031 519 1267 564">...</td> <td data-bbox="1267 519 1495 564">...</td> </tr> <tr> <td data-bbox="261 564 558 609">...</td> <td data-bbox="558 564 794 609">...</td> <td data-bbox="794 564 1031 609">...</td> <td data-bbox="1031 564 1267 609">...</td> <td data-bbox="1267 564 1495 609">...</td> </tr> </tbody> </table> <p>– Определить: – Общее число жителей на каждом этаже; – Этаж, на котором проживает максимальное количество жителей – Среднее количество жителей в доме.</p>	Номер этажа	Число жителей					1	2	3	4	1	2
Номер этажа	Число жителей																									
	1	2	3	4																						
1																						
2																						
...																						
5	<p>Известны данные о доходах каждого из 6 строительных магазинов за последние 3 года:</p> <table border="1" data-bbox="261 882 1495 1012"> <thead> <tr> <th data-bbox="261 882 571 967" rowspan="2">Название магазина</th> <th colspan="3" data-bbox="571 882 1495 927">Доход</th> </tr> <tr> <th data-bbox="571 927 880 967">2018</th> <th data-bbox="880 927 1190 967">2019</th> <th data-bbox="1190 927 1495 967">2020</th> </tr> </thead> <tbody> <tr> <td data-bbox="261 967 571 1012">...</td> <td data-bbox="571 967 880 1012">...</td> <td data-bbox="880 967 1190 1012">...</td> <td data-bbox="1190 967 1495 1012">...</td> </tr> </tbody> </table> <p>Определить: – название магазина, который получил максимальный доход из указанного в таблице, и год, за который данный магазин его получил; – для каждого магазина – год, за который этот магазин получил минимальный доход.</p>	Название магазина	Доход			2018	2019	2020														
Название магазина	Доход																									
	2018	2019	2020																							
...																							
6	<p>Известны данные о расходах 5 строительных участков на стройматериалы за 3 квартала текущего года:</p> <table border="1" data-bbox="261 1323 1495 1453"> <thead> <tr> <th data-bbox="261 1323 571 1364" rowspan="2">Название участка</th> <th colspan="3" data-bbox="571 1323 1495 1364">Расходы на стройматериалы</th> </tr> <tr> <th data-bbox="571 1364 880 1408">1-й квартал</th> <th data-bbox="880 1364 1190 1408">2-й квартал</th> <th data-bbox="1190 1364 1495 1408">3-й квартал</th> </tr> </thead> <tbody> <tr> <td data-bbox="261 1408 571 1453">...</td> <td data-bbox="571 1408 880 1453">...</td> <td data-bbox="880 1408 1190 1453">...</td> <td data-bbox="1190 1408 1495 1453">...</td> </tr> </tbody> </table> <p>Определить: – общую сумму расходов каждого участка за три квартала; – название участка, у которого наименьшие расходы за три квартала.</p>	Название участка	Расходы на стройматериалы			1-й квартал	2-й квартал	3-й квартал														
Название участка	Расходы на стройматериалы																									
	1-й квартал	2-й квартал	3-й квартал																							
...																							
7	<p>Известны данные о количестве дней аренды каждого из 5 видов строительной техники строительными участками трех жилых комплексов:</p> <table border="1" data-bbox="261 1677 1495 1852"> <thead> <tr> <th data-bbox="261 1677 571 1718" rowspan="2">Название строительной техники</th> <th colspan="3" data-bbox="571 1677 1495 1718">Количество дней аренды:</th> </tr> <tr> <th data-bbox="571 1718 880 1803">Жилой комплекс «Акварель»</th> <th data-bbox="880 1718 1190 1803">Жилой комплекс «Времена года»</th> <th data-bbox="1190 1718 1495 1803">Жилой комплекс «Дубки»</th> </tr> </thead> <tbody> <tr> <td data-bbox="261 1803 571 1852">...</td> <td data-bbox="571 1803 880 1852">...</td> <td data-bbox="880 1803 1190 1852">...</td> <td data-bbox="1190 1803 1495 1852">...</td> </tr> </tbody> </table> <p>Определить: – общее количество дней аренды строительной техники на всех участках; – участок, имеющий наибольшее количество дней аренды.</p>	Название строительной техники	Количество дней аренды:			Жилой комплекс «Акварель»	Жилой комплекс «Времена года»	Жилой комплекс «Дубки»														
Название строительной техники	Количество дней аренды:																									
	Жилой комплекс «Акварель»	Жилой комплекс «Времена года»	Жилой комплекс «Дубки»																							
...																							

Продолжение таблицы 21

1	2				
8	Известны данные о количестве студентов, обучающихся в каждой из 5 групп на 4 курсах строительного факультета:				
	Курс	Количество студентов			
	Стр(ба)1	Стр(ба)2	Стр(ба)3	Стр(ба)4	Стр(ба)5
...		
<p>Определить:</p> <ul style="list-style-type: none"> – общее количество студентов, обучающихся в 5 группах на каждом курсе; – курс, на котором обучается меньше всего студентов и их количество, и сколько процентов составляет это количество относительно общего числа студентов строительного факультета. 					
9	Известны данные о доходе строительного магазина от продажи каждого из 5 наименований стройматериала за летний период:				
	Наименование стройматериала	Доход			
		июнь	июль	август	
<p>Определить:</p> <ul style="list-style-type: none"> – общий доход магазина, полученный от продажи стройматериалов за лето; – средний размер дохода, полученного магазином за август от продажи стройматериалов; – количество видов стройматериала доход от продажи которых в августе превысил найденный средний показатель, и название этих видов стройматериала 					
10	Известны данные о доходах строительной организации от продажи квартир в каждом из 5 жилых комплексов за 3 квартала текущего года:				
	Название жилого комплекса	Доход (млн. руб.)			
		1 квартал	2 квартал	3 квартал	
<p>Определить:</p> <ul style="list-style-type: none"> – квартал, в котором организация получила минимальный общий доход; – средний размер дохода, полученного организацией за третий квартал; – количество и название жилых комплексов, принесших за данный квартал доход, не превышающий найденного среднего показателя. 					

Продолжение таблицы 21

1	2																							
11	<p>Известны данные о доходах от продажи 5 наименований отделочных материалов тремя филиалами строительного магазина:</p> <table border="1" data-bbox="261 389 1503 564"> <thead> <tr> <th data-bbox="261 389 571 519" rowspan="2">Наименование материала</th> <th colspan="3" data-bbox="571 389 1503 434">Доход (руб)</th> </tr> <tr> <th data-bbox="571 434 880 519">Восточный филиал</th> <th data-bbox="880 434 1190 519">Центральный филиал</th> <th data-bbox="1190 434 1503 519">Западный филиал</th> </tr> </thead> <tbody> <tr> <td data-bbox="261 519 571 564"></td> <td data-bbox="571 519 880 564"></td> <td data-bbox="880 519 1190 564"></td> <td data-bbox="1190 519 1503 564"></td> </tr> </tbody> </table> <p>Определить:</p> <ul style="list-style-type: none"> – доход, полученный магазином от продажи каждого вида отделочных материалов тремя филиалами; – наименование отделочного материала, от продажи которого был получен наибольший доход из указанного в таблице, и каким филиалом он был получен. 	Наименование материала	Доход (руб)			Восточный филиал	Центральный филиал	Западный филиал																
Наименование материала	Доход (руб)																							
	Восточный филиал	Центральный филиал	Западный филиал																					
12	<p>Известны данные о доходе строительной фирмы от продажи каждой из 4 видов квартир за 1 квартал:</p> <table border="1" data-bbox="261 913 1503 1137"> <thead> <tr> <th data-bbox="261 913 571 1003" rowspan="2">Вид квартиры</th> <th colspan="3" data-bbox="571 913 1503 958">Доход (млн. руб)</th> </tr> <tr> <th data-bbox="571 958 880 1003">январь</th> <th data-bbox="880 958 1190 1003">февраль</th> <th data-bbox="1190 958 1503 1003">март</th> </tr> </thead> <tbody> <tr> <td data-bbox="261 1003 571 1048">Студия</td> <td data-bbox="571 1003 880 1048"></td> <td data-bbox="880 1003 1190 1048"></td> <td data-bbox="1190 1003 1503 1048"></td> </tr> <tr> <td data-bbox="261 1048 571 1093">Однокомнатная</td> <td data-bbox="571 1048 880 1093"></td> <td data-bbox="880 1048 1190 1093"></td> <td data-bbox="1190 1048 1503 1093"></td> </tr> <tr> <td data-bbox="261 1093 571 1137"></td> <td data-bbox="571 1093 880 1137"></td> <td data-bbox="880 1093 1190 1137"></td> <td data-bbox="1190 1093 1503 1137"></td> </tr> </tbody> </table> <p>Определить:</p> <ul style="list-style-type: none"> – общий доход, полученный за каждый месяц; – месяц, в котором был получен минимальный доход от продажи квартир и сколько процентов составляет эта сумма относительно общего дохода от продажи квартир за весь период. 	Вид квартиры	Доход (млн. руб)			январь	февраль	март	Студия				Однокомнатная											
Вид квартиры	Доход (млн. руб)																							
	январь	февраль	март																					
Студия																								
Однокомнатная																								
13	<p>Известны данные о количестве книг, поступивших в каждый из 5 отделов университетской библиотеки за последние 3 года:</p> <table border="1" data-bbox="261 1442 1503 1711"> <thead> <tr> <th data-bbox="261 1442 571 1532" rowspan="2">Название отдела</th> <th colspan="3" data-bbox="571 1442 1503 1487">Количество книг</th> </tr> <tr> <th data-bbox="571 1487 880 1532">2018</th> <th data-bbox="880 1487 1190 1532">2019</th> <th data-bbox="1190 1487 1503 1532">2020</th> </tr> </thead> <tbody> <tr> <td data-bbox="261 1532 571 1576">экономический</td> <td data-bbox="571 1532 880 1576"></td> <td data-bbox="880 1532 1190 1576"></td> <td data-bbox="1190 1532 1503 1576"></td> </tr> <tr> <td data-bbox="261 1576 571 1621">юридический</td> <td data-bbox="571 1576 880 1621"></td> <td data-bbox="880 1576 1190 1621"></td> <td data-bbox="1190 1576 1503 1621"></td> </tr> <tr> <td data-bbox="261 1621 571 1666">научный</td> <td data-bbox="571 1621 880 1666"></td> <td data-bbox="880 1621 1190 1666"></td> <td data-bbox="1190 1621 1503 1666"></td> </tr> <tr> <td data-bbox="261 1666 571 1711"></td> <td data-bbox="571 1666 880 1711"></td> <td data-bbox="880 1666 1190 1711"></td> <td data-bbox="1190 1666 1503 1711"></td> </tr> </tbody> </table> <p>Определить:</p> <ul style="list-style-type: none"> – общее количество книг, поступивших в каждый из отделов библиотеки за 3 года; – название отдела и соответствующий год, в который поступило наибольшее количество книг в течение каждого из 3 годов. 	Название отдела	Количество книг			2018	2019	2020	экономический				юридический				научный							
Название отдела	Количество книг																							
	2018	2019	2020																					
экономический																								
юридический																								
научный																								

Продолжение таблицы 21

1	2			
14	Известны данные о доходах каждой из 5 строительных организаций за летний период:			
	Название организации	Доход (руб)		
		Июнь	Июль	Август
Определить: – общий доход, полученный каждой строительной организацией за лето; – название организации, которая получила наибольший общий доход за лето и долю (в %) этого дохода в суммарном доходе всех организаций.				
15	Известны данные об итогах сдачи студентами 1 курса строительного факультета экзаменов по 5 дисциплинам:			
	Название дисциплины	Количество студентов, сдавших на оценку		
		отлично	хорошо	удовлетворительно
Определить: – название дисциплины, по которой оценку неудовлетворительно имеют наименьшее количество студентов; – для каждой дисциплины – самую распространенную оценку, которую получили студенты.				

7.3 Вопросы для самопроверки

1. Какие массивы называются двумерными?
2. Чем отличается двумерный массив от одномерного?
3. Как задать двумерный массив?
4. Что такое индексы массива, для чего они используются?
5. Как выполняется нумерация элементов двумерного массива?
6. Как обратиться к элементу двумерного массива?
7. Какие алгоритмические конструкции используются для задания двумерного массива? Какие операторы языка программирования используются для этого?
8. Какие функции можно использовать для задания и считывания элементов двумерного массива из ячеек листа MS Excel в VBA?

Список использованных источников

1. Волченков Н.Г. Основы программирования на языке Visual Basic для офисных приложений: Учебное пособие [Электронный ресурс]. – М.: НИЯУ МИФИ, 2018. – 166 с.
2. Бегункова, Н. О Программирование на VBA в MS Excel: методические указания к выполнению лабораторных работ № 1–4 по информатике для студентов дневной формы обучения / сост. Н. О. Бегункова, Н. Д. Белова, Т. А. Бочарова. – Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2010. – 44 с
3. Демидов, Л. Н. Основы информатики [Текст]: учебник для студентов, обучающихся по направлению подготовки "Информатика и вычислительная техника" / Л. Н. Демидов [и др.]. - Москва: КНОРУС, 2018. - 392 с.: ил. - (Бакалавриат). - На обл. и тит. л.: Электронно-библиотечная система ВООК.ru. - Библиогр.: с. 390-391. - ISBN 978-5-406-06333-0.
4. Кузьменко, В. Г. VBA. Эффективное использование / В. Г. Кузьменко. – М.: Бином-Пресс., 2012. – 624 с.: ил.
5. Макарова, Н. В. Информатика [Текст] : учеб. для вузов / Н. В. Макарова, В. Б. Волков. - СПб. : Питер, 2012. - 574 с.: ил. - (Учебник для вузов. Стандарт третьего поколения). - Библиогр. в конце гл. - ISBN 978-5-496-00001-7.
6. Молоков, К. А. Основы информатики и программирование под Windows [Текст] : учебное пособие / К. А. Молоков; Дальневост. федер. ун-т. - Москва : Проспект, 2018. - 224 с. : ил. - Слов. терминов: с. 205-219. - Библиогр.: с. 220-221. - ISBN 978-5-392-28068-1.
7. Назаров С.В. Программирование на MS Visual Basic: Учебное пособие / Назаров С.В., Мельников П.П. – М.: Финансы и статистика, 2002. – 320 с.
8. Основные принципы и концепции программирования на языке VBA в Excel: Авторы: Серафима Белоусова, Ирина Бессонова / [Электронный ресурс] / ИНТУИТ национальный открытый университет. – Режим доступа: <https://www.intuit.ru/studies/courses/536/392/info>

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

10. Сергеева, Л. А. Создание первых проектов в среде Visual Basic for Applications MS Excel 2010: методические указания к выполнению лабораторных работ по программированию для студентов, изучающих раздел информатики «Алгоритмизация и программирование» /сост. Л. А. Сергеева. – Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2015. – 23 с.

11. Симонович, С.В. Информатика. Базовый курс [Текст]: учеб. Пособие для студентов вузов: для бакалавров и специалистов / под ред. С. В. Симоновича. - 3-е изд. - СПб.: Питер, 2012. - 638 с.: ил. - (Учебник для вузов)-ISBN 978-5-459-00439-7.

12. Токарева, М. А. Алгоритмизация и программирование. Лабораторный практикум по информатике для студентов экономических специальностей: учебное пособие/ М. А. Токарева, М. А. Корякина – Оренбург: ГОУ ОГУ, 2006. – 162с.

13. Черных, Т. А. Основы офисного программирования в MS Excel : учебное пособие / Т. А. Черных, Ю. В. Полищук, А. В. Максименко; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2013. – 121 с.

14. Эволюция и классификация языков программирования [Электронный ресурс] Инфопедия. – Режим доступа: <https://infopedia.su/14x8559.html>

15. Эволюция языков программирования [Электронный ресурс] Урок информатики. – Режим доступа: <http://informatics-lesson.ru/c/evolution-languages.php>