

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

Кафедра программного обеспечения вычислительной
техники и автоматизированных систем

Т.М. Зубкова

ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКОГО ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ПО ТЕХНОЛОГИИ WIMP

**Методические указания
к выполнению лабораторных работ по дисциплине
«Проектирование человеко-машинного интерфейса»**

Рекомендовано к изданию Редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Оренбургский государственный университет»

Оренбург
2011

УДК 681.3 (07)
ББК 32.973.26-018я73
3 91

Рецензент – профессор, доктор технических наук В.И. Чепасов

Зубкова, Т.М.

391

Проектирование графического пользовательского интерфейса по технологии WIMP: методические указания к выполнению лабораторных работ по дисциплине «Проектирование человеко-машинного интерфейса» /Т.М. Зубкова; Оренбургский гос. ун-т.- Оренбург: ОГУ, 2011. – 46 с.

Методические указания для выполнения лабораторных работ по дисциплине «Проектирование человеко-машинного интерфейса» предназначены для оказания методической помощи при выполнении лабораторных работ. Данная дисциплина входит в базовую часть профессионального цикла дисциплин бакалавров очной формы обучения по направлению 231000 –«Программная инженерия» с профилем подготовки «Разработка программно-информационных систем».

В методических указаниях изложены теоретические основы для проектирования диалогов в графическом пользовательском интерфейсе, цель и задачи лабораторных работ, а также примеры выполнения.

УДК 681.3 (07)
ББК 32.973.26-018я73

© Зубкова Т.М., 2011
© ОГУ, 2011

Содержание

Введение	4
1 Цель и задачи лабораторных работ.....	5
2 Краткие теоретические аспекты для выполнения лабораторных работ.....	6
2.1 Основные понятия и определения.....	6
2.2 Рекомендации по проектированию диалога пользователя с ПС.....	7
2.3 Основные элементы пользовательского интерфейса.....	10
2.4 Особенности графического интерфейса.....	17
2.5 Компоненты графического интерфейса.....	19
3. Примеры выполнения лабораторных работ.....	26
3.1 Изучение предметной области и оценка интерфейса конкурирующих систем – Лабораторная работа №1.....	26
3.2 Описание контекста использования ПС и бизнес-ролей пользователей – Лабораторная работа №1.....	28
3.3 Составление сценариев работы пользователей в виде схемы алгоритма, описывающей весь процесс использования ПС для выполнения конкретной задачи – Лабораторная работа №1.....	28
3.4 Разработка полной схемы экранов проектируемой ПС, определение типа и формы каждого диалога, а также синтаксиса и семантики используемых языков – Лабораторная работа №2.....	31
3.5 Проектирование меню MDI-интерфейса различного вида: основное, панели инструментов, контекстное и кнопочное. Использование директивной формы диалога (как альтернативной) с использованием соответствия основным командам комбинации клавиш – Лабораторная работа №2.....	34
3.6 Определение состава, внешнего вида и взаимного расположения визуальных интерфейсных компонентов в соответствии с эргономическими принципами проектирования ПС – Лабораторная работа №3.....	37
3.7 Разработка ПС, содержащего элементы управления в многооконных интерфейсах с использованием инструментальных сред программирования графического диалога. Программирование различных реакций ПС на действия пользователей – Лабораторная работа №3.....	38
4. Заключение.....	44
5. Список использованных источников.....	45

Введение

Методические указания предназначены для выполнения лабораторных работ по курсу «Проектирование человеко-машинного интерфейса».

Задачи, которые решаются при выполнении работ соответствуют рабочей программе по данному курсу:

- обучить тенденциям развития пользовательских интерфейсов;
- обучить устройству и режимам человеко-машинного диалога;
- обучить компьютерному представлению и визуализации информации;
- обучить описанию взаимодействия пользователя с компьютерной средой в заданной проблемной области.

Выполнение лабораторной работы позволяет формировать элементы следующих компетенций в соответствии с ФГОС ВПО и ООП ВПО по данному направлению подготовки (специальности):

а) общекультурных:

- владеть культурой мышления, способность к обобщению, анализу, восприятию информации, постановке цели и выбору путей ее достижения (ОК-1);
- готов к кооперации с коллегами, работе в коллективе (ОК-3);
- стремится к саморазвитию, повышению своей квалификации и мастерства (ОК-6);

б) профессиональных (ПК):

- понимание основных концепций, принципов, теорий и фактов, связанных с информатикой (ПК-1);
- способность к формализации в своей предметной области с учетом ограничений используемых методов исследования (ПК-2);
- способность создавать программные интерфейсы (ПК-14).

1 Цель и задачи лабораторных работ

Цель лабораторных работ: закрепление теоретических знаний и формирование практических умений и навыков использования инструментальных сред программирования для реализации диалогов в графическом пользовательском интерфейсе по технологии WIMP.

Постановка задачи

1. Изучить основные аспекты предметной области, произвести оценку интерфейса конкурирующих систем.

2. Описать контекст использования системы, бизнес-роли пользователей.

3. Составить сценарии работы пользователей в виде схемы алгоритма, описывающие весь процесс использования системы для выполнения конкретной задачи.

4. Разработать полную схему экранов системы, определить типы и формы каждого диалога, а также синтаксис и семантику используемых языков.

5. Разработать меню MDI-интерфейса различных видов: основное, панели инструментов, контекстное и кнопочное. Как альтернативу использовать директивную форму диалога, поставив в соответствие основным командам комбинации клавиш.

6. В соответствии с эргономическими принципами проектирования программных средств определить состав, внешний вид и взаимное расположение визуальных интерфейсных компонентов.

7. Используя инструментальные среды программирования графического диалога, разработать программное средство, содержащее элементы управления в многооконных интерфейсах. Распечатать макеты экранов с описанием функциональности каждого интерфейсного элемента. Запрограммировать различные реакции ПС на действия пользователя.

2. Краткие теоретические аспекты для выполнения лабораторных работ

2.1 Основные понятия и определения

Пользовательский интерфейс – это совокупность информационной модели проблемной области, средств и способов взаимодействия пользователя с информационной моделью, а также компонентов, обеспечивающих формирование информационной модели в процессе работы ПС.

Под информационной моделью понимается условное представление проблемной области, формируемое с помощью компьютерных (визуальных и звуковых) объектов, отражающих состав и взаимодействие реальных компонентов проблемной области. Средства и способы взаимодействия с информационной моделью определяются составом аппаратного и программного обеспечения, имеющегося в распоряжении пользователя, и от характера решаемой задачи.

Таким образом, эффективность работы пользователя определяется не только функциональными возможностями имеющихся в его распоряжении аппаратных и программных средств, но и доступностью для него этих возможностей. В свою очередь, полнота исполнения потенциальных возможностей имеющихся ресурсов зависит от качества пользовательского интерфейса.

Пользовательские интерфейсы большинства современных программных средств (ПС) строятся по технологии WIMP: W - Windows (окна), I - Icons (пиктограммы), M - Mouse (мышь), P - Pop-up (всплывающие или выпадающие меню). Основными элементами графических интерфейсов, таким образом, являются окна, пиктограммы, компоненты ввода-вывода и мышь, которую используют в качестве указующего устройства и устройства прямого манипулирования объектами на экране.

2.2 Рекомендации по проектированию диалога пользователя с ПС

Не смотря на то, что пользователь не воспринимает компьютер как полноценного собеседника, тем не менее, обмен информацией между пользователем и программным обеспечением по всем формальным признакам соответствует понятию «диалог» в общепринятом смысле. Основные правила, которые должны соблюдаться, чтобы диалог оказался конструктивным:

- участники диалога должны понимать язык друг друга;
- они не должны говорить одновременно;
- очередное высказывание должно учитывать как общий контекст диалога, так и последнюю информацию, полученную от собеседника.

Если собеседники обсуждают вопросы, относящиеся к какой-либо специальной области, они должны придерживаться единой терминологии, если же один из них пытается что-то объяснить другому, ему следует сначала пояснить основные термины и понятия.

Таким образом, при проектировании пользовательского интерфейса необходимо определить:

- применяемую терминологию;
- структуру диалога;
- возможный сценарий развития диалога;
- содержание управляющих сообщений и данных, которыми могут обмениваться человек и приложение (семантику сообщений);
- визуальные атрибуты отображаемой информации (синтаксис сообщений).

Выбор структуры диалога – это первый из этапов, который должен быть выполнен при разработке интерфейса. Существует несколько вариантов структуры диалога типа «вопрос-ответ», но каждая из них имеет свои особенности и наиболее удобна для определенного класса задач.

Структура диалога типа «вопрос-ответ» основана на аналогии с обычным интервью. Система берет на себя роль интервьюера и получает информа-

цию от пользователя в виде ответов на вопросы. Диалог в виде вопросов и ответов в достаточной степени обеспечивает поддержку пользователя, так как даже краткий наводящий вопрос при разумном построении может быть самопоясняющим.

Диалог на основе меню является наиболее популярным вариантом организации запросов на ввод данных во время диалога, управляемого компьютером.

Существует несколько основных форматов представления меню на экране:

- список объектов, выбираемых прямым указанием, либо указанием номера;
- меню в виде строки данных;
- меню в виде блока данных;
- меню в виде пиктограмм.

Меню можно применять для ввода, как управляющих сообщений, так и данных.

Диалог на основе экранных форм допускает обработку на одном шаге диалога нескольких ответов. На практике формы используются в основном там, где учет какой-либо деятельности требует ввода достаточно стандартного набора данных.

Одним из типов заполнения форм являются также многовариантные меню. В таких меню пользователю предоставляется список вариантов, и он не ограничен возможностью единственного выбора, может указать несколько вариантов.

Диалог на основе командного языка часто используется в операционных системах и располагается на другом конце спектра структур диалога по отношению к структуре типа «меню». Исторически это первая из реализованных структур диалога.

При такой организации диалога программная система не выводит ничего, кроме постоянной подсказки (приглашения на ввод команды), которая означает готовность системы к работе. Каждую команду вводят с новой строки и обычно

заканчивают нажатием клавиши «Ввод». Ответственность за правильность задаваемых команд ложится на пользователя.

Развитие диалога во времени можно рассматривать как последовательность переходов системы из одного состояния в другое. Очевидно, что ни одно из этих состояний не должно быть «тупиковым», т.е. пользователь должен иметь возможность перейти из любого текущего состояния диалога в требуемое (за один или несколько шагов). Для этого в ходе разработки интерфейса необходимо определить все возможные состояния диалога и пути перехода из одного состояния в другое. Другими словами, необходимо разработать сценарий диалога.

Целями разработки сценария диалога являются:

- выявление и устранение возможных тупиковых ситуаций в ходе развития диалога;
- выбор рациональных путей перехода из одного состояния диалога в другое (из текущего в требуемое);
- выявление неоднозначных ситуаций, требующих оказания дополнительной помощи пользователю.

Сложность разработки сценария определяется в основном двумя факторами: функциональными возможностями создаваемого приложения (т.е. числом и сложностью реализуемых функций обработки информации) и степенью неопределенности возможных действий пользователя.

В свою очередь, степень неопределенности действий пользователем зависит от выбранной структуры диалога. Наибольшей вариантностью обладает диалог на основе меню, наименьшей – диалог типа «вопрос – ответ», управляемый пользователем.

Сценарий диалога, можно упростить, снизив степень неопределенности действий пользователя. Возможными способами решения этой задачи являются:

- использование смешанной структуры диалога (применение меню с целью «ограничения свободы» пользователя там, где это возможно);

- применение входного контроля вводимой информации (команд и данных).

Дополнительные возможности по снижению неопределенности действий пользователя предоставляет объектно-ориентированный подход к разработке интерфейса, при котором для каждого объекта заранее устанавливается перечень свойств и допустимых операций. Наиболее эффективен такой подход при создании графического интерфейса.

Способ описания сценария диалога зависит от степени его сложности.

Независимо от способа описания сценария его основной структурной единицей является шаг диалога, соответствующий одному акту взаимодействия пользователя с системой. Сценарий диалога позволяет описать процесс взаимодействия пользователя с приложением на уровне решаемой им прикладной задачи. При разработке шага диалога указываются непосредственно участники и содержание сообщений.

2.3 Основные элементы пользовательского интерфейса

В интерфейсах современного программного обеспечения присутствуют кнопки, списки, поля ввода, меню, окна, пиктограммы, курсоры.

Существует несколько видов кнопок:

- командные кнопки;
- кнопки доступа к меню;
- чекбоксы и радиокнопки.

При нажатии на командную кнопку запускает какое-либо явное действие. Считать экранную кнопку нажатой нужно тогда, когда пользователь отпускает нажатую кнопку мыши и курсор находится на экранной кнопке.

Чем больше кнопка, тем легче попасть в нее курсором. Для сигнализации пользователю, того, что кнопка нажимаема, является придание кнопке псевдообъема, т.е. визуальной высоты.

Кнопка должна показывать пользователям свои возможные и текущие состояния. Все командные кнопки должны иметь названия, выраженные в виде глаголов в форме инфинитива (Открыть, Удалить, Добавить). По ним понятно, какое действие произойдет после нажатия.

К группе командных кнопок относится кнопка доступа к меню. Существует много ситуаций, когда раскрывающийся список не помещается в отведенное для него место, поскольку текст в списке слишком велик. В этом случае используют кнопку, нажатие на которую вызывает меню.

Чекбоксы и радиокнопки являются кнопками отложенного действия, т.е. их нажатие не должно инициировать какое-либо немедленное действие. С их помощью пользователи вводят параметры, которые скажутся после, когда действие будет запущено другим элементами управления.

Традиционно сложилось так, что чекбоксы выглядят как квадраты, а радиокнопки – как кружки. Нарушать это правило не рекомендуется. Желательно вертикально располагать чекбоксы и радиокнопки в группе, поскольку это облегчает поиск конкретного элемента.

Каждая подпись около кнопки должна однозначно показывать эффект от выбора соответствующего элемента. Поскольку радиокнопки и чекбоксы не вызывают немедленного действия, формулировать подписи к ним лучше всего в форме существительных, хотя возможно использование глаголов (если изменяется не свойство данных, а запускается какое-либо действие). Подписи к стоящим параллельно кнопкам лучше стараться делать примерно одинаковой длины. Все подписи обязаны быть позитивными (т.е. не содержать отрицания).

В радиокнопках и чекбоксах должны нажиматься не только визуальный индикатор переключения, т.е. кружок или квадратик, но ещё и подпись.

Списки бывают следующих видов:

- пролистываемые списки;
- раскрывающиеся списки;
- комбобоксы.

Пролистываемые списки могут обеспечивать как единственный (аналогично группе радиокнопок), так и множественный выбор (аналогично группе чекбоксов), раскрывающиеся списки работают как радиокнопки.

Ширина списка должна быть достаточна для того, чтобы пользователь мог определить различия между элементами.

В списках могут присутствовать пиктограммы отдельных элементов, которые существенно повышают субъективную привлекательность интерфейса.

Самым простым вариантом списка является раскрывающийся список. Его достоинство заключается в том, что малая высота списка позволяет с большой легкостью визуально отображать составные команды.

Более сложным вариантом списка является пролистываемый список. Пролистываемые списки могут позволять пользователям совершать как единственный, так и множественный выбор.

По-вертикали в список должно помещаться четыре – восемь строк. Полосы прокрутки надо добавлять только в том случае, если без них невозможно обойтись.

Список единственного выбора является промежуточным вариантом между группой радиокнопок и раскрывающимся списком. Он меньше группы радиокнопок с аналогичным числом элементов, но больше раскрывающегося списка. Соответственно, использовать его стоит только в условиях экономии пространства экрана.

Комбобоксами (ComboBox) – называются гибриды списка с полем ввода. Пользователь может выбрать существующий элемент, либо ввести свой. Комбобоксы бывают двух видов – раскрывающиеся и расширенные.

Раскрывающиеся комбобоксы выглядят также как раскрывающиеся списки, визуально отличаясь от них только наличием индикатора фокуса ввода (когда элемент выделен).

Расширенные комбобоксы имеют уникальный вид, отличающийся от остальных элементов управления. Он занимает много места на экране. Поскольку комбобоксы являются гибридами списков и полей ввода, к ним приме-

нимы те же требования.

Вместе с командными кнопками, чекбоксами и радиокнопками, поля ввода являются основой любого интерфейса. Основная часть требований к полям ввода касается размера. Размер по вертикали и ширина поля должны быть производными от размера, вводимого текста.

«Крутилка» (spinner, little arrow) – это поле ввода, обладающее двумя полезными возможностями:

- чтобы ввести значение в «крутилку», пользователю не обязательно бросать мышь и переносить руку на клавиатуру;
- при вводе значения мышью система может позволить пользователям вводить только корректные данные, это резко уменьшает вероятность человеческой ошибки.

Как и ранее описанные элементы управления, ползунки позволяют пользователям выбирать значение из списка, не позволяя вводить произвольное значение. Ползунки незаменимы, если пользователям надо дать возможность выбрать значение, стоящее в хорошо ранжирующемся ряду, когда:

- значений в ряду много;
- нужно передать пользователям ранжируемость значений;
- необходимо дать возможность пользователям быстро выбрать значение из большого количества.

Меню – метод взаимодействия пользователя с системой, при котором пользователь выбирает из предложенных вариантов, а не предоставляет системе свою команду.

Существуют несколько различных вариантов меню.

Во-первых, это:

- статические меню, т.е. меню, постоянно присутствующие на экране. Характерным примером такого типа меню является панель инструментов;
- динамические меню, в которых пользователь должен вызвать меню, чтобы выбрать какой-либо элемент. Примером является обычное контекстное меню.

Во-вторых, это:

- меню, разворачивающиеся в пространстве (например, обычное выпадающее меню). Всякий раз, когда пользователь выбирает элемент нижнего уровня, верхние элементы остаются видимыми;

- меню, разворачивающееся во времени. При использовании таких меню элементы верхнего уровня по тем или иным причинам исчезают с экрана.

Пиктограммы в меню, повторяющие пиктограммы в панели инструментов, обладают способностью обучать пользователей возможностям использования панели. Помимо этого они ускоряют поиск известного элемента и точность его выбора. Пиктограммы лучше работают, когда ими снабжены не все элементы. Когда все элементы имеют пиктограммы, разборчивость каждого отдельного элемента падает.

Пользователей нужно снабжать чувством контроля над системой. Применительно к меню это значит, что по виду элемента пользователи должны догадываться, что произойдет после выбора. Сделать это довольно трудно, поскольку на экране нет места для таких подсказок. Поэтому нужно показать пользователям, какой элемент запускает действие или меняет параметр, а какой открывает окно с продолжением диалога.

Элементы в меню нужно группировать максимально логично. Можно между группами помещать пустой элемент (разделитель) или же размещать отдельные группы в разных уровнях иерархии.

Наличие многих уровней вложенности в меню приводит к «каскадным ошибкам» – выбор неправильного элемента верхнего уровня неизбежно приводит к тому, что все следующие элементы также выбираются неправильно. При этом широкие меню больше нравятся пользователям. Поэтому большинство разработчиков интерфейсов стараются создавать широкие, а не глубокие меню.

Преимущество контекстных (всплывающих) меню заключается в том, что они полностью встраиваются в контекст действий пользователей.

С появлением графического режима, стало возможным реализовать в интерфейсе метафору рабочего стола. Появились окна программ, окна документов и диалоговые окна.

Существуют несколько типов окон:

- главные окна программы;
- окна документа;
- режимные (модальные) диалоговые окна;
- безрежимные (немодальные) диалоговые окна;
- палитры;
- окна браузера.

Структура и само устройство окна или экрана является, самым существенным фактором, влияющим на качество интерфейса. Производительность пользователей можно повысить вдвое, просто изменив расположение элементов управления, не меняя сами эти элементы.

Окна должны иметь следующие особенности:

- хорошо сканироваться взглядом, т.е. его основные части должны быть сразу видны и заметны;
- читаться, как текст;
- удовлетворять закону «релевантное – вперед» (чаще всего используемые элементы должны быть расположены в левой верхней части экрана).

Площадь экрана ограничена, а количество элементов управления, которых может понадобиться необходимо уместить в едином функциональном блоке (т.е. окне), ничем не ограничена. В этом случае приходится использовать вкладки.

Теоретически число вкладок может быть сколь угодно большим. На практике оно ограничивается размером области, в которые ярлыки вкладок могут помещаться. Дело в том, что если ширина всех ярлыков будет больше ширины окна, придется либо делать несколько строк ярлыков, либо скрывать часть из них, пока пользователь не нажмет специальную кнопку. И то и другое плохо.

Фактически, каждая вкладка представляет собой отдельное диалоговое окно внутри другого диалогового окна. Один экран должен содержать только те элементы, которые в нем нужны и которые пользователь может в этом экране ожидать.

Помимо навигации между экранами, существует еще и навигация внутри отдельных экранов. Пользователям необходимо дать возможность максимально быстро переходить к необходимым элементам управления. Для этого у них есть два способа – мышь и клавиатура.

Все программы строятся на основе двух типов окон: главных и вспомогательных, таких как диалоги и окна документов. Выбор типа окон – первый шаг в создании облика программы. В настоящее время распространен подход, при котором разработчик рассматривает интерфейс окна с точки зрения функциональности. На каждую функцию создается свое окно. На самом деле правильнее рассматривать разработку интерфейса с точки зрения задач и целей пользователя.

В традиционном ПО пиктограмм используется довольно много:

- пиктограмма системы;
- пиктограммы отдельных файлов системы, которые пользователи могут как-либо изменять;
- пиктограммы документов, которые создаются и редактируются системой;
- пиктограммы инструментов системы (не всегда, многие системы обходятся без них);
- пиктограммы панели инструментов и меню;
- пиктограммы объектов, использующихся для реализации непосредственного манипулирования.

Все пиктограммы в системе должны обладать единой стилистикой. Под стилистикой понимается последовательное выдерживание единого ракурса и иных изобразительных приемов. Направление теней во всех пиктограммах должно быть одинаковым: снизу справа.

Курсоры это средство обеспечения обратной связи. Так, например, при подведении курсора к углу окна, курсор изменяется, показывая пользователю, что форму окна можно увеличить.

2.4 Особенности графического интерфейса

В основу разработки практически любого графического пользовательского интерфейса (GUI – Graphical User Interface) положены три метафоры: «рабочий стол», «работаешь с тем, что видишь», «видишь, что получил».

Метафора «рабочий стол», в частности, означает следующее.

Для человека, сидящего за рабочим столом, доступны как определенные источники информации, так и средства обработки этой информации. При этом на рабочем столе могут находиться документы, представленные в различной форме: текст, таблицы, графики, рисунки и т.д., относящиеся либо к различным задачам, либо к разным аспектам одной и той же задачи. В любом случае человек имеет возможность просмотреть любой из этих документов, сделать пометки или выборку из них, сравнить содержащиеся в них данные.

Другие две метафоры являются развитием идеи «рабочего стола».

В каждый момент времени сидящий за столом человек может работать только с теми документами, которые он видит перед собой. Если необходимый документ в данный момент отсутствует на столе, его предварительно требуется достать из ящика стола, из папки или из портфеля.

Выполняя какие-то действия над документами, человек, естественно, тут же видит результат своей деятельности.

В рамках графического интерфейса пользователя все три метафоры получили достаточно адекватное воплощение пространство экрана монитора – это рабочий стол пользователя, необходимые для решения задачи объекты представлены на нем в виде соответствующих графических образов (пиктограмм и окон). Для того чтобы изменить рабочую среду, пользователю достаточно изменить состав объектов, представленных на рабочем столе, при этом

все необходимые действия выполняются не с помощью команд, а путем прямого манипулирования объектами (точнее их образами).

Прямое манипулирование объектами обладает следующими достоинствами:

- обеспечивает визуальный контроль за выполняемыми операциями;
- позволяет легко восстановить предшествующее состояние «рабочего стола»;
- позволяет решать различные задачи, используя ограниченный набор стандартных операций (открытие/закрытие окна, буксировка объекта, изменение атрибутов окна или объекта и т.п.).

Еще одна важная особенность современных графических интерфейсов – это многооконность.

Многооконная технология обеспечивает пользователю доступ к большему объему информации, чем при использовании одного экрана. Кроме того, имея через окна доступ к нескольким источникам информации одновременно, пользователь может объединять имеющуюся в них информацию. Например, изображения, полученные с помощью графического редактора, можно включить в текстовый документ.

С помощью нескольких окон пользователь может также одновременно анализировать информацию, представленную на разных уровнях детализации. Наличие на экране нескольких окон или пиктограмм позволяет «расширить» кратковременную память пользователя.

Но прямое манипулирование не всегда удобно, прежде всего, для опытного пользователя. Часто некоторую последовательность действий можно выполнить быстрее с помощью одной команды (или макроса), чем посредством серии манипуляций мышью.

С появлением инструментов визуального программирования, таких как Visual C, Visual Basic, Power Builder, Delphi создание графического пользовательского интерфейса перестало быть прерогативой лишь немногих специалистов. Но одновременно с этим возникла проблема, способная свести на нет все преиму-

щества быстрой разработки GUI. Эта проблема заключается в появлении большого количества плохих графических интерфейсов: не согласованных, не дружелюбных, громоздких, слабо функциональных, не помогающих, а мешающих пользователю решать стоящие перед ним задачи.

2.5 Компоненты графического интерфейса

«Рабочий стол» предоставляет пользователю первичную рабочую область, он заполняет экран и формирует визуальный фон для всех выполняемых операций (рисунок 1). Тем не менее, «Рабочий стол» является не просто фоном, он может также быть использован в качестве основы для размещения объектов файловой системы. Кроме того, для компьютера, подключенного к сети, «Рабочий стол» служит в качестве частной рабочей области, через которую пользователь может получить доступ к другим объектам сети.

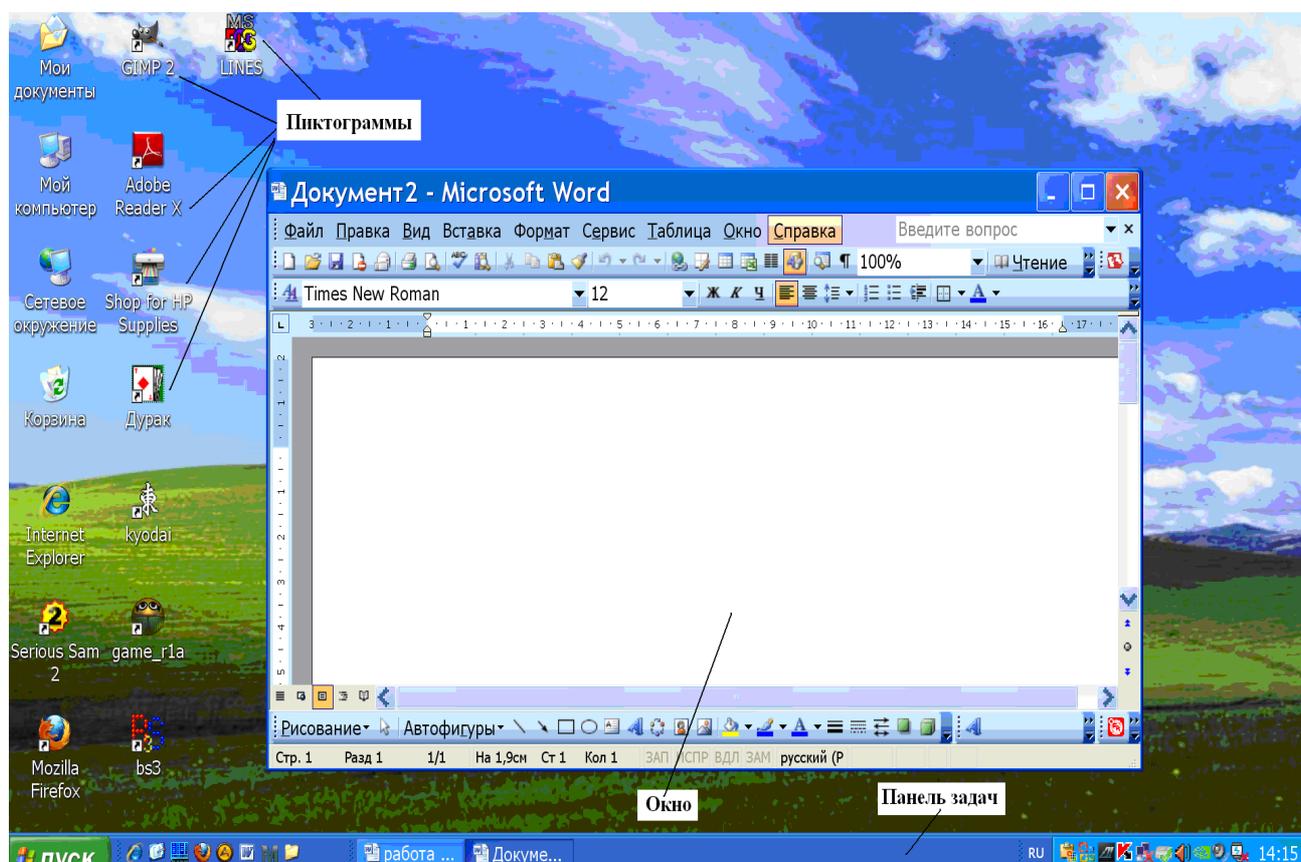


Рисунок 1 – Рабочий стол

Пиктограммы используются для визуального представления на экране объектов или задач. Как правило, это небольшие законченные рисунки, ото-

бражающие сущность представляемых объектов или явлений. Поскольку пиктограммы являются одним из основных средств взаимодействия пользователя с приложением, важно не только обеспечить поддержку существующих (системных) пиктограмм, но и разработать новые, от того насколько они будут соответствовать своему предназначению, будет зависеть и эффективность работы пользователя.

Окно – специальная область физического экрана, с помощью которой пользователь имеет возможность получить отображение определенного аспекта решаемой задачи.

В реальном мире взгляд через разные окна позволяет получить различные изображения внешнего мира. Аналогичную роль играют окна и в графическом интерфейсе. Окно является средством просмотра и редактирования информации, а также отображения содержимого и свойств объектов. Окна могут использоваться также для вывода на экран значений параметров, результатов выполнения команд наборов инструментов и сообщений, информирующих пользователя о конкретной ситуации. Рисунок 2 демонстрирует различные варианты использования окон.

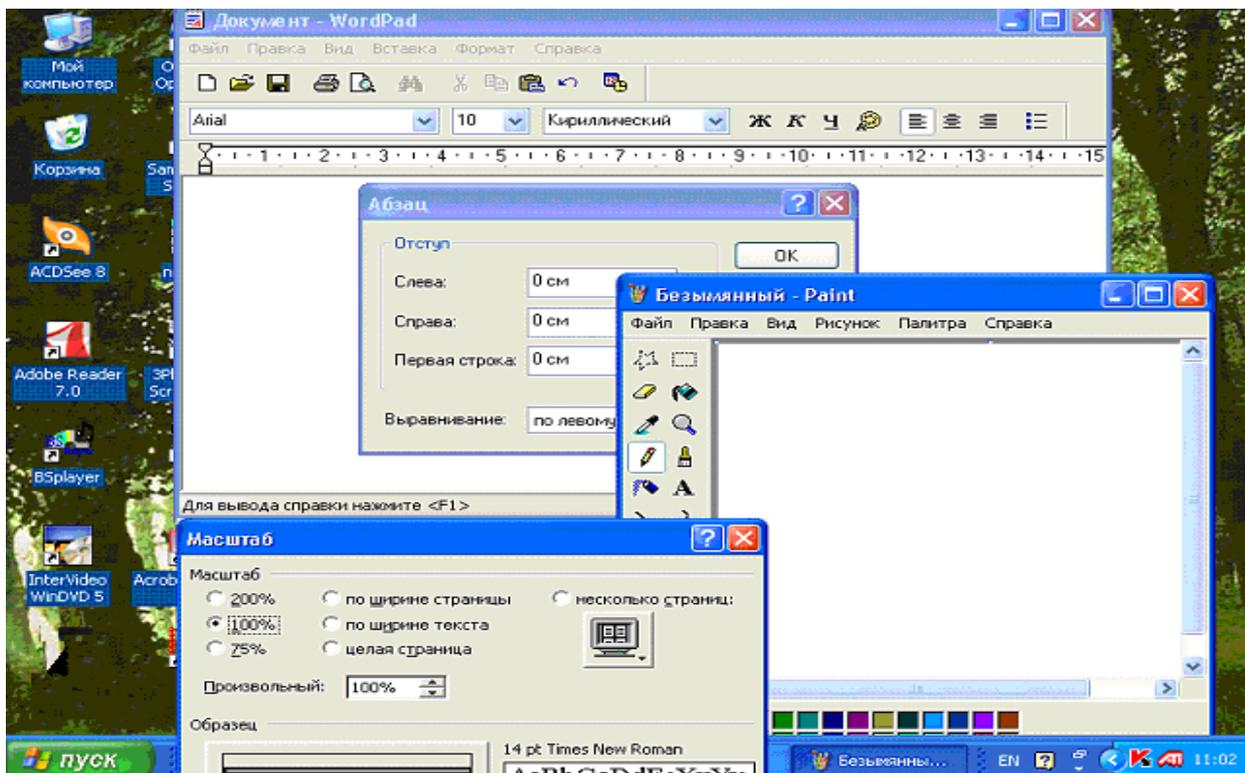


Рисунок 2 – Использование окон приложения

Пользователь может взаимодействовать с объектами приложения, используя различные устройства ввода. Наиболее распространенные из них – мышь, клавиатура и перо.

Мышь – является основным устройством ввода при использовании графического интерфейса. Она функционально связана с графическим символом на экране, который называется указателем. Позиционируя указатель, и нажимая или щелкая кнопку мыши, пользователь может выбирать объекты и операции.

По мере того, как пользователь перемещает указатель по экрану, его форма может изменяться, чтобы обеспечить информирование пользователя (обратная связь) о конкретной позиции, операции или состоянии. В таблице 1 показаны наиболее распространенные формы указателя, используемые в MS Windows, и их назначение.

Таблица 1 – Формы указателя курсора

Форма	Позиция на экране	Допустимое или выполняемое действие
	Поверх большинства объектов	Указание выбора или перемещения
I	Над текстом	Выбор текста
	Над любым объектом или позицией	Выполнение операции
	Над любым объектом или позицией	Выполнение операции в фоновом режиме, указатель остается интерактивным
	Поверх большинства объектов	Режим контекстно-зависимой помощи
	Внутри окна	Масштабирование изображения
	Поверх регулируемой границы	Изменение размера по вертикали

Продолжение таблицы 1

Форма	Позиция на экране	Допустимое или выполняемое действие
	Поверх регулируемой границы	Изменение размера по горизонтали
	Поверх регулируемой границы	Изменение размера по диагонали
	Вдоль границы столбца	Изменение ширины столбца
	Вдоль границы строки	Изменение высоты строки
	Над блоком разбиения окна	Изменение размера (ширины) вертикальных подокон
	Над блоком разбиения окна	Изменение размера (высоты) горизонтальных подокон
	Над объектом-приемником	Приемник недоступен для перетаскиваемого объекта

Взаимодействие пользователя с приложением посредством мыши основано на использовании левой и правой кнопок мыши (в дальнейшем соответственно ЛКМ и ПКМ). Действия, выполняемые с помощью ПКМ, обычно дублируют функции, которые доступны через ЛКМ, но могут быть реализованы альтернативным способом. Система позволяет пользователю изменять распределение кнопок.

Для большинства операций, выполняемых с помощью мыши, нажатие кнопки только идентифицирует начало выполнения операции. Сама же операция выполняется при освобождении кнопки. Исключением является функция авто повторения. Например, установив указатель на стрелку полосы прокрутки и удерживая нажатой кнопку мыши, можно обеспечить непрерывную прокрутку до тех пор, пока кнопка не будет отпущена.

В таблице 2 приведено описание основных действий, выполняемых с помощью мыши.

Таблица 2 – Основные действия, выполняемые с помощью мыши

Действие	Описание
Указание (Pointing)	Установка указателя на конкретный объект на экране без использования кнопок мыши. Указание обычно является частью подготовки к выполнению некоторого другого действия. Для указания во многих случаях имеется возможность обеспечить визуальную обратную связь с пользователем (например, посредством изменения формы указателя).
Щелчок (Clicking)	Щелчок одной из кнопок мыши на объекте, как правило, в момент щелчка мышь не перемещается, щелчок идентифицирует (выбирает) или активизирует объект.
Двойной щелчок (Double-Clicking)	«Сдвоенный» щелчок одной из кнопок мыши на объекте, объект может быть, как выбран ранее, так и не выбран, двойной щелчок обеспечивает выполнение предопределенной операции для данного объекта.
Нажатие (Pressing)	Предполагает установку указателя над одним из объектов (или позиций) на экране с последующим нажатием кнопки мыши, кнопка удерживается в нажатом состоянии достаточно продолжительное время (не менее 1с), обычно нажатие является начальной фазой операции выбора.
Перемещение (Dragging)	Перемещение указателя при нажатой кнопке мыши, как правило, используется для выполнения операций выбора и прямого манипулирования.

Клавиатура – это основное средство ввода или редактирования текстовой информации. При реализации графического интерфейса она может использоваться для ввода других типов данных, а также для управления, переключения режимов и как средство ускоренного доступа к объектам и операциям.

С точки зрения организации взаимодействия пользователя с объектами приложения все клавиши можно разделить на следующие функциональные группы:

- текстовые клавиши и клавиши пунктуации;

- клавиши навигации;
- функциональные клавиши;
- клавиши доступа;
- клавиши режима;
- клавиши-акселераторы.

Клавиша доступа (иногда именуемая также мнемонической клавишей) – это текстовая клавиша, которая при использовании в комбинации с клавишей <Alt> активизирует соответствующий элемент управления. Клавиша доступа должна соответствовать одному из символов текстовой метки этого элемента управления. Например, комбинация клавиш <Alt>+O может активизировать, элемент управления, именуемый метку «Открыть». Эффект активизации элемента управления зависит от типа этого элемента.

Клавиша доступа должна быть уникальной в пределах своей области действия (т.е. одна и та же клавиша не должна использоваться для доступа более чем к одному элементу управления, расположенному в этой области). В противном случае с помощью данной клавиши всегда будет активизироваться только один элемент – расположенный первым в этой области.

Клавиши режима изменяют способ действия других клавиш (или других устройств ввода). Различают два типа таких клавиш: клавиши-переключатели и клавиши-модификаторы.

Клавиша-переключатель включает или выключает конкретный режим при каждом очередном нажатии. Например, нажатие клавиши <Caps Lock> приводит к переключению с верхнего регистра на нижний и наоборот.

Клавиша-модификатор, в отличие от клавиш-переключателей, устанавливает режим, который остаётся в силе, только пока клавиша-модификатор удерживается в нажатом состоянии. К ним относятся клавиши <Shift>, <Ctrl> и <Alt>. Их использование во многих случаях оказывается предпочтительнее из-за того, что они требуют привлечения внимания пользователя к выбору режима и вместе с тем позволяют легко отменить выбранный режим.

Клавиши-акселераторы (называемые также горячими клавишами) – это

клавиши или комбинации клавиш, которые обеспечивают быстрый доступ к часто выполняемым операциям. В качестве таких комбинаций рекомендуется использовать <Ctrl>+<символ> и функциональные клавиши (с F1 по F12). При назначении клавиш-акселераторов необходимо учитывать следующие рекомендации:

- используйте комбинацию <Shift>+<клавиша> для расширения или дополнения действия, выполняемого с помощью этой <клавиши> без <Shift>. Например, если комбинация клавиш <Alt>+<Tab> обеспечивает переключение между окнами сверху вниз, то комбинация <Shift>+<Alt>+<Tab> переключает окна в обратном порядке;

- используйте комбинации <Ctrl>+<клавиша> для усиления эффекта действия, выполняемого с помощью <клавиши>. Например, при редактировании текста клавиша <Home> обеспечивает переход на начало строки, а комбинация <Ctrl>+<Home> – в начало текста;

- избегайте комбинаций <Alt>+<клавиша>, поскольку они могут конфликтовать со стандартным клавиатурным доступом к меню и элементам управления интерфейса. Комбинации <Alt>+<Tab>, <Alt>+<Esc> и <Alt>+<Spacebar> зарезервированы для системного использования <Alt>+<цифра> обеспечивают ввод специальных символов;

- учитывайте назначения клавиш-акселераторов, использованные разработчиками системного программного обеспечения. Например, в ОС MS Windows комбинация <Ctrl>+C используется для выполнения команды Copy (Копировать);

- предоставляйте пользователю право изменять назначения клавиш-акселераторов в вашем приложении, когда это возможно;

- используйте клавишу <Esc> для прерывания выполняемой операции, как правило, она также используется в качестве клавиши-акселератора для выполнения команды Cancel (Отменить);

- сделайте ваше приложение нечувствительным к случайным (ошибочным) комбинациям клавиш.

3 Примеры выполнения лабораторных работ

3.1 Изучение предметной области и оценка интерфейса конкурирующих систем – Лабораторная работа №1

После определения предметной области необходимо изучить ее основные аспекты и оценить интерфейсы конкурирующих систем.

Рассмотрим это на примере задачи вычислительного типа – работе с матрицами.

Программное средство (ПС) «Матрица» - компьютерная программа учебного назначения, предназначенная помочь при изучении линейной алгебры. Он позволяет решать все задачи стандартного курса линейной алгебры, выполнять действия с матрицами, решать численно основные задачи линейной алгебры и представлять их результаты в матричной форме, конструировать алгоритмы решения основных задач линейной алгебры. Данное ПС позволяет:

- вводить произвольные матрицы, элементами которых являются действительные числа или значения арифметических выражений;
- увидеть на экране все введенные матрицы в естественной форме;
- выполнять все арифметические действия с матрицами;
- транспонировать матрицы;
- выполнять элементарные преобразования матриц;
- находить максимальный или минимальный элемент в строке, в столбце или по всей матрице;
- вызывать ранее определенные матрицы из файла для просмотра или для дальнейшего исследования;
- сохранять в файле и печатать все определенные матрицы.

В качестве аналога данной ПС можно использовать MATLAB.

MATLAB (сокращение от англ. «*Matrix Laboratory*») – пакет прикладных программ для решения задач технических вычислений и одноименный язык программирования, используемый в этом пакете. MATLAB используют

инженеры, научные работники, студенты. MATLAB как язык программирования был разработан Кливом Моулером в конце 1970-х годов, когда он был деканом факультета компьютерных наук в Университете Нью-Мексико. Целью разработки служила задача дать студентам факультета возможность использования программных библиотек Linpack и EISPACK без необходимости изучения Фортрана. Вскоре новый язык распространился среди других университетов и был с большим интересом встречен учёными, работающими в области прикладной математики. Инженер Джон Литтл познакомился с этим языком во время визита Клива Моулера в Стэнфордский университет в 1983 году. Поняв, что новый язык обладает большим коммерческим потенциалом, он объединился с Кливом Моулером и Стивом Бангертом и совместными усилиями они переписали MATLAB на С и основали в 1984 компанию The MathWorks для дальнейшего развития. Эти переписанные на С библиотеки долгое время были известны под именем JASCRAS. Первоначально MATLAB предназначался для проектирования систем управления (основная специальность Джона Литтла), но быстро завоевал популярность во многих других научных и инженерных областях. Он также широко использовался и в образовании, в частности, для преподавания линейной алгебры и численных методов.

Основной особенностью языка MATLAB являются его широкие возможности по работе с матрицами, которые создатели языка выразили в лозунге «думай векторно».

ПС «Матрица» будет полезна студентам, изучающим линейную алгебру. Она имеет методическое сопровождение, включающее описание методов решения всех перечисленных выше задач. По сравнению с универсальными математическими пакетами «Матрица» компактна, значительно проще в работе (большинство операций выполняется одним нажатием клавиши), не требует специального изучения, использует более естественную для линейной алгебры форму записи и представления результатов, предъявляет минимальные требования к ресурсам компьютера.

3.2 Описание контекста использования ПС и бизнес-ролей пользователей – Лабораторная работа №1

Пользователи должны обладать элементарными навыками работы с клавиатурой и мышью, а также иметь представление о линейной алгебре и знать, что такое матрица.

Данное программное средство предусматривает одну бизнес роль – роль пользователя, который хочет получить результат вычислений при работе с ПС. Пользователь может выбрать в списке имеющихся задач, необходимую для него, изучить данную тему и непосредственно произвести необходимые действия над матрицами, при этом меняя размерности матрицы и сохранять результаты. Сохранение результатов в последствие дает возможность считывать данные матрицы из файла.

В других случаях пользователей может быть несколько, например для контрольно-обучающихся программ.

1) пользователь (ученик), проходит курс обучения и подвергается тестированию, просматривает результаты предыдущих тестирований;

2) редактор (педагог), имеет право добавлять, удалять, редактировать учебный материал (курс), тестовые задания, просматривать отчеты о тестировании;

3) администратор, имеет возможность распределять полномочия пользователей;

4) гость, имеет возможность, проходить курс обучения и подвергается тестированию с возможностью просмотра только последнего результата.

3.3 Составление сценариев работы пользователей в виде схемы алгоритма, описывающей весь процесс использования ПС для выполнения конкретной задачи – Лабораторная работа №1

На примере ПС «Матрица» сценарии работы могут выглядеть как показано на рисунке 3.

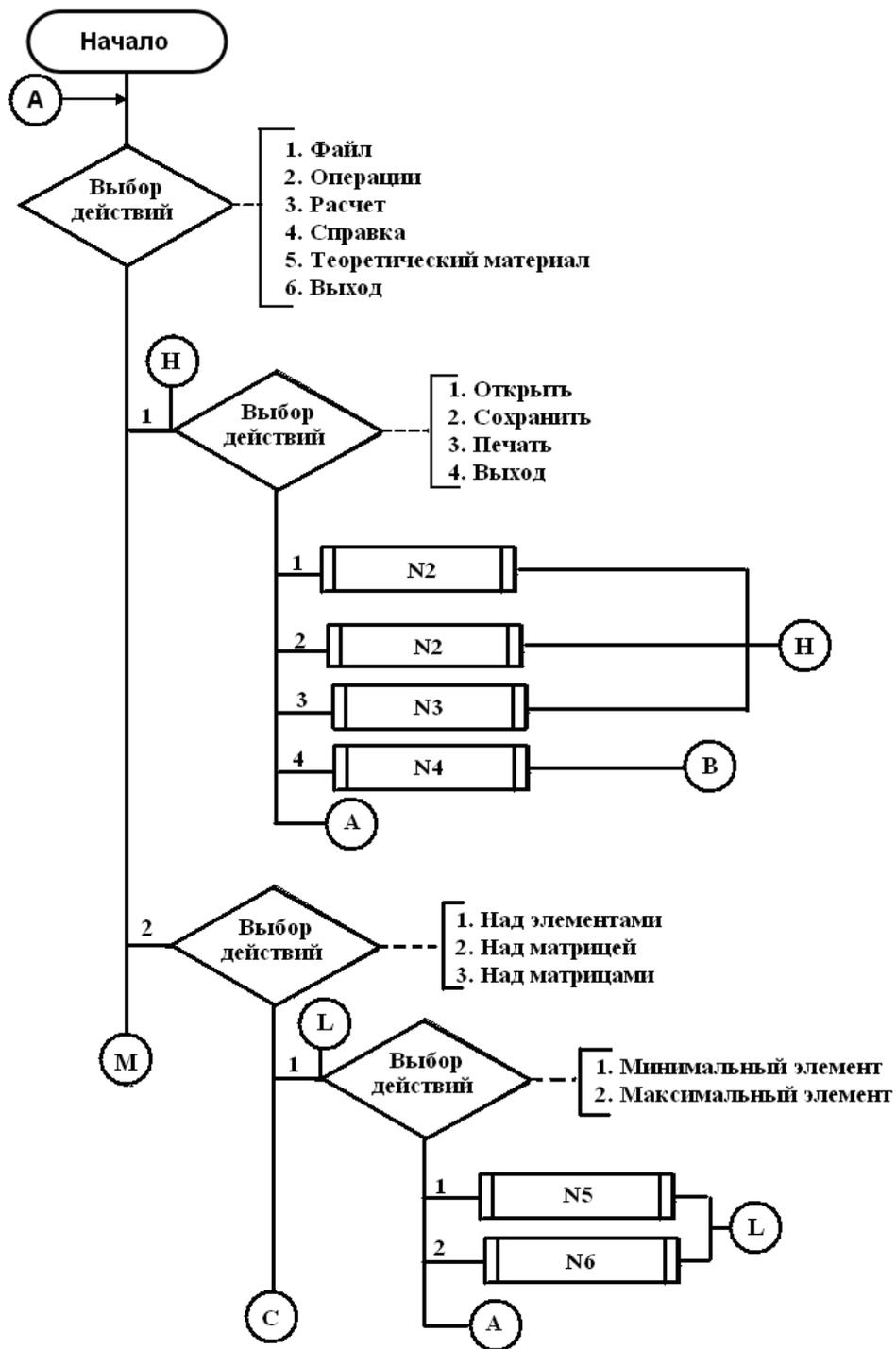


Рисунок 3 – Схема алгоритма, описывающая сценарии работы ПС «Матрица», лист 1

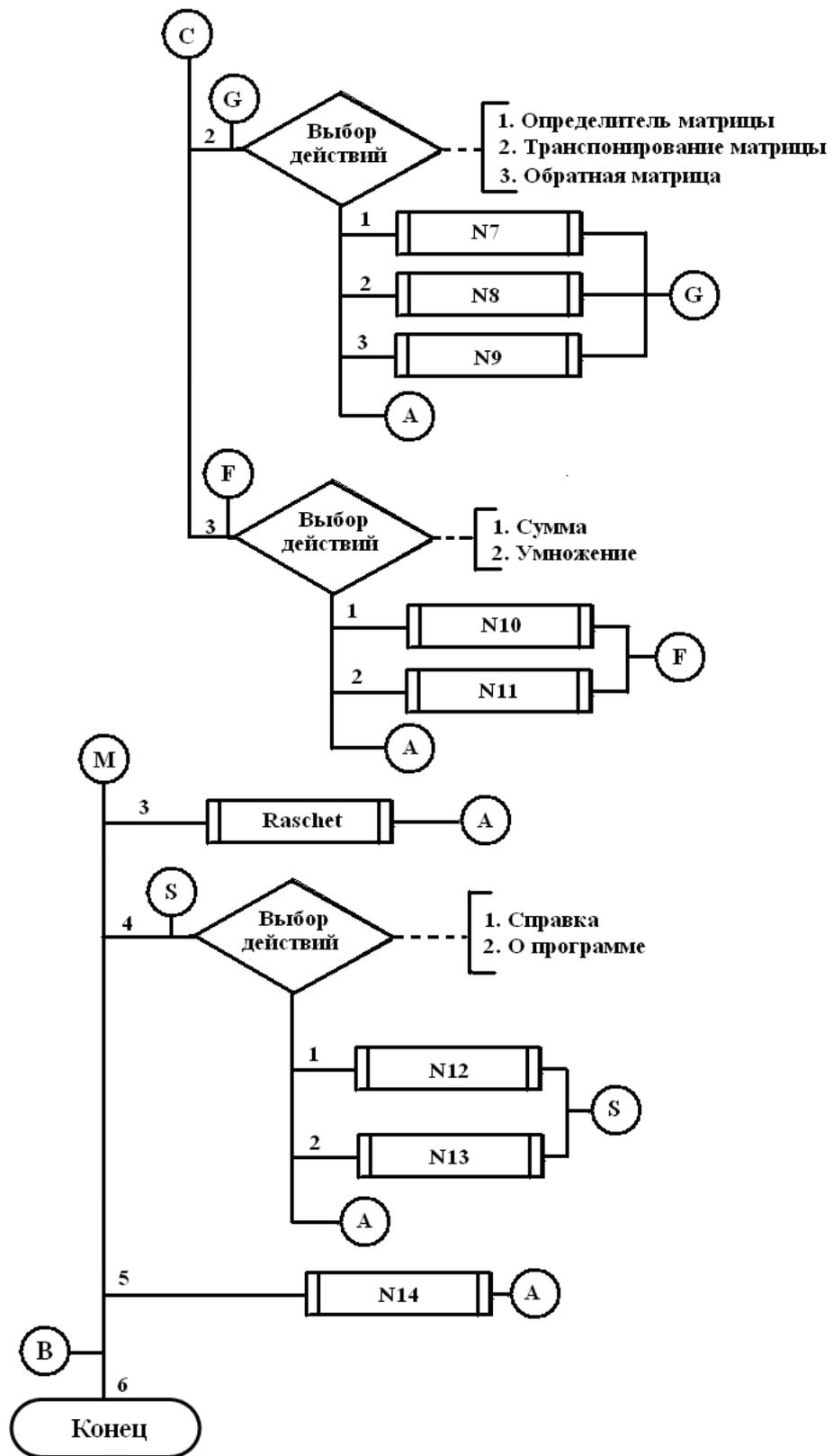


Рисунок 3, лист 2

3.4 Разработка полной схемы экранов проектируемой ПС, определение типа и формы каждого диалога, а также синтаксиса и семантики используемых языков – Лабораторная работа №2

Полная схема экранов представлена на рисунке 4. На рисунке видно как из основного окна программы можно переходить и возвращаться из других окон, то есть проводить навигацию.

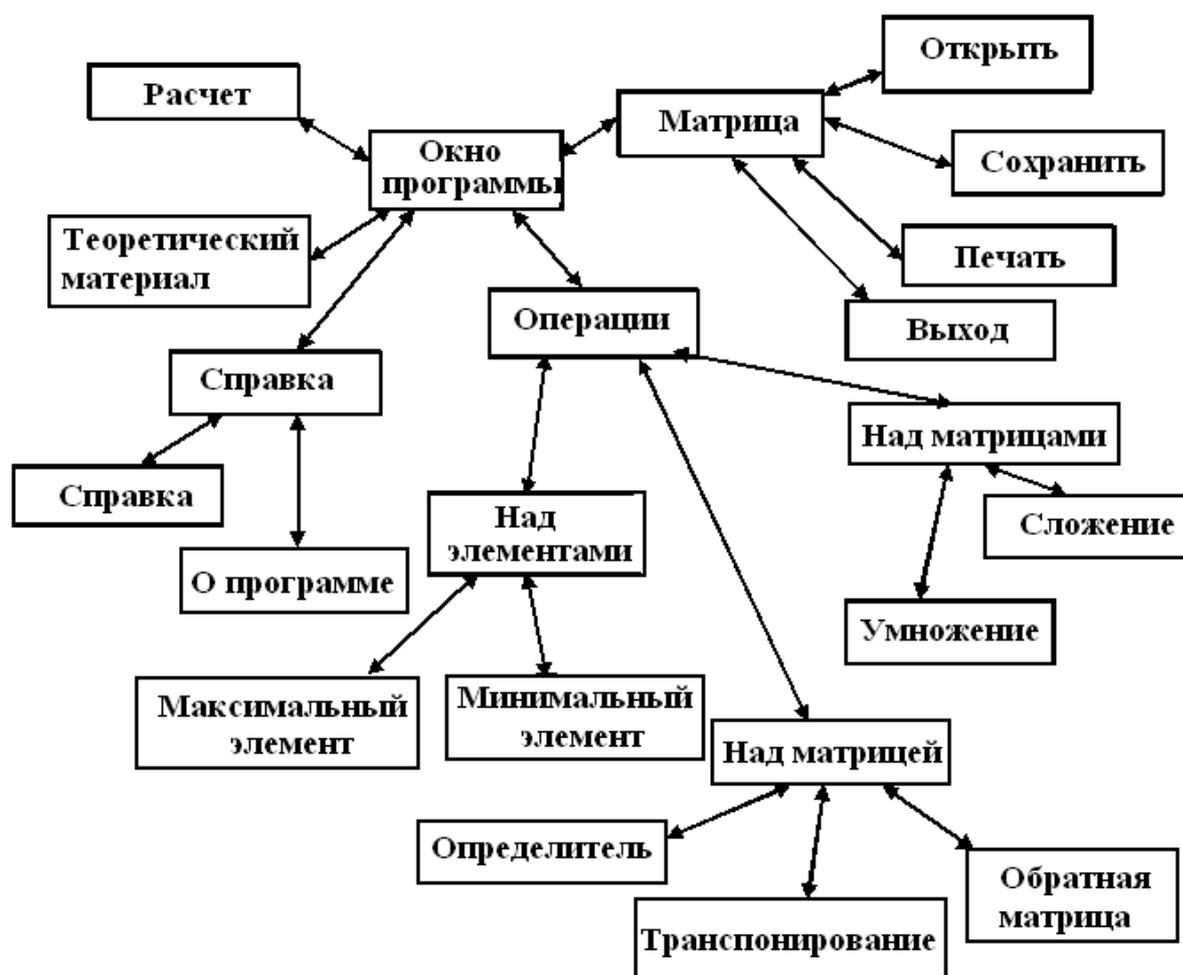


Рисунок 4 – Полная схема экранов ПС «Матрица»

Диалоги рассматриваемой ПС «Матрица» имеют тип диалога управляемого пользователем, т.е. сценарий диалога зависит от действий пользователя. Система предусматривает реализацию различных пользовательских сценариев.

Все диалоги имеют табличную форму. Это означает, что пользователь выбирает предложенные программой варианты работы с матрицами, что

сокращает количество возможных ошибок. Форма очень удобна для непрофессионалов, так как проще выбирать, чем вспоминать.

Язык диалога для табличной формы имеет простейший синтаксис и однозначную семантику.

Основные устройства ввода информации клавиатура и мышь, устройством вывода является монитор. Дополнительным устройством вывода является принтер. Пользователь может отпечатать результаты расчетов, что может являться частью документирования.

Диалог требует от пользователя выбор задачи, определения данных, выбор алгоритма. В результате может появиться сообщение об ошибке ввода (при некорректном вводе).

Для контрольно-обучающей ПС этот пункт может быть более емким.

Нумерация диалогов:

- 1– Авторизация
- 2 – Выбор действия
- 3 – Добавление пользователя
- 4 – Удаление пользователя
- 5 – Редактирование пользователя
- 6 – Добавление курса
- 7 – Удаление курса
- 8 – Редактирование курса
- 9 – Добавление теста
- 10 – Удаление теста
- 11 – Редактирование теста
- 12 – Просмотр курса
- 13 – Просмотр результата тестирования
- 14 – Прохождение теста

В таблице 3 приведены формы и типы диалогов.

В таблице 4 представлен синтаксис диалогов.

В таблице 5 показана семантика диалогов.

Таблица 3 – Формы и типы диалогов

Диалог	Форма диалога	Тип диалога
1	Директивная	Управляемый пользователем
2, 3, 5, 6, 8, 9, 11, 12, 13	Табличная	Управляемый пользователем
4, 7, 10, 14	Табличная	Управляемый программой

Таблица 4 – Синтаксис диалогов

Номер диалога	Синтаксис
1, 5	Для имени: a-z, A-Z, "_", а-я, А-Я для пароля: a-z, A-Z, 0-9, "/", ", ", "+", "-", "=", "?", "*", "_", "."
3	a-z, A-Z, "_", а-я, А-Я
6, 8, 9, 11	Выводимые на печать символы ANSI

Таблица 5 – Семантика диалогов

Номер диалога	Семантика
1,5	Имя, пароль
3	Имя пользователя
6, 8	Название курса, текст курса
9, 11	Название теста, текст тестовых вопросов

В таблице 6 приведены устройства ввода-вывода для каждого диалога.

Таблица 6 – Устройства ввода-вывода

Устройство ввода	Диалог	Устройства вывода
Клавиатура/мышь	1-14	Монитор

Основные устройства мышь, клавиатура, монитор.

Дополнительное устройство – принтер.

Сообщения, вызывающие пользователя на диалог приведены в таблице 7.

Таблица 7 – Сообщения, поддерживающие диалог

Номер диалога	Сообщение
1	Введите имя пользователя и пароль
3	Введите имя нового пользователя и пароль
4	Выберите пользователя для удаления
6	Введите название и текст курса
7	Выберите учебный курс для удаления
9	Введите название нового теста и текст тестовых заданий
10	Выберите тест для удаления

3.5 Проектирование меню MDI-интерфейса различного вида: основное, панели инструментов, контекстное и кнопочное. Использование директивной формы диалога (как альтернативной) с использованием соответствия основным командам комбинации клавиш – Лабораторная работа №2

Для обеспечения удобства использования программы, меню должно иметь интуитивно-понятный интерфейс, который не будет вызывать особых трудностей у неподготовленного пользователя. Также, в программе должны быть предусмотрены различные подсказки или пояснения, облегчающие использование программы. В программе должна быть предусмотрена справка, ко-

торая подробно описывает принципы работы с программой. Основное меню программы «Матрица» показано на рисунке 5.

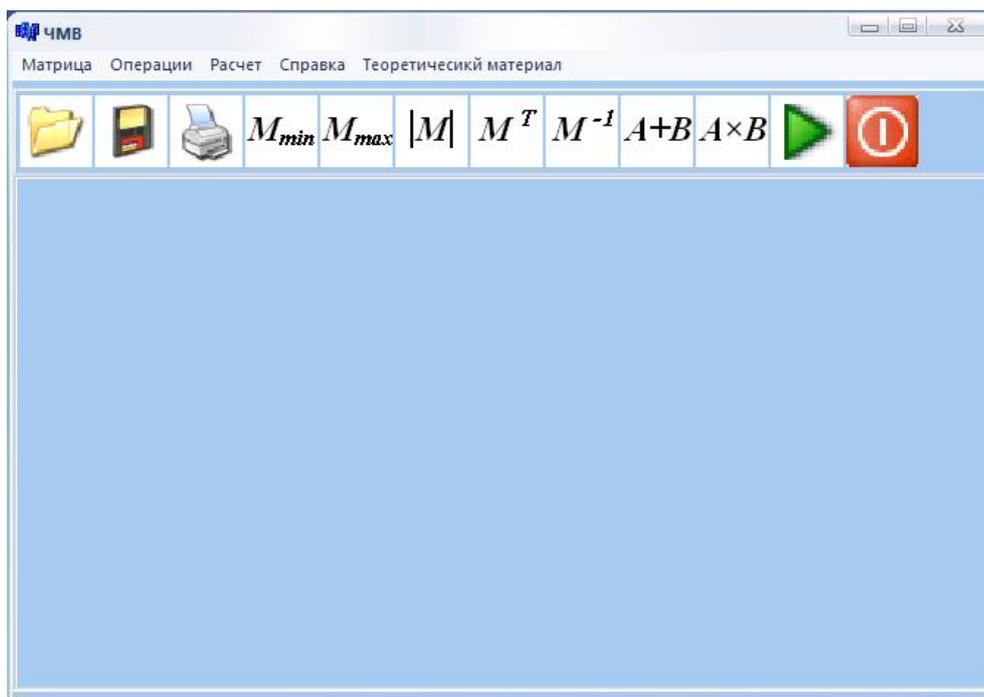


Рисунок 5 – Интерфейс программы

В основном меню представлена возможность доступа ко всем основным функциям программы. Некоторые из этих функций продублированы в других видах меню, для реализации более удобного доступа к ним. Основное меню состоит из следующих вкладок: «Матрица», «Операции», «Расчет», «Справка», «Теоретический материал». «Матрица» содержит пункты: «Открыть», «Сохранить», «Печать», «Выход». «Теоретический материал» включает в себя: «Открыть теорию». «Операции» содержит пункты: «Над элементами», который в свою очередь содержит пункты: «Минимальный элемент» и «Максимальный элемент»; «Над матрицей», который в свою очередь тоже содержит пункты: «Определитель матрицы», «Транспонирование матрицы», «Обратная матрица»; «Над матрицами», который содержит пункты «Сумма» и «Умножение». «Справка» включает саму справку и пункт «О программе».

Для обеспечения удобства навигации, необходимо предусмотреть меню различных видов. Это делается для облегчения навигации в программе и осуществления группировки элементов управления по различным признакам. Кон-

текстное меню показано на рисунке 6.

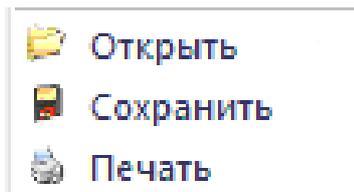


Рисунок 6 – Контекстное меню ПС «Матрица»

На рисунке 7 представлено кнопочное меню, предоставляющее возможности более удобной навигации. Кнопочное меню дублирует основное меню, за исключением кнопки «О программе».

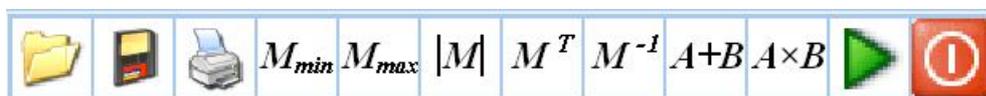


Рисунок 7 – Кнопочное меню

В программе предусмотрено управление меню при помощи клавиатуры, а также возможность сочетания клавиш для быстрого доступа к функциям программы. Это управление меню при помощи «горячих клавиш», изображено на рисунках 8-12.

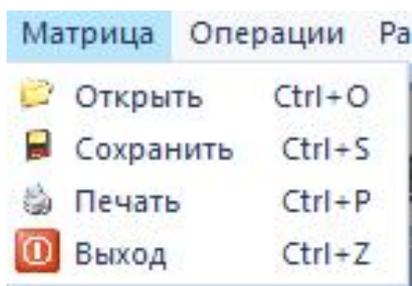


Рисунок 8 – «Горячие клавиши» пункта меню «Матрица»

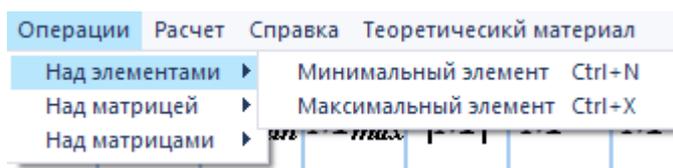


Рисунок 9 – «Горячие клавиши» пункта меню «Над элементами»

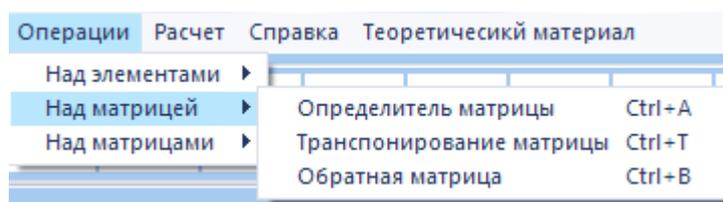


Рисунок 10 – «Горячие клавиши» пункта меню «Над матрицей»

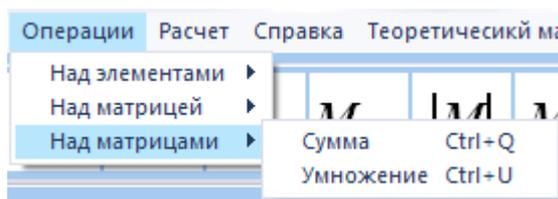


Рисунок 11 – «Горячие клавиши» пункта меню «Над матрицами»

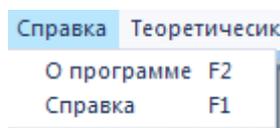


Рисунок 12 – «Горячие клавиши» пункта меню «Справка»

3.6 Определение состава, внешнего вида и взаимного расположения визуальных интерфейсных компонентов в соответствии с эргономическими принципами проектирования ПС – Лабораторная работа №3

При проектировании данного интерфейса учитывались способности человека, связанные с восприятием, запоминанием и обработкой информации.

Данное программное средство дает возможность контроля пользователем, т.е.:

- пользователь может выбрать работать ли ему с мышью, клавиатурой или использовать их комбинации;
- возможность обратной связи;
- интерфейс является простым, что обеспечивает его наибольшую понятность;

В интерфейсе данного программного средства используются холодные цвета, чтобы пользователь сосредоточился на выполнении поставлен-

ной задачи, а не на отдельных компонентах интерфейса.

Для легкого ориентирования в интерфейсе, используется быстрый доступ к компонентам в виде пиктограмм, на которых изображены объекты реального мира.

Учитывая перечисленные принципы разработки, данный интерфейс разработан следующим образом:

- строка меню и часть панели инструментов располагается в верхнем левом углу экрана, под ними располагается окно для работы с матрицами;
- программные пиктограммы располагаются в правом верхнем углу экрана.

3.7 Разработка ПС, содержащего элементы управления в многооконных интерфейсах с использованием инструментальных сред программирования графического диалога. Программирование различных реакций ПС на действия пользователей – Лабораторная работа №3

Многообразие элементов управления имеет за собой несколько функций, которые положительно влияют на восприятие графического интерфейса программы пользователем, во-первых, это функциональное разнообразие программы, т.е. множество функций, которое можно реализовать с помощью тех или иных элементов управления, а во-вторых, это возможность создать свой непосредственный стиль приложения на основе различного вида форм элементов управления, которые в избытке существуют во всяческих библиотеках компонент.

Спецификация типовых объектов управления графического интерфейса представлена в таблице 8.

На рисунке 13 показан макет главного окна ПС с описанием функциональности интерфейсных элементов.

Таблица 8 – Спецификация типовых объектов управления графического интерфейса

Название	Описание
Меню	Выбор опций работы программы по пунктам
Кнопка	Выполняет действия по заранее описанному алгоритму
Текстовое окно	Служит для отображения текстовой информации произвольного вида
Линии горизонтальной и вертикальной прокрутки	Для горизонтального и вертикального перемещения документа
Однострочный текстовый редактор	Служит для ввода или отображения строки данных

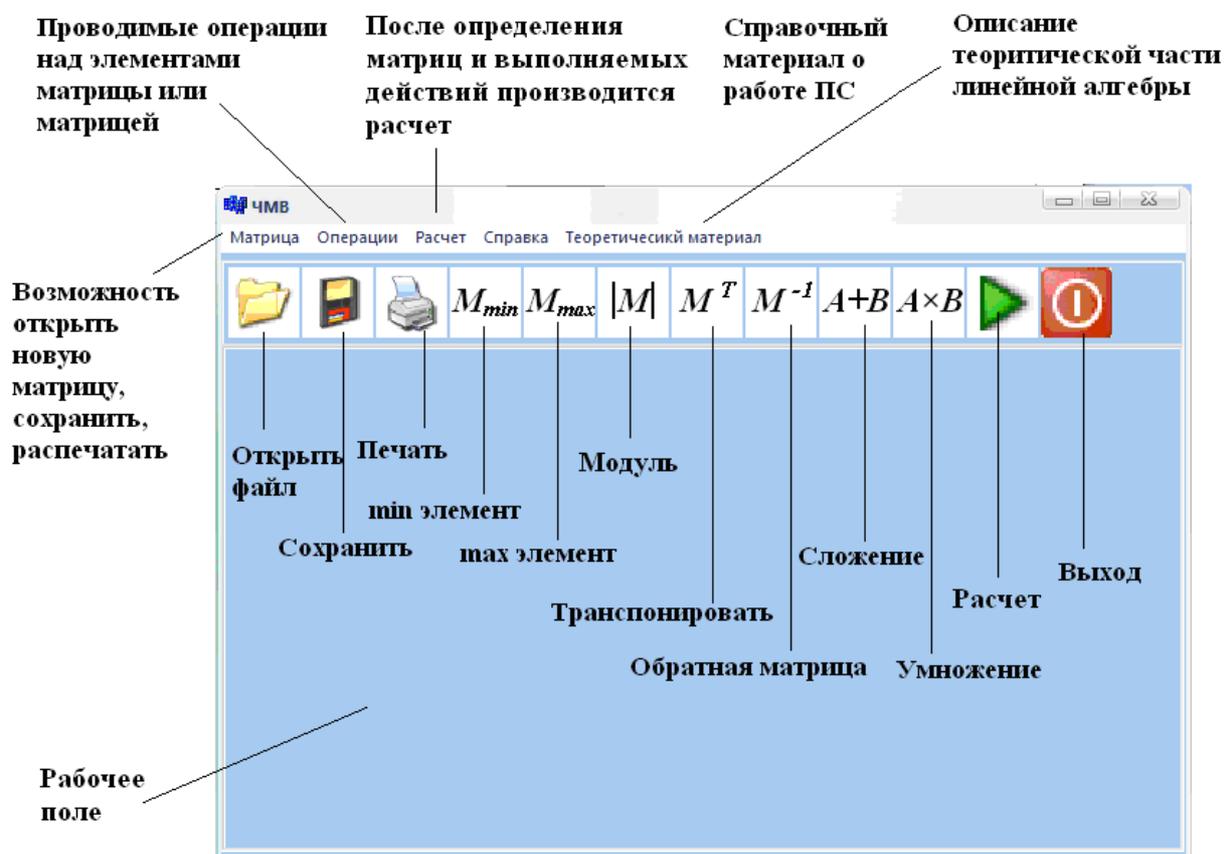


Рисунок 13 – Интерфейсные элементы главного окна программы

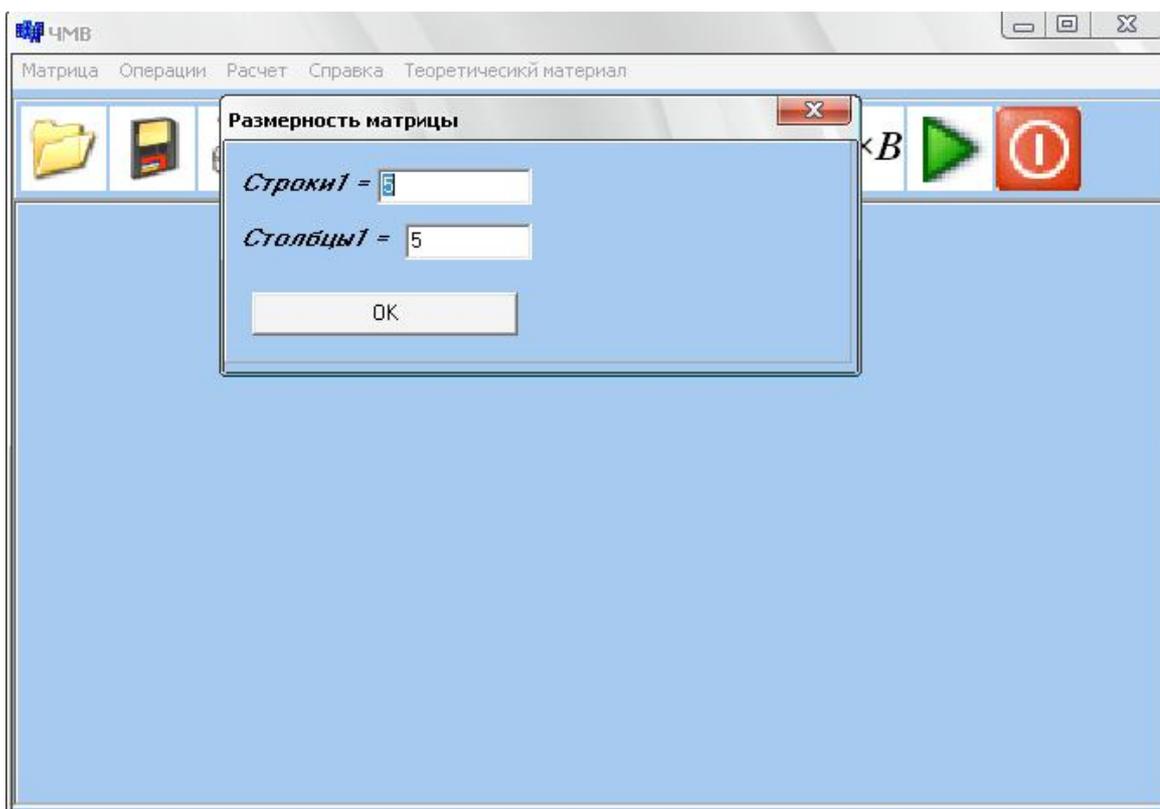


Рисунок 15 – Элементы окна Операции (создание новой матрицы)

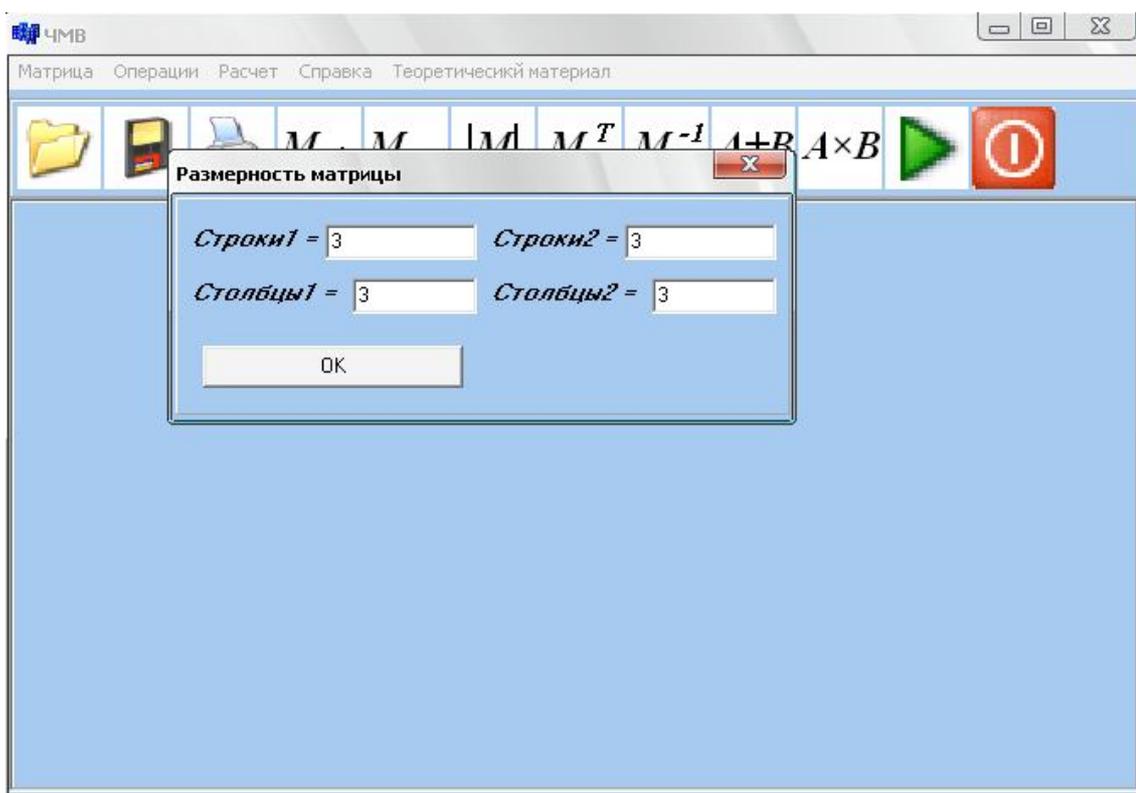


Рисунок 16 – Элементы окна Операции (создание новых матриц)

На рисунках 17 и 18 показаны интерфейсные элементы окна Расчет.

Реакция программного средства на действия пользователя определяет алгоритм поведения пользователя для достижения им той или иной цели.

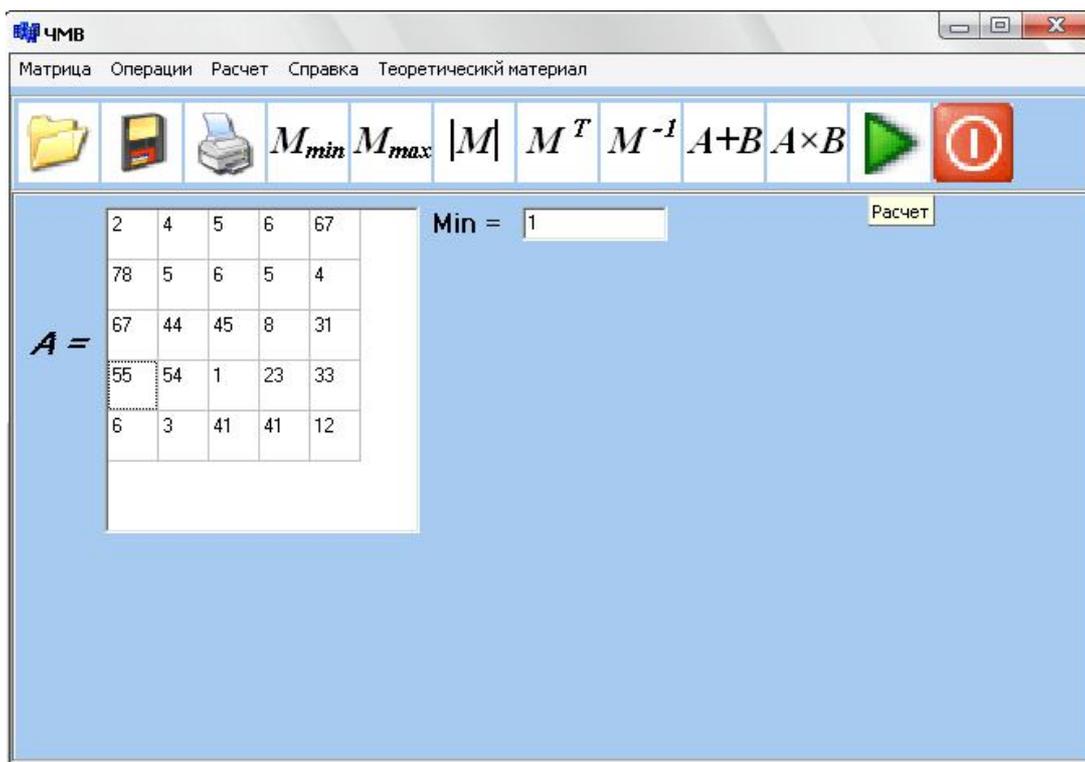


Рисунок 17 – Интерфейсные элементы окна Расчет (работа с одной матрицей)

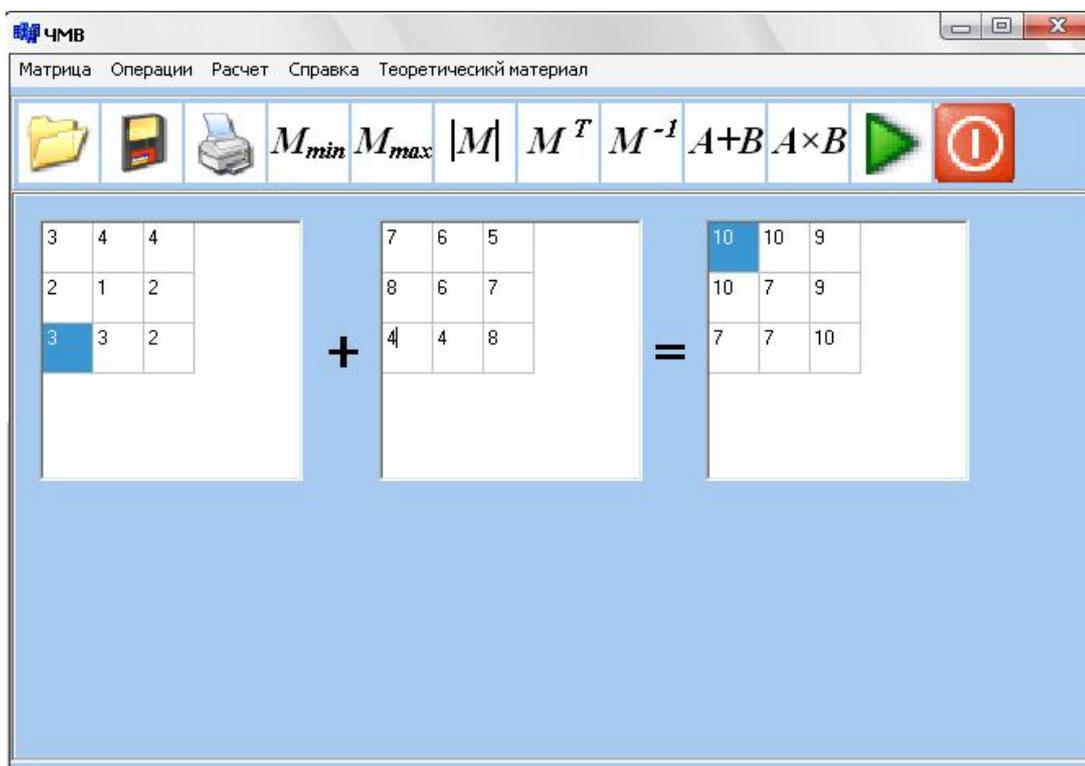


Рисунок 18 – Интерфейсные элементы окна Расчет (работа с двумя матрицами)

Одно из реакции программного средства на действие пользователя может быть представлено при вводе не всех данных, или при вводе некорректных данных (рисунок 19).

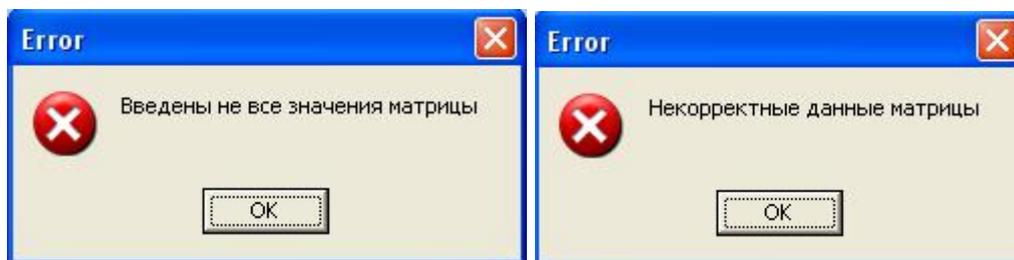


Рисунок 19 – Сообщение об ошибке

При сложении матриц, если пользователь ввел неправильную размерность, или размерность введена некорректно (рисунок 20).

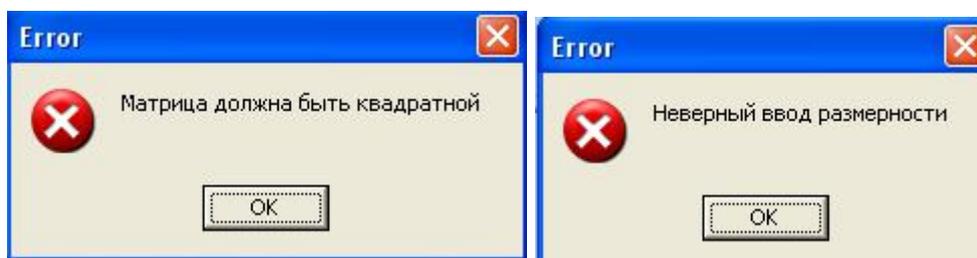


Рисунок 20 – Сообщение об ошибке

По условию умножения матриц, строки первой матрицы должны быть равны столбцам второй матрицы. В том случае, если этого не произошло, то выдается сообщение об ошибке (рисунок 21).

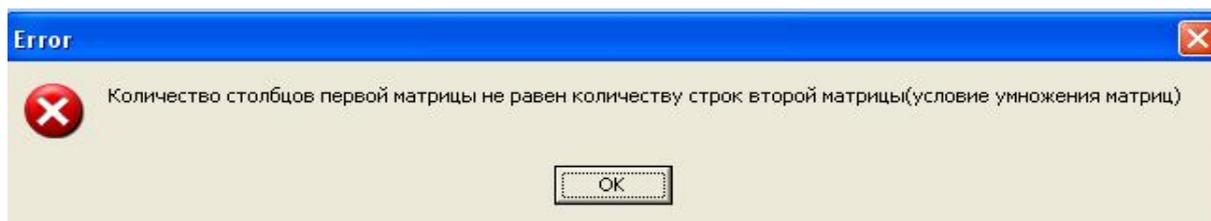


Рисунок 21 – Сообщение об ошибке

В том случае, если не подключен принтер для распечатывания результата, то выдается сообщение (рисунок 22).

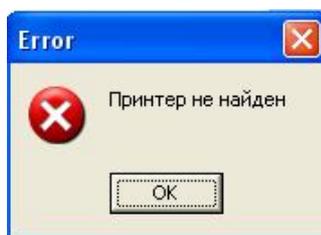


Рисунок 22 – Сообщение об ошибке

Заключение

В результате выполнения лабораторных работ студенты закрепляют теоретические знания и формируют практические навыки проектирования диалогов пользователя с ПС, а также использования инструментальных сред программирования для реализации диалогов в графическом пользовательском интерфейсе по технологии WIMP.

Представленный теоретический материал позволяет использовать рекомендации по проектированию диалога пользователя с ПС, изучить особенности графического интерфейса и составляющие компоненты.

На примере задачи вычислительного типа – ПС Матрица» и контрольно обучающего ПС описано изучение предметной области, оценка интерфейсов конкурирующих систем, описан контекст использования ПС и бизнес-роли пользователей. Также составлен сценарий работы пользователей в виде схемы алгоритма, разработана полная схема экранов проектируемой ПС, определены типы и формы каждого диалога, а также синтаксис и семантика используемых языков. Показано проектирование меню MDI-интерфейса различного вида: основное, панели инструментов, контекстное и кнопочное, определен состав внешнего вида и взаимного расположения визуальных интерфейсных компонентов в соответствии с эргономическими принципами проектирования ПС.

Приведены элементы управления в многооконных интерфейсах с использованием инструментальных сред программирования графического диалога, а также различные реакции ПС на действия пользователей.

Список использованных источников

- 1 Акчурин, Э. А. Человеко-машинное взаимодействие: учебное пособие / Э. А. Акчурин. – М.: СОЛОН-ПРЕСС, 2008. – 96 с.: ил. ISBN 978-5-91359-022-0
- 2 Глушаков, С. В. Microsoft Access 2007: теория баз данных, таблицы и запросы, формы и отчеты, пользовательский интерфейс: лучший самоучитель / С. В. Глушаков, А. С. Сурядный, М. И. Шумилов. – М.: АСТ, 2008. – 444 с.: ил. ISBN 978-5-17-053642-9
- 3 Гультияев, А. К. Проектирование и дизайн пользовательского интерфейса / А. К. Гультияев, В. А. Машин. – СПб.: КОРОНА, 2000. – 349 с.: ил. ISBN 5-7931-0092-X
- 4 Константин, Ларри Л. Человеческий фактор в программировании / Ларри Л. Константин; [Пер. с англ. Ю. Асотова]. – СПб.: М.: Символ, 2004. – 382 с. ISBN 5-93286-044-8
- 5 Купер, А. Аллен Купер об интерфейсе. Основы проектирования взаимодействия. /Купер А., Рейман Р., Кронин Д. [Пер. с англ.] – СПб.: Символ-Плюс, 2009. – 688 с.: ил. ISBN 978-5-93286-132-5
- 6 Логунова, О. С. Человеко-машинное взаимодействие: теория и практика: Учебное пособие / О. С. Логунова, И. М. Ячиков, Е. А. Ильина. – Ростов н/Д: Феникс, 2006. – 286 с.: ил. – (Высшее образование). ISBN 5-222-09156-2
- 7 Магазанник, В. Д. Человеко-компьютерное взаимодействие: учебное пособие для студентов высших учебных заведений, обучающихся по специальности «Прикладная информатика (по областям)» / В. Д. Магазанник. – М.: Логос, 2007. – 254 с.: ил. ISBN 978-5-98704-241-0
- 8 Мандел, Т. Разработка пользовательского интерфейса: [Пер. с англ.] / Тео Мандел. – М.: ДМК Пресс, 2001. – 416 с.: ил. ISBN 5-94074-069-3
- 9 Минаси, М. Графический интерфейс пользователя. Секреты проектирования /М. Минаси; Перевод с англ. Р. П. Богатырева. – М.: Мир, 1996. – 159 с.: ил. ISBN 5-03-003155-3

10 Раскин, Д. Интерфейс: новые направления в проектировании компьютерных систем /Д. Раскин; [пер. Ю. Асотова]. – Санкт-Петербург; М.: Символ®, 2010. – 268 с.: ил. ISBN 978-5-93286-030-4

11 Рубин, А.Г. Пользовательский интерфейс для прикладных задач /А.Г. Рубин, В.К. Смирнов, В.П. Тульский. – М., 2000. - 29 с.: ил.

12 Торрес, Роберт Дж. Практическое руководство по проектированию и разработке пользовательского интерфейса/ Р. Дж. Торрес; [Пер. с англ. и ред. В.М. Неумоина]. – М.: Вильямс, 2002 (ГПП Печ. Двор). – 390 с.: ил. ISBN 5-8459-0367-X