**Введение**

На курсовое проектирование была поставлена задача разработать электронное средство обучения «Визуальные компоненты Delphi».

Цель курсового проекта заключается в знакомстве массового пользователя с типами компонентов Delphi, визуальными компонентами Delphi, их видами, видимыми элементами пользовательского интерфейса, используемыми при разработке проектов.

Создаваемое электронное средство обучения рассчитано на любого рода пользователей. Применить его смогут люди начинающие заниматься программированием, интересующиеся визуальным программированием.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название “Анализ задачи”. В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе “Инструменты разработки” будет рассмотрена среда, в которой создается данный курсовой проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе “Проектирование задачи” будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет четко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

“Реализация задачи” – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут четко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвертый раздел – “Тестирование”. В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе “Применение” будет описано назначение, область применения, среда функционирования курсовой программы. Также в нем будет описано использование справочной системы.

“Заключение” будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В “Литературе” будет приведен список используемых при разработке источников.

В приложениях к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

# Анализ задачи

* 1. **Постановка задачи**

Темой данного курсового проекта является «Разработка электронного средства обучения «Визуальные компоненты Delphi».

Цель курсового проекта заключается в знакомстве массового пользователя с типами компонентов Delphi, визуальными компонентами Delphi, их видами, видимыми элементами пользовательского интерфейса, используемыми при разработке проектов.

Создаваемое электронное средство обучения рассчитано на любого рода пользователей. Применить его смогут люди начинающие заниматься программированием, интересующиеся визуальным программированием.

Данный проект должен стать общедоступным для всех пользователей. В поставленной задаче необходимо реализовать максимально простой пользовательский интерфейс, позволяющий использовать проект пользователю, не обладающему дополнительными знаниями электронно-вычислительной техники.

В разрабатываемой программе будут использоваться три вида данных. К входной информации можно отнести вводимые пользователем значения, например ответы на вопросы при прохождения теста. К выходной – результат прохождения теста.

Постоянной информацией в проекте будут являться текстовые файлы, графические изображения.

Программный продукт предоставляет функционал для следующего ряда пользователей: администратор – организовывает работу программой, гость – просмотр возможностей, взаимодействие с визуальными компонентами Delphi, прохождение теста, просмотр справочной системы.

Разрабатываемый программный продукт должен позволять выполнять следующие действия:

- просмотр полной информации о компонентах;

- взаимодействие с визуальными компонентами;

- просмотр справки;

-прохождение теста;

# Инструменты разработки

1. **Delphi 10** – главная среда разработки программы, в которой прописывается код и создаются формы. Для разработки данного проекта будет выбрана среда Delphi 10, так как это самое удобная и доступная среда разработки на данный момент. Delphi 10 - язык программирования, относящийся к классу RAD- (Rapid Application Development – «Средство быстрой разработки приложений») средств CASE – технологии. Delphi 10 сделал разработку приложений для Windows быстрым и приятным процессом. Теперь разрабатывать сложные и интересные проекты можно только одним человеком, использующим Delphi 10.
2. **Microsoft Word 2010** – редактор текста для написания документации.
3. **Microsoft Power Point 2010** – программа для создания презентации.
4. **Google-браузер** – для нахождения информации о создании электронных средств обучения в среде Delphi 10, поиск информации о визуальных компонентах Delphi.

Для оптимального функционирования разрабатываемого приложения необходимо наличие у компьютера следующих параметров:

− 1.8 GHz процессор (или более быстрый);

− 1 GB RAM;

− от 200 MB доступного места на жестком диске.

Разработка ведется на ноутбуке Acer. У данного ноутбука следующие параметры:

* процессор Intel pentium;
* объем ОЗУ 4 гб;

- объем места на HDD – 512 гб ;

* Видеоадаптер NVIDIA GeForse 820M;
* ОС – Windows 7.

Как видно разрабатываемое приложение не очень требовательно к аппаратным ресурсам, что, является большим плюсом.

# Требования к приложению

На этапе исследования предметной области был установленный целый ряд требований, которые предъявляют к разрабатываемой задаче.

При моделировании форм следует учесть такие моменты:

* интуитивно понятный интерфейс
* небольшое количество информации на каждой форме.

Так же следует обратить внимание на такие моменты:

* большие окна формы
* визуально понятный и приятный глазу интерфейс;
* средние кнопки.

Минимальные системные требования:

− 1.8 GHz процессор (или более быстрый);

− 1 GB RAM;

− от 75 MB доступного места на жестком диске.

Рекомендуемые:

− двухъядерный AMD Athlon или аналогичный процессор Intel (или более быстрый) ;

− 2 GB RAM ;

− от 75 MB доступного места на жестком диске.

# Требования к надежности:

Специальных требований к надежности не предъявляется.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбое (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

реализации данного приложения. В этом разделе будут четко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвертый раздел – “Тестирование”. В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе “Применение” будет описано назначение, область применения, среда функционирования курсовой программы. Также в нем будет описано использование справочной системы.

“Заключение” будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В “Литературе” будет приведен список используемых при разработке источников.

В приложениях к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

**Проектирование задачи**

**2.1 Организация данных**

Проектирование задачи – это очень важный и ответственный этап в разработке любого приложения.

Важным является он вследствие того, что методы, по средствам которых пользователь управляет формами, построены на высокой степени специализации каждого из компонентов.

Необходимым условием при разработке данного приложения является описание организации данных, т.е. логическая и физическая структура данных в контексте среды разработки. В разрабатываемой программе будут использоваться три вида данных.

Первым видом являются данные, которые будут введены разработчиком на этапе реализации задачи. Сюда можно отнести изображения (иконки), описание, исходные коды сортировок.

Вторым видом данных, используемых в программе, является вводимая пользователем информация. Входной информацией в разрабатываемой программе будут являться данные, которые будут отсортированы: количество данных (количество элементов), и сами данные (элементы). Необязательно заполнять сами элементы вручную, поскольку эти данные, по желанию пользователя, могут быть сгенерированы самой программой.

Третьим видом данных является результат программы – отсортированные пользовательские данные. Его также относят к отдельному виду, так как ни пользователь, ни разработчик его не вводят, а программа сама получает его в результате выполнения определенных действий. Таким образом, организация данных является важной задачей при разработке данной и любой программы.

**2.2 Процессы**

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно должна заниматься разрабатываемая программа. Главной её задачей является знакомство массового пользователя с типами компонентов Delphi, визуальными компонентами Delphi, их видами, видимыми элементами пользовательского интерфейса, используемыми при разработке проектов.

Для реализации задач используются процедуры и функции. С помощью процедуры, например, будет осуществляться закрытие программы.

Пользователь взаимодействует с визуальными компонентами Delphi, знакомясь с ними. На форме множество компонентов, а так же краткая информация о них. Каждый компонент имеет свои свойства, которые пользователь может посмотреть в справочной системе, которая будет доступна на главной форме.

**2.3 Описание внешнего пользовательского интерфейса**

Важным при выполнении курсового проекта является организация диалога между пользователем и самой программой. Во многом это зависит от того, как программист разрабатывает данную программу, какие компоненты использованы и какие методы автоматизированы.

Особое внимание уделяется интерфейсу. Внешний вид программы организован так, чтобы пользователь понял, что от него требуется. Таким образом, ключевые кнопки графически выделены для визуализации интерактивных возможностей приложения (Рисунок 1).

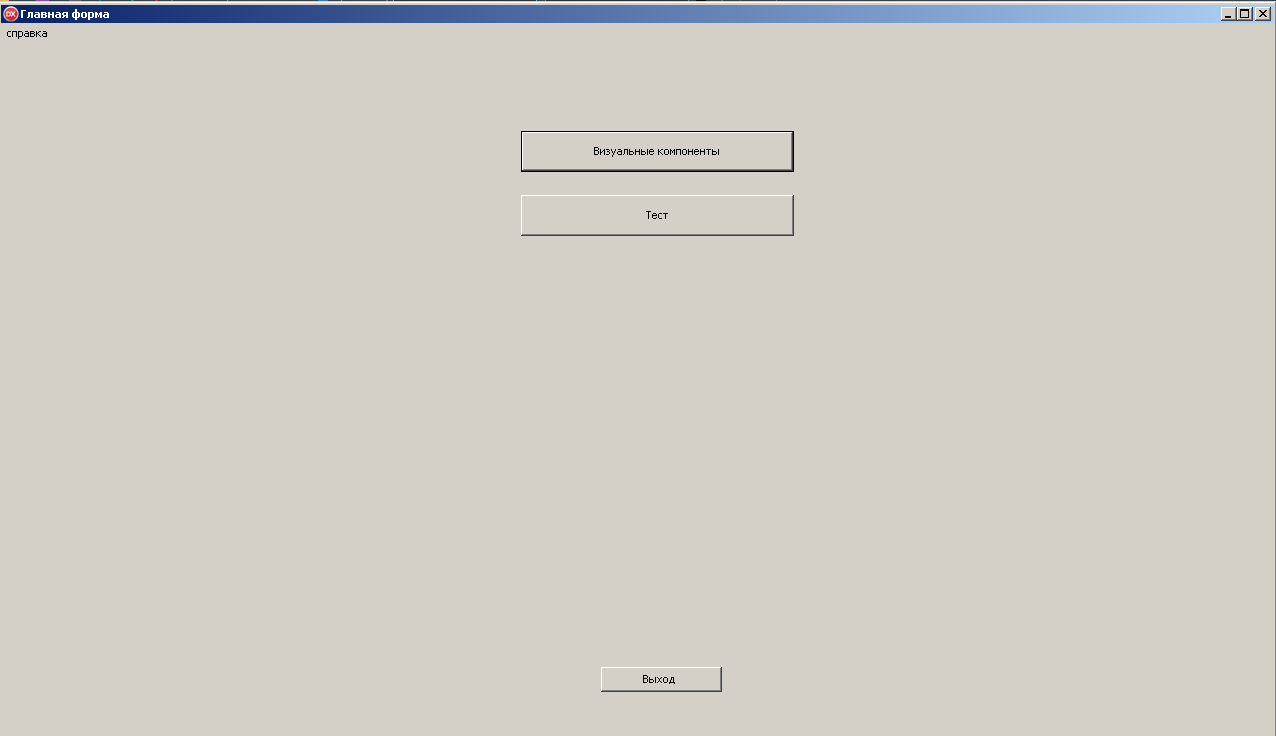


Рисунок 1 – Визуализация интерактивных возможностей приложения

Для организации эффективной работы пользователя создаётся целостное приложение данной предметной области, в котором все компоненты приложения сгруппированы по функциональному назначению, со всех второстепенных форм можно вернуться на главную (Рисунок 2). При этом же главной целью является удобный и приятный графический интерфейс пользователя.

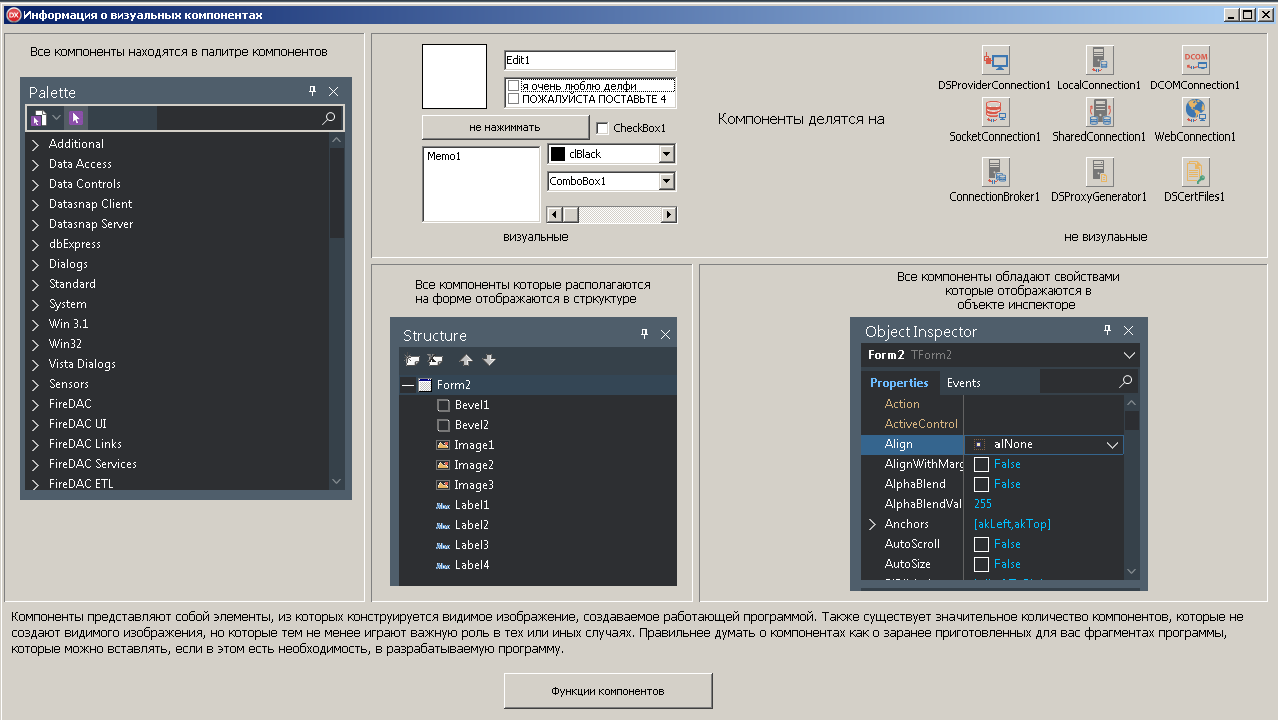


Рисунок 2 – Графический интерфейс программы

Таким образом, для успешной работы всего проекта реализован интуитивно понятный интерфейс и управление с приятной гаммой цветов и шрифтами.

Структура навигации по проекту представлена на рисунке 3.

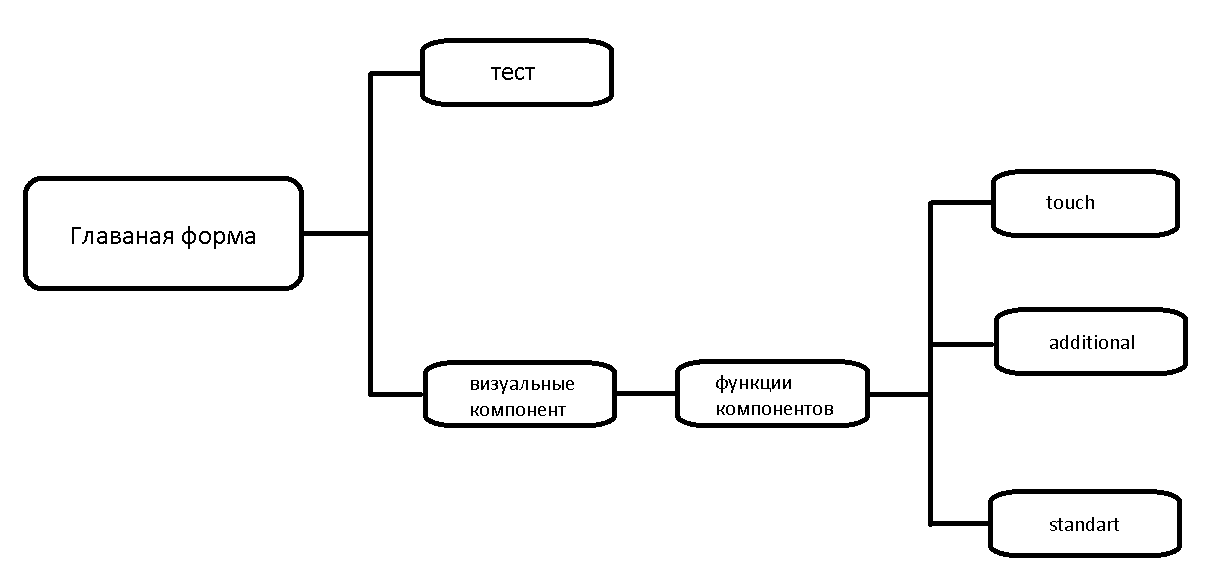


Рисунок 3 – Структура навигации по проекту

**Реализация**

**3.1 Реализация проекта**

**3.1.1 Структура программы**

Данный курсовой проект содержит 6 модулей. Далее рассмотрим назначение каждого модуля:

1 Модуль Unit1 – данный модуль является главной формой;

2 Модуль Unit2 – данный модуль является общей информацией об визуальных компонентах Delphi;

3 Модуль Unit3 – данный модуль отвечает за отображение трех групп компонентов;

4 Модуль Unit4 – представляет собой загрузочное меню;

5 Модуль Unit6 –данный модуль отвечает за отображение компонентов standart;

6 Модуль Unit8 – данный модуль отвечает за отображение компонентов touch;

7 Модуль Unit13 – данный модуль отвечает за отображение компонентов additional;

8 Модуль Unit16 – данный модуль предназначен для теста.

**3.1.2 Структура и описание процедур и функций пользователя**

Описание разработанных процедур находятся в таблице 1.

Таблица 1 – Процедуры и функции

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Имя процедуры (функции) | В каком модуле находится | За каким  компонентом закреплена | Назначение |
| 1 | 2 | 3 | 4 | 5 |
| 1 | procedure TForm1.Button1Click(Sender: TObject); | Unit1 | Button1 | Переход на следующую форму |
| 2 | procedure TForm1.Button4Click(Sender: TObject); | Unit1 | Button4 | Закрытие формы |
| 3 | procedure TForm1.N1Click(Sender: TObject); | Unit1 | MainMenu1 | Справочная система |
| 4 | procedure TForm6.Button1Click(Sender: TObject); | Unit6 | Button1 | Отоброжение номера выделеленного в данный момент компонента |
| 5 | procedure TForm6.Button2Click(Sender: TObject); | Unit6 | Button2 | Проверка, если на компоненте CheackBox1(CheckBox1.Checked), то программу можно закрывать |
| 6 | procedure TForm6.Button4Click(Sender: TObject); | Unit6 | Button4 | Удаление выбраной строки в ListBox |
| 7 | procedure TForm6.Button5Click(Sender: TObject); | Unit6 | Button5 | Полная очистка listbox |
| 8 | procedure TForm6.Button6Click(Sender: TObject); | Unit6 | Button6 | Добавление строки в ListBox |
| 9 | procedure TForm6.CheckBox2Click(Sender: TObject); | Unit6 | CheckBox2 | Отоброжает стоит ли флажок на компоненте CheckBox2 |
| 10 | procedure TForm6.ListBox1Click(Sender: TObject); | Unit6 | ListBox1 | Отоброжение нажатой строки в Edit |
| 11 | procedure TForm6.RadioGroup2Click(Sender: TObject); | Unit6 | RadioGroup2 | Получение номера выделенного в данный момент компонента |
| 12 | procedure TForm6.ScrollBar1Change(Sender: TObject); | Unit6 | ScrollBar1 | Отображает ширину прокрутки |
| 13 | procedure TForm8.Button10Click(Sender: TObject); | Unit8 | Button10 | Выделяет весь текст |
| 14 | procedure TForm8.Button1Click(Sender: TObject); | Unit8 | Button1Click | Открывает сохраненный текст |
| 15 | procedure TForm8.Button2Click(Sender: TObject); | Unit8 | Button2 | Сохранение текста |
| 16 | procedure TForm8.Button3Click(Sender: TObject); | Unit8 | Button3 | Копирует текст |
| 17 | procedure TForm8.Button4Click(Sender: TObject); | Unit8 | Button4 | Вырезает текст |
| 18 | procedure TForm8.Button5Click(Sender: TObject); | Unit8 | Button5 | Очистить все |
| 19 | procedure TForm8.Button6Click(Sender: TObject); | Unit8 | Button6 | Вставить текст |
| 20 | procedure TForm13.ColorBox1Change(Sender: TObject); | Unit13 | ColorBox1 | Изменение цвета формы |
| 21 | procedure TForm16.Button1Click(Sender: TObject); | Unit16 | Button1Click | Следующий вопрос |
| 22 | procedure TForm16.RadioGroup1Click(Sender: TObject); | Unit16 | RadioGroup1 | Выбор варианта |

**3.1.3 Описание использованных компонентов**

Описание использованных для разработки приложения компонентов приводится в таблице 2.

Таблица 2 – Использованные компоненты

|  |  |  |  |
| --- | --- | --- | --- |
| №  № | Компонент | На какой форме расположен | Назначение |
| 1 | TMainMenu | Form1 | Используется для создания  главного меню проекта |
| 2 | TButton | Form1,  Form2,  Form3,  Form6,  Form8,  Form13,  Form16 | Используется как кнопка, которая производит действие после нажатия |
| 3 | TTimer | Form4 | Используется для определения длительности  загрузочного экрана |
| 4 | TMainMenu | Form1 | Используется для отображения справочной системы |
| 5 | TLabel | Form2,  Form3,  Form6,  Form8,  Form13,  Form16 | Отображение надписей на  форме |
| 6 | TImage | Form2 | Отображение изображения |
| 7 | TTouchKeyboard | Form8 | Экранная клавиатура |
| 8 | TBevel | Form2,  Form3,  Form6,  Form8,  Form13,  Form16 | Отображение рамки |
| 9 | TShape | Form13 | Отоброжение простейших геометрических фигур |
| 10 | TBitBtn | Form13 | Кнопка Button, но с возможностью ставить графическое изображение |
| 11 | TGauge | Form4 | Отображает этап загрузки |
| 12 | TCheckBox | Form6 | Служит для представления пользователю элемента выбора — флажка |
| 13 | TMemo | Form8,  Form6,  Form2 | Предназначен для вывода на экран нескольких строк текста |
| 14 | TEdit | Form6 | Предназначен для вывода на экран одной строки текста |
| 15 | TComboBox | Form6 | Вариант списка с присоединенным дополнительным полем, в котором отображается выбранный элемент списка |
| 16 | TGroupBox | Form6 | Необходим для объединения нескольких компонентов |
| 17 | TListBox | Form6 | Применяется в программах для Windows |
| 18 | TRadioButton | Form6 | Интеpфейсный объект |
| 19 | TRadioGroup | Form6 | Управления с взаимосвязанными |
| 20 | TScrollBar | Form6 | Используется как вспомогательный инструмент в других элементах управления |
| 21 | TMaskEdit | Form13 | Используется как и компонент TEdit - для ввода и хранения строки текста |
| 22 | TColorBox | Form13 | Комбинированный список, опциями которого являются цвета |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в Таблице 3.

Таблица 3 – Спецификация программы

|  |  |  |
| --- | --- | --- |
| № | Имя файла | Назначение |
| 1 | Labyrinth.exe | Исполняемый файл проекта |
| 2 | Labyrinth.dpr | Файл проекта, связывает все файлы, из которых состоит приложение |
| 3 | Unit1.pas | Файл программного модуля для формы Form1 |
| 4 | Unit2.pas | Файл программного модуля для формы  Form2 |
| 5 | Unit3.pas | Файл программного модуля для формы  Form3 |
| 6 | Unit6.pas | Файл программного модуля для формы  Form6 |
| 7 | Unit8.pas | Файл программного модуля для формы Form8 |
| 8 | Unit13.pas | Файл программного модуля для формы Form13 |
| 9 | Unit16.pas | Файл программного модуля для формы Form16 |
| 10 | Unit4.pas | Файл программного модуля для формы Form4 |
| 11 | Labyrinth.exe | Установочный файл |
| 12 | Delphihelp.chm | Файл справки, содержит помощь по работе с программой |
| 13 | Unit1.dfm | Форма, предназначенная для отображения главной формы |
| 14 | Unit2.dfm | Форма на которой отображается информация о визуальных компонентах |
| 15 | Unit3.dfm | Форма на которой располагаются визуальные компоненты |
| 16 | Unit4.dfm | Форма загрузочного экрана |
| 17 | Unit6.dfm | Форма на которой отображаются компоненты standart |
| 18 | Unit8.dfm | Форма на которой отображаются компоненты touch |
| 19 | Unit13.dfm | Форма на которой отображаются компоненты additional |
| 20 | Unit16.dfm | Форма на которой расположен тест |

**4 Тестирование**

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчёт о результатах тестирования предоставлен в Таблице 4.

Таблица 4 – Отчёт результатах тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тест | Ожидаемый результат | Физический результат | Результат тестирования |
| 1 | Проверка входа пользователя в систему | Открытие главного окна | Открытие главного окна | Выполнено |
| 2 | Проверка кнопок переходов | Переход на следующие формы без задержек | Переход на следующие формы осуществляется без задержек | Выполнено |
| 3 | Проверка открытия справочной системы | Открытие справочной системы | Открытие справочной системы | Выполнено |
| 4 | Проверка корректного отображения текста | Отображение корректного текста | Отображение корректного текста | Выполнено |
| 5 | Проверка кнопок для редактирования текста | Текст редактируется | Текст редактируется | Выполнено |
| 6 | Проверка функциональности кнопок | Кнопки функционирует | Кнопки функционирует | Выполнено |
| 7 | Проверка компонентов | Все компоненты отображаются и функционируют | Все компоненты отображаются и функционируют | Выполнено |
| 8 | Проверка пункта меню «Выйти» | Закрытие программы | Закрытие программы | Выполнено |
| 9 | Проверка теста | Тест проходится нормально | Тест проходится нормально | Выполнено |

Когда я делал свой программный продукт, я столкнулся с рядом проблем, которые исправил. Например: я не знал как открыть справочную систему, но проблема решена, и пользователь теперь может просматривать справочную систему.

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

При использовании данного игрового приложения пользователь не столкнется с ошибками системы.

**5 Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Разрабатываемое электронное средство обученияявляется продуктом сферы образования. Созданное электронное средство обучения рассчитано на любого рода пользователей. Применить его смогут люди начинающие заниматься программированием, интересующиеся визуальным программированием.

Быстродействие любой программы во многом зависит от характеристик выбранного персонального компьютера: рабочей частоты процессора, объема оперативной памяти и т.д. Несмотря на все реализованные в ней задачи, она легко запускается и функционирует на любых машинах.

Тестирование проводилось на разных классах ЭВМ и работать с данной программой было комфортно. Программа разработана на ПК со следующими характеристиками:

* процессор Intel pentium;
* объем ОЗУ 4 гб;

- объем места на HDD – 512 гб ;

* Видеоадаптер NVIDIA GeForse 820M;
* ОС – Windows 7.

**5.2 Инсталляция**

Для того, чтобы установить программу необходимо запустить файл Setup.exe. Появится окно установки электронного средства обучения «Визуальные компоненты Delphi».

При нажатии на кнопку Далее появляется окно выбора каталога установки.

Следуем инструкции и устанавливаем приложение.

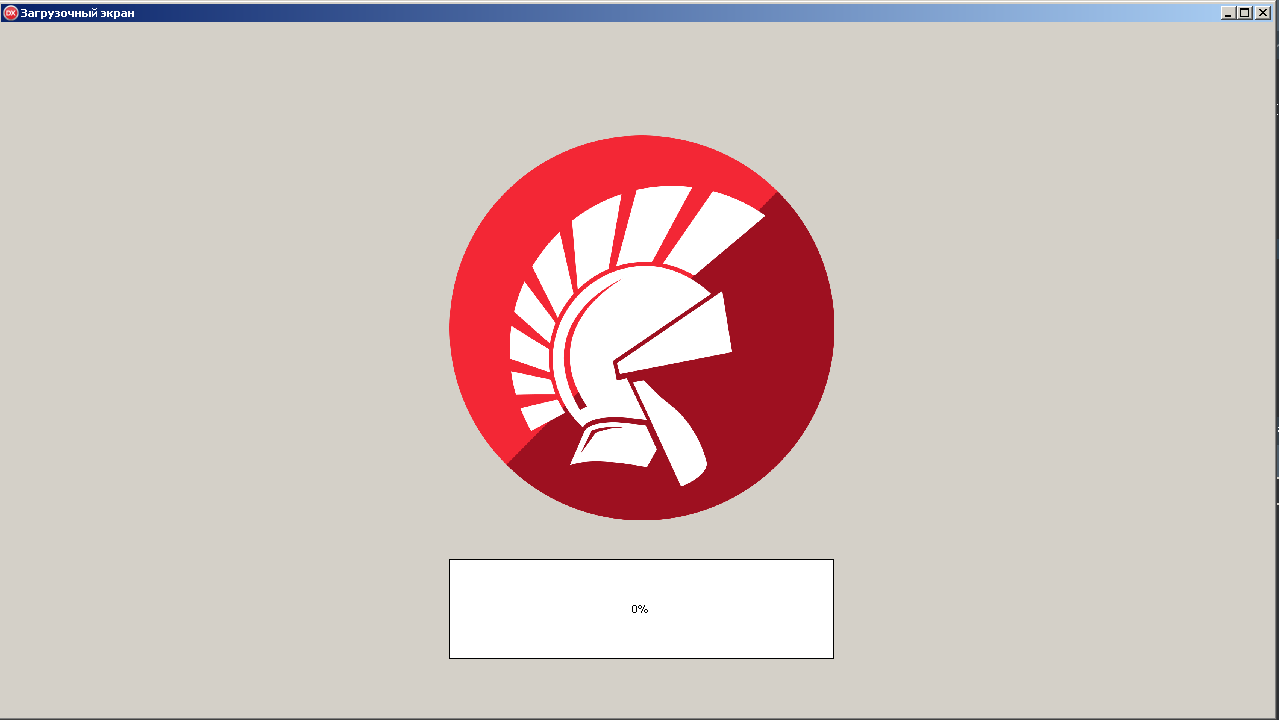
**5.3 Выполнение программы**

**5.3.1 Запуск программы**

Данную программу можно запустить с помощью ярлыка на рабочем столе. Необходимо дважды щелкнуть левой кнопкой мыши на ярлыке с названием Визуальные компоненты Delphi.

По подготовленным тестам будет осуществляться функциональное и полное тестирование программного продукта. Отчет о результатах тестирования будет представлен в 4 разделе пояснительной записки.

**5.3.2 Инструкции по работе с программой**



После запуска на экране появляется загрузочное меню. (Рисунок 4).

Рисунок 4 – Загрузочное меню

После окончания загрузки открывается главная форма. Где гость может открыть справочную систему или перейти на следующую форму нажав кнопку Визуальные компоненты Delphi. (Рисунок 5).

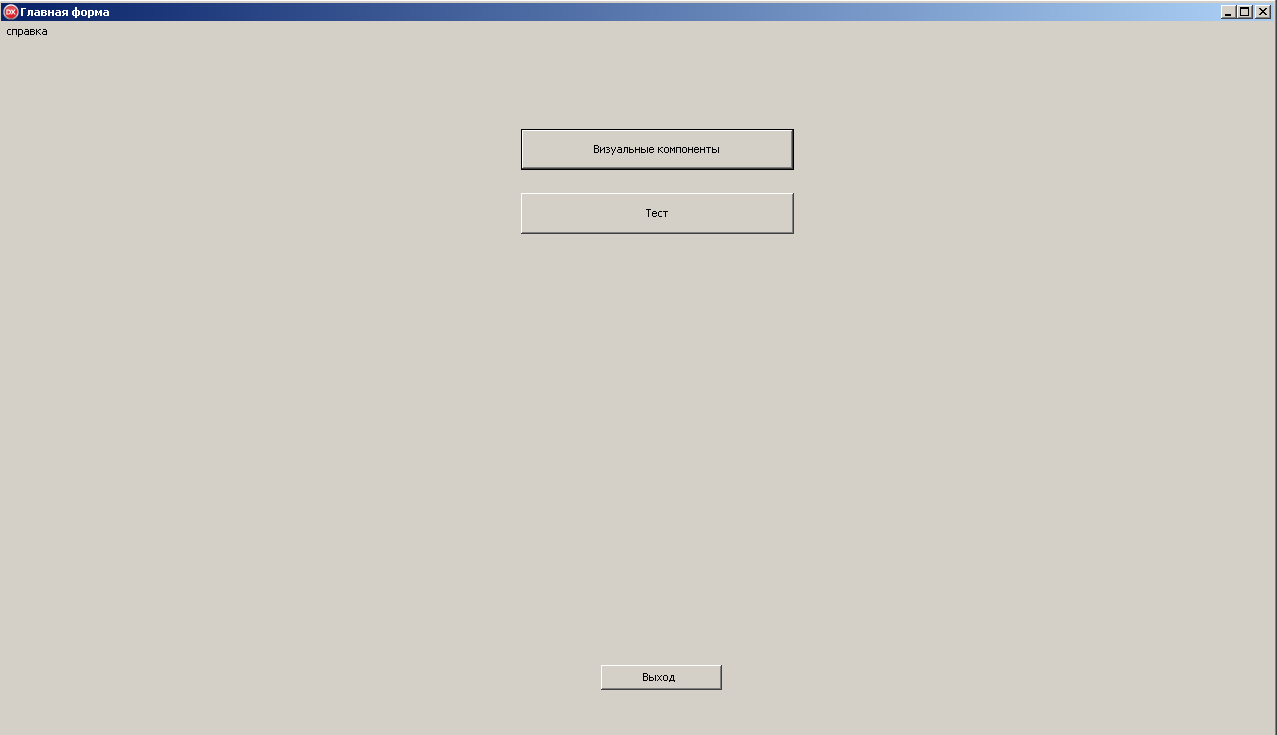


Рисунок 5 – Главная форма

При нажатии на кнопку «тест» будет осуществлён переход на форму с тестом. Где гость может пройти его. (Рисунок 6).

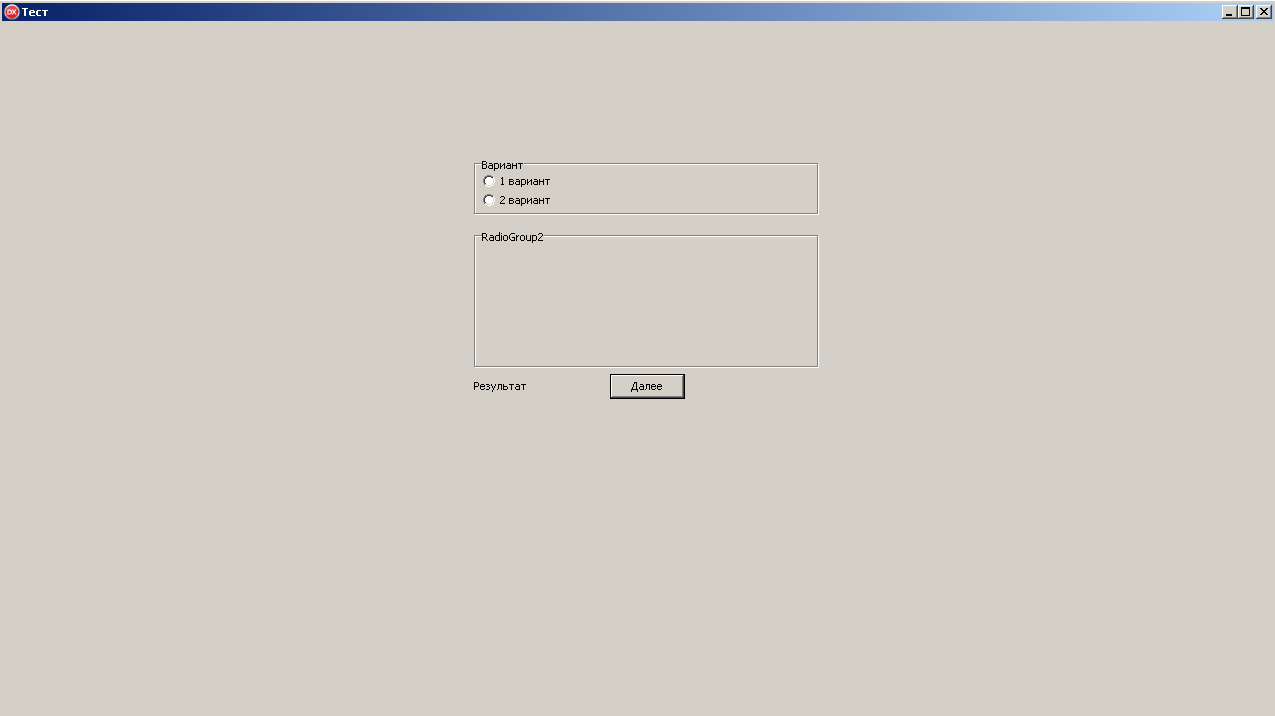


Рисунок 6 – Форма с тестом

При нажатии кнопки Визуальные компоненты гость перейдет на следующую форму. (Рисунок 7).

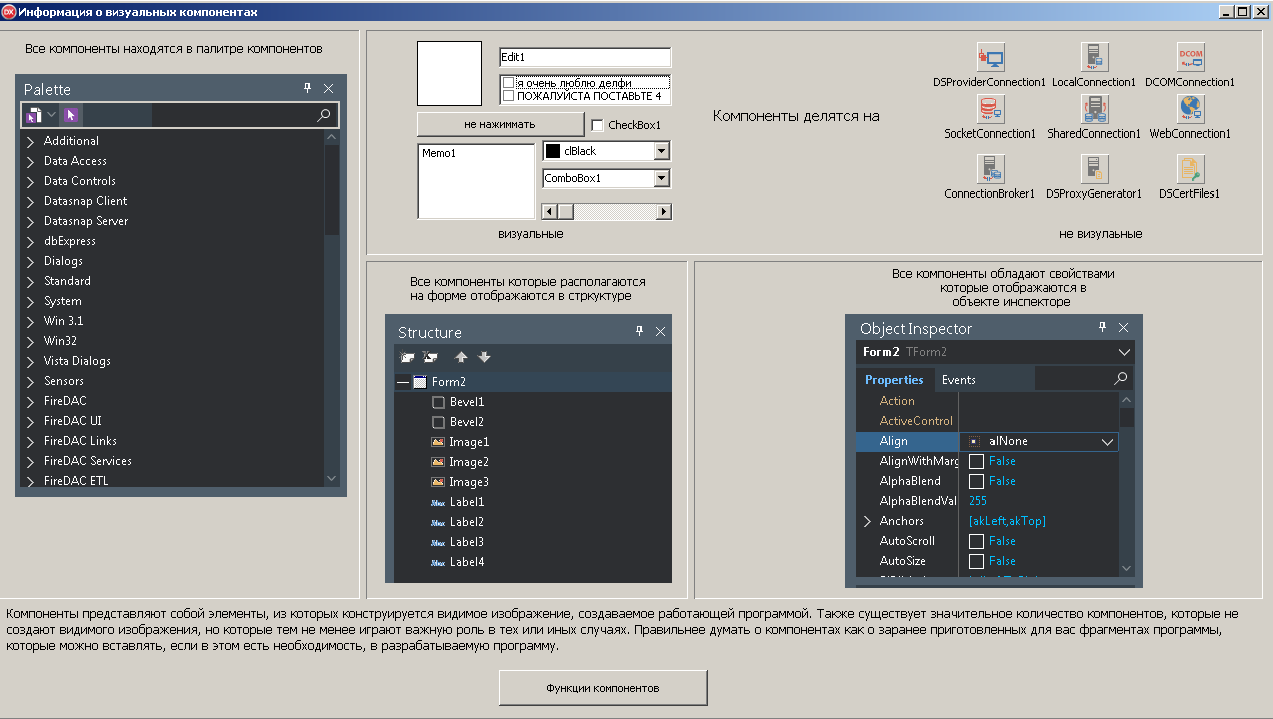


Рисунок 7 – Подтверждение регистрации

При нажатии кнопки Функции компонентов произойдет переход на форму Визуальные компоненты. Гость может выбрать любую из предложенных форму с компонентами. (Рисунок 8).

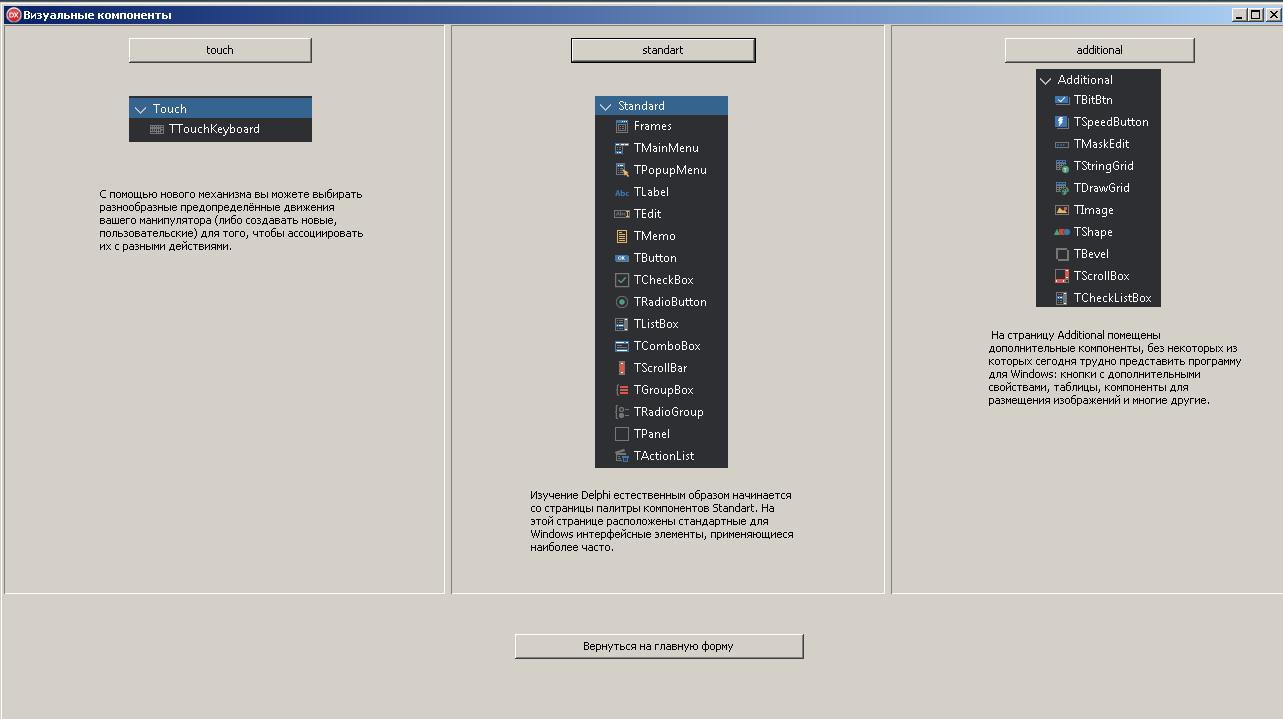


Рисунок 8 – Визуальные компоненты

При переходе на форму с компонентами touch, отобразиться редактор кода с которым можно повзаимодействовать и вернуться на главную форму, после чего перейти опять к форме с Визуальными компонетами. (Рисунок 9).



Рисунок 9 – Компоненты touch

При переходе на форму с компонентами standart, отобразиться множество компонетов которые выполняют различные функции, гость так же может почитать информацию о них и вернуться на главную форму, после чего перейти опять к форме с Визуальными компонетами. (Рисунок 10).

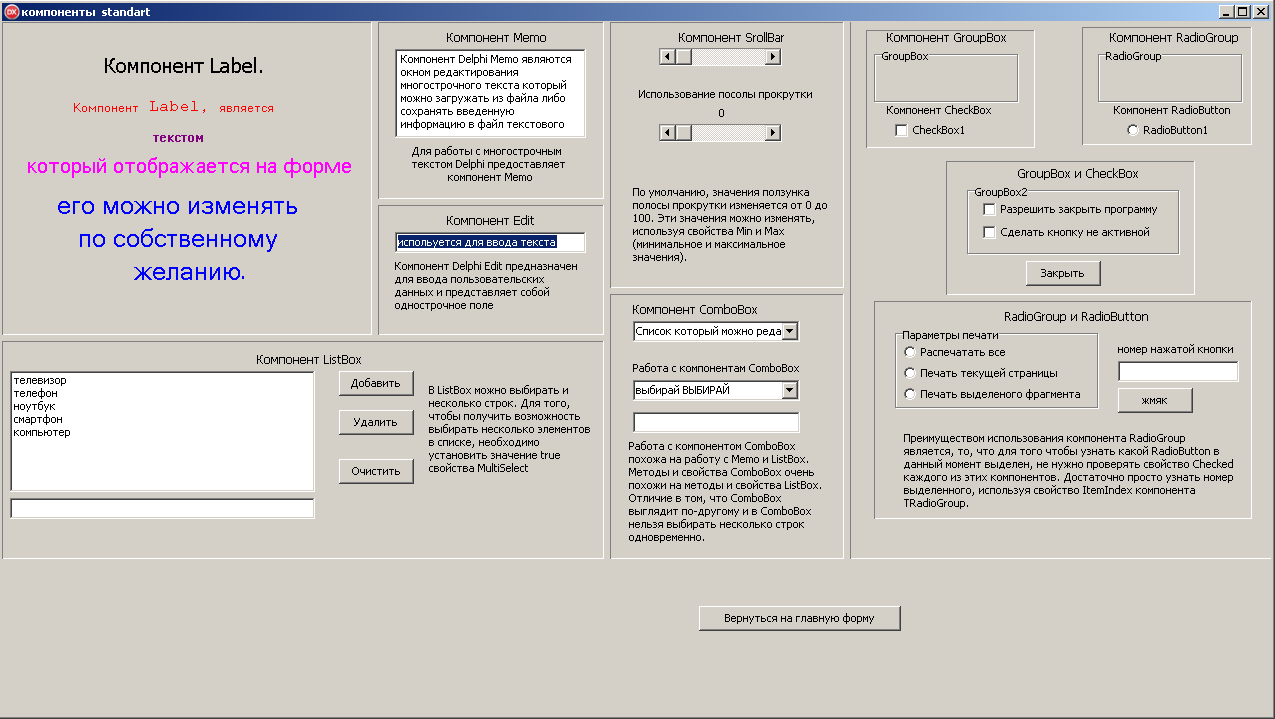


Рисунок 10 – Компоненты standart

При переходе на форму с компонентами additional, отобразиться множество компонетов которые выполняют различные функции, гость так же может почитать информацию о них и вернуться на главную форму, после чего перейти опять к форме с Визуальными компонетами. (Рисунок 11).

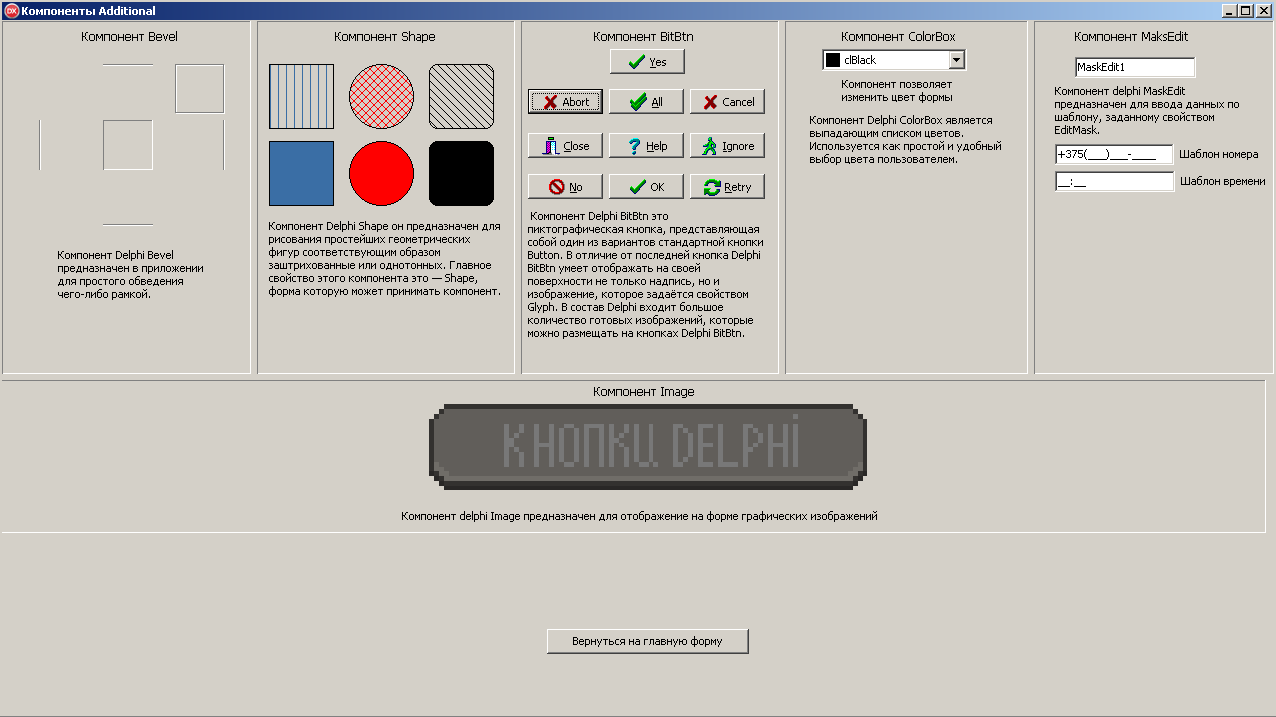


Рисунок 11 – Компоненты standart

На главной форме можно открыть справочную систему. (Рисунок 12).

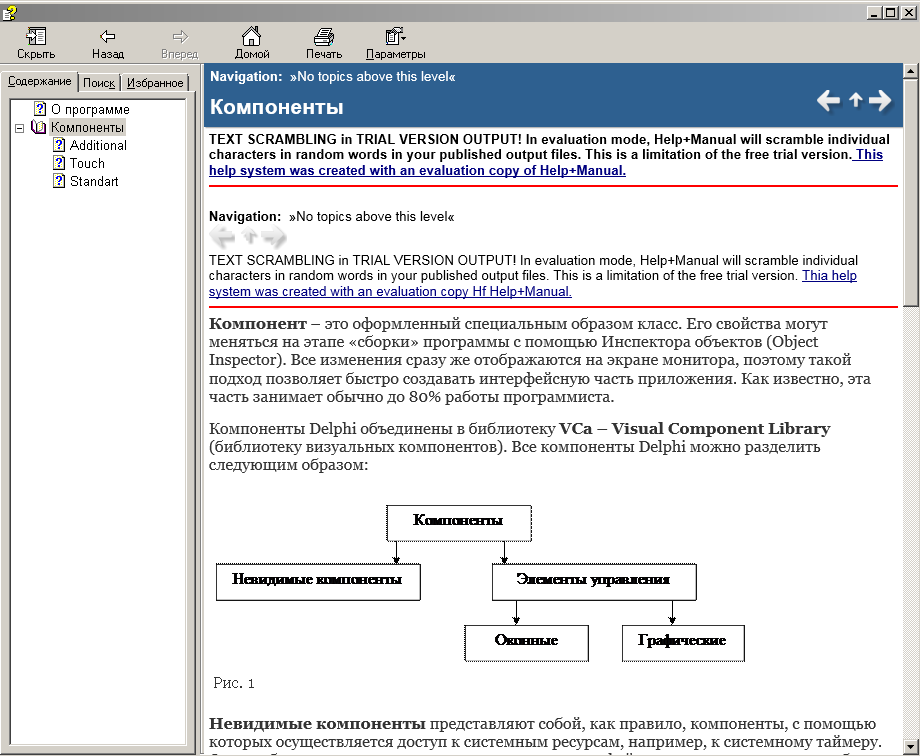


Рисунок 12 – Компоненты standart

**5.3.3 Завершение работы с программой**

Завершить работу с программой можно несколькими способами:

1. Нужно нажать на крестик в углу формы
2. Также можно закрыть игровое приложение нажав кнопку(Рисунок 13).

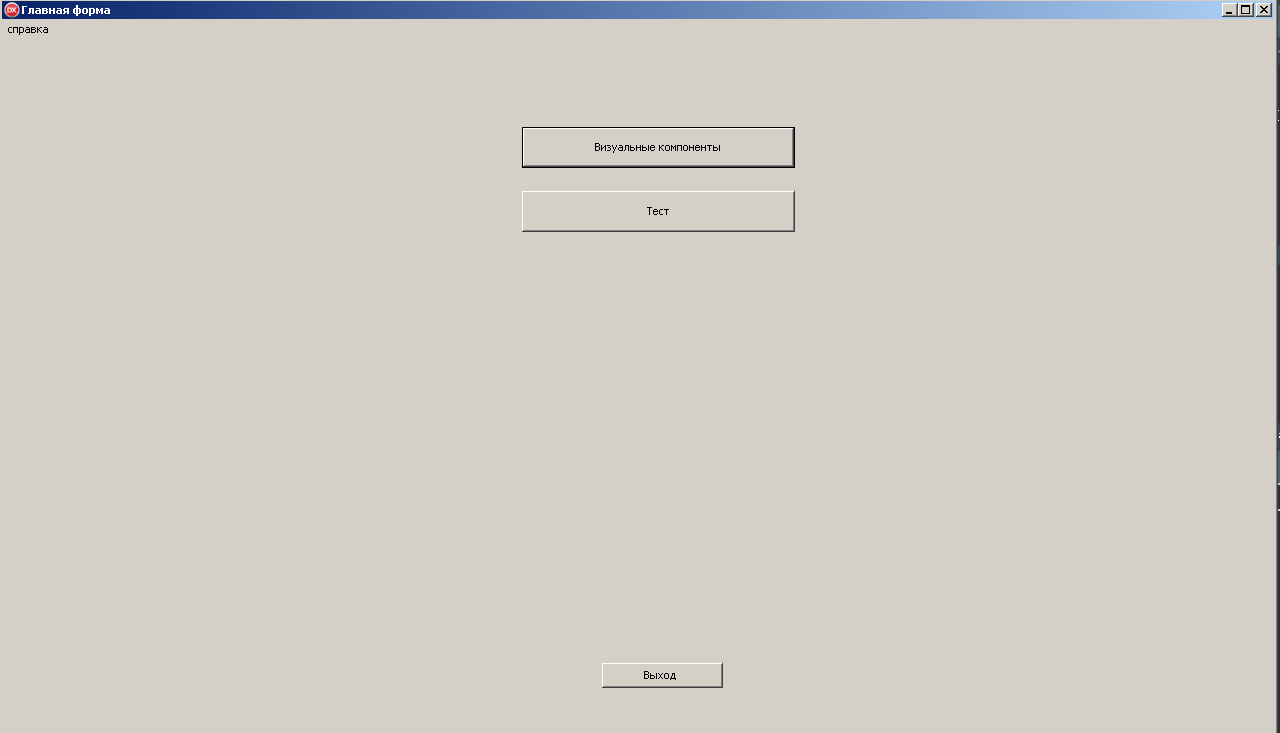


Рисунок 13 – Выход из приложения

**5.4 Использование системы справочной информации**

Справочную систему можно запустить с помощью пункта меню Справка или нажать на кнопку главного меню «Справка». (Рисунок 14).

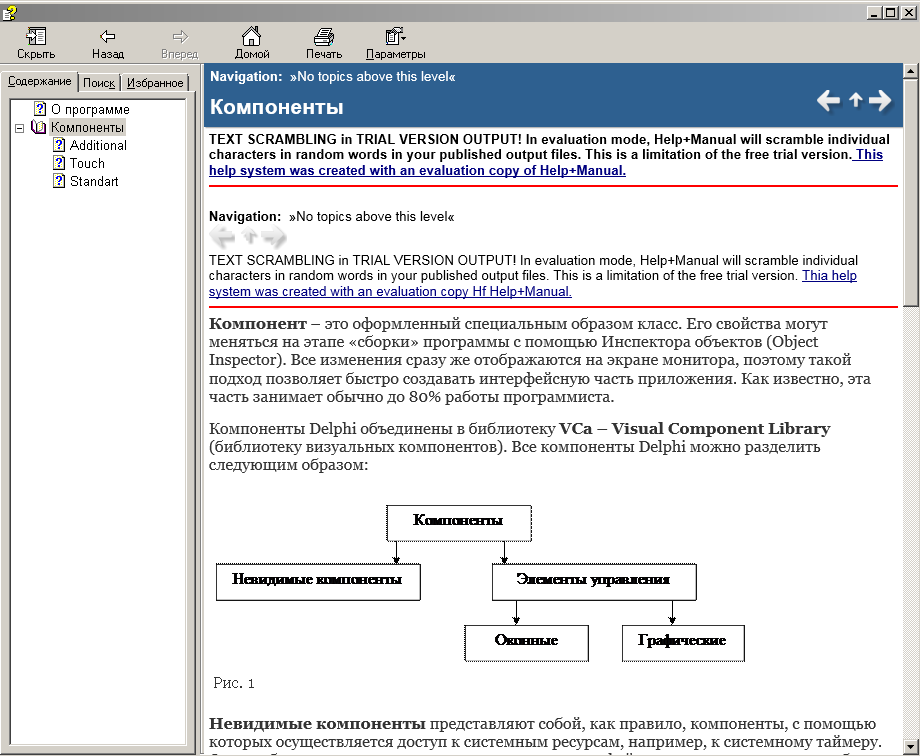


Рисунок 14 – Запуск справочной системы

## **Заключение**

Цель курсового проекта заключается в знакомстве массового пользователя с типами компонентов Delphi, визуальными компонентами Delphi, их видами, видимыми элементами пользовательского интерфейса, используемыми при разработке проектов.

В поставленной задаче был реализован простой, понятный, удобный, интересный, пользовательский интерфейс.

В ходе тестирования все исключительные ситуации были обработаны. Проект работает без сбоев и ошибок.

Продукт имеет дальнейшее развитие, так как на данный момент функционал недостаточно большой. Можно развить данный программный продукт, добавив новые компоненты, базу данных с компонентами. Несмотря на все это, данный программный продукт имеет законченный вид.

Исходя из этого, можно сделать вывод, что программа реализована успешно, а главное были приобретены новые знания в сфере создания электронных средств обучения.

**Список использованных источников**

1. Информация о всех компонентах и примерах [электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.h-delphi.ru>

2 . Информация о визуальных компонентах Delphi [электронный ресурс]. – Электронные данные. – Режим доступа: <https://studopedia.ru/13_47931_II-vizualnie-komponenti-Delphi.html>

3. Информация о компоненте touch [электронный ресурс]. – Электронные данные. – Режим доступа: <http://decoding.dax.ru/practic/TouchKeyboard/TouchKeyboard.html>

## **Приложение А**

## **Листинг программы**

unit Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, ShellAPI, Vcl.Menus;

type

TForm1 = class(TForm)

Button1: TButton;

Button3: TButton;

Button4: TButton;

MainMenu1: TMainMenu;

N1: TMenuItem;

procedure Button4Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure N1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

uses Unit2, Unit16;

procedure TForm1.Button1Click(Sender: TObject);

begin

Form1.Hide;

Form2.Show;

end;

procedure TForm1.Button3Click(Sender: TObject);

begin

Form16.Show;

end;

procedure TForm1.Button4Click(Sender: TObject);

begin

Form1.Close;

end;

procedure TForm1.N1Click(Sender: TObject);

begin

ShellExecute(0, PChar('Open'), PChar('Delphihelp.chm'), nil, nil, SW\_SHOW);

end;

end.

unit Unit2;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.pngimage,

Vcl.ExtCtrls, Vcl.CheckLst, Vcl.Buttons;

type

TForm2 = class(TForm)

Label1: TLabel;

Image1: TImage;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Image3: TImage;

Bevel1: TBevel;

Bevel2: TBevel;

Bevel3: TBevel;

Image4: TImage;

Label5: TLabel;

Bevel4: TBevel;

Label6: TLabel;

Shape1: TShape;

CheckListBox1: TCheckListBox;

Button1: TButton;

ColorBox1: TColorBox;

Memo1: TMemo;

CheckBox1: TCheckBox;

ScrollBar1: TScrollBar;

ComboBox1: TComboBox;

Edit1: TEdit;

Image2: TImage;

Button8: TButton;

Label7: TLabel;

procedure Button8Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.dfm}

uses Unit3;

procedure TForm2.Button8Click(Sender: TObject);

begin

Form3.Show;

Form2.Hide;

end;

end.

unit Unit4;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.StdCtrls, Vcl.MPlayer,

Vcl.Imaging.pngimage, Vcl.Samples.Gauges;

type

TForm4 = class(TForm)

Image1: TImage;

Gauge1: TGauge;

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

implementation

{$R \*.dfm}

end.

unit Unit6;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.StdCtrls, Vcl.Buttons;

type

TForm6 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

Memo1: TMemo;

RadioButton1: TRadioButton;

ComboBox1: TComboBox;

GroupBox1: TGroupBox;

RadioGroup1: TRadioGroup;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Bevel1: TBevel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

CheckBox5: TCheckBox;

Label10: TLabel;

Button2: TButton;

Bevel3: TBevel;

Bevel4: TBevel;

GroupBox2: TGroupBox;

CheckBox1: TCheckBox;

CheckBox2: TCheckBox;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Bevel5: TBevel;

Bevel6: TBevel;

RadioGroup2: TRadioGroup;

Label14: TLabel;

Label15: TLabel;

Edit2: TEdit;

Button1: TButton;

Label16: TLabel;

Bevel7: TBevel;

ScrollBar1: TScrollBar;

Label17: TLabel;

Label18: TLabel;

ScrollBar3: TScrollBar;

Label19: TLabel;

Edit3: TEdit;

ComboBox2: TComboBox;

Label20: TLabel;

Bevel8: TBevel;

Bevel9: TBevel;

Bevel10: TBevel;

Bevel11: TBevel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

Label24: TLabel;

Bevel2: TBevel;

ListBox1: TListBox;

Edit4: TEdit;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Label25: TLabel;

Label26: TLabel;

Button3: TButton;

procedure Button2Click(Sender: TObject);

procedure CheckBox2Click(Sender: TObject);

procedure RadioGroup2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure ScrollBar1Change(Sender: TObject);

procedure ComboBox2Change(Sender: TObject);

procedure ListBox1Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form6: TForm6;

implementation

{$R \*.dfm}

uses Unit2, Unit1, Unit3;

procedure TForm6.Button1Click(Sender: TObject);

//при нажатии на кнопку в компоненте edit2 отоброжался номер выделеленного в данный момент компонента

begin

Edit2.Text:=IntToStr(RadioGroup2.ItemIndex);

end;

procedure TForm6.Button2Click(Sender: TObject);

begin

//выполняется проверка, если на компоненте CheackBox1(CheckBox1.Checked), то программу можно закрывать и она закроется, если же нет, то появится сообщение о том, что программу закрывать нельзя

if CheckBox1.Checked then

Close

else

ShowMessage('Нельзя закрыть программу');

end;

procedure TForm6.Button3Click(Sender: TObject);

begin

Form1.Show;

Form6.Close;

end;

procedure TForm6.Button4Click(Sender: TObject);

//удаление выбраной строки в ListBox

begin

ListBox1.Items.Delete(ListBox1.ItemIndex);

end;

procedure TForm6.Button5Click(Sender: TObject);

//кнопка для полной очистки listbox

begin

ListBox1.Clear;

end;

procedure TForm6.Button6Click(Sender: TObject);

//добавление строки в ListBox с помощья компонента Edit

begin

ListBox1.Items.Add(Edit4.Text);

end;

procedure TForm6.Button8Click(Sender: TObject);

begin

Form6.Close;

end;

procedure TForm6.CheckBox2Click(Sender: TObject);

//свойство Checked показывает, стоит ли флажок на компоненте CheckBox2, или нет

begin

Button2.Enabled := not CheckBox2.Checked;

end;

procedure TForm6.ComboBox2Change(Sender: TObject);

//вывод в edit

begin

Edit3.Text := ComboBox2.Items.Strings[ComboBox2.ItemIndex];

end;

procedure TForm6.ListBox1Click(Sender: TObject);

//когда нажимаем на какую-нибудь из строк в listbox, то она высветится в edit

begin

Edit4.Text := ListBox1.Items.Strings[ListBox1.ItemIndex];

end;

procedure TForm6.RadioGroup2Click(Sender: TObject);

//обращаяся к свойству ItemIndex компонента Radiogroup, мы получаем номер выделенного в данный момент компонента, необходимо преобразовать целое число в строку использую функцию IntToStr

begin

Label15.Caption:=IntToStr(RadioGroup2.ItemIndex);

end;

procedure TForm6.ScrollBar1Change(Sender: TObject);

//событие change позволяющие отображать ширину прокрутки Scrollbar

begin

Label18.Caption := IntToStr(ScrollBar1.Position);

end;

end.

unit Unit3;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.pngimage,

Vcl.ExtCtrls, Vcl.Buttons;

type

TForm3 = class(TForm)

Button2: TButton;

Button3: TButton;

Button10: TButton;

Image2: TImage;

Image9: TImage;

Image10: TImage;

Bevel2: TBevel;

Bevel3: TBevel;

Button1: TButton;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Bevel1: TBevel;

procedure Button5Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button10Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

{$R \*.dfm}

uses Unit4, Unit5, Unit12, Unit9, Unit15, Unit6, Unit13, Unit7, Unit10, Unit11,

Unit8, Unit1, Unit2;

procedure TForm3.Button10Click(Sender: TObject);

begin

Form8.Show;

Form3.Hide;

end;

procedure TForm3.Button1Click(Sender: TObject);

begin

Form1.Show;

Form3.Close;

end;

procedure TForm3.Button2Click(Sender: TObject);

begin

Form6.Show;

Form3.Hide;

end;

procedure TForm3.Button3Click(Sender: TObject);

begin

Form13.Show;

Form3.Hide;

end;

procedure TForm3.Button5Click(Sender: TObject);

begin

Form4.Show;

Form3.Hide;

end;

procedure TForm3.Button6Click(Sender: TObject);

begin

Form10.Show;

Form3.Hide;

end;

procedure TForm3.Button7Click(Sender: TObject);

begin

Form5.Show;

Form3.Hide;

end;

end.

unit Unit8;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Touch.Keyboard, Vcl.StdCtrls,

Vcl.Buttons;

type

TForm8 = class(TForm)

TouchKeyboard1: TTouchKeyboard;

Label1: TLabel;

Memo1: TMemo;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Button10: TButton;

Label2: TLabel;

Button7: TButton;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button10Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure Button9Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form8: TForm8;

implementation

{$R \*.dfm}

uses Unit1, Unit3;

procedure TForm8.Button10Click(Sender: TObject);

begin

// Кнопка "Выделить все"

// Если компонент не в фокусе} то выделение не увидим

Memo1.SetFocus;

Memo1.SelectAll;

end;

procedure TForm8.Button1Click(Sender: TObject);

begin

Memo1.Lines.LoadFromFile('mytetxt.txt'); // Кнопка"Открыть

end;

procedure TForm8.Button2Click(Sender: TObject);

begin

Memo1.Lines.SaveToFile('mytetxt.txt'); // Кнопка"Сохранить

end;

procedure TForm8.Button3Click(Sender: TObject);

begin

Memo1.CopyToClipboard; // Кнопка "Копировать"

end;

procedure TForm8.Button4Click(Sender: TObject);

begin

Memo1.CutToClipboard; // Кнопка "Вырезать"

end;

procedure TForm8.Button5Click(Sender: TObject);

begin

Memo1.Clear; // Кнопка "Очистить все"

end;

procedure TForm8.Button6Click(Sender: TObject);

begin

Memo1.PasteFromClipboard; // Кнопка "Вставить"

end;

procedure TForm8.Button7Click(Sender: TObject);

begin

Form1.Show;

Form8.Close;

end;

procedure TForm8.Button8Click(Sender: TObject);

begin

Form3.Show;

Form8.Close;

end;

procedure TForm8.Button9Click(Sender: TObject);

begin

Form1.Show;

Form8.Close;

end;

end.

unit Unit13;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.Tabs, Vcl.DockTabSet,

Vcl.ButtonGroup, Vcl.CategoryButtons, Vcl.StdCtrls, Vcl.ValEdit, Vcl.CheckLst,

Vcl.Grids, Vcl.Mask, Vcl.Buttons, Vcl.Imaging.pngimage;

type

TForm13 = class(TForm)

BitBtn1: TBitBtn;

MaskEdit1: TMaskEdit;

Image1: TImage;

Shape1: TShape;

Bevel1: TBevel;

ColorBox1: TColorBox;

Label1: TLabel;

Bevel2: TBevel;

Bevel3: TBevel;

Bevel4: TBevel;

Bevel5: TBevel;

Bevel6: TBevel;

Bevel7: TBevel;

Bevel8: TBevel;

Label2: TLabel;

Shape2: TShape;

Shape3: TShape;

Label3: TLabel;

Label4: TLabel;

Bevel10: TBevel;

Shape4: TShape;

Shape5: TShape;

Shape6: TShape;

Label5: TLabel;

Bevel11: TBevel;

BitBtn2: TBitBtn;

BitBtn3: TBitBtn;

BitBtn4: TBitBtn;

BitBtn5: TBitBtn;

BitBtn6: TBitBtn;

BitBtn7: TBitBtn;

BitBtn8: TBitBtn;

BitBtn9: TBitBtn;

BitBtn10: TBitBtn;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

MaskEdit2: TMaskEdit;

Label10: TLabel;

Label11: TLabel;

MaskEdit3: TMaskEdit;

Bevel12: TBevel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Bevel9: TBevel;

Bevel13: TBevel;

Button3: TButton;

procedure ColorBox1Change(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form13: TForm13;

implementation

{$R \*.dfm}

uses Unit1, Unit3;

procedure TForm13.Button3Click(Sender: TObject);

begin

Form1.Show;

Form13.Close;

end;

procedure TForm13.ColorBox1Change(Sender: TObject);

//изменение цвета формы

begin

form13.Color:=ColorBox1.Selected;

end;

end.

unit Unit16;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls;

type

TForm16 = class(TForm)

RadioGroup1: TRadioGroup;

Button1: TButton;

RadioGroup2: TRadioGroup;

Label1: TLabel;

procedure RadioGroup1Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form16: TForm16; f: text; s: string; Nvern, ball: integer;

implementation

{$R \*.dfm}

procedure TForm16.Button1Click(Sender: TObject);

begin

//Если выбран вариант ответа и не достигнут конец файла

if (RadioGroup2.ItemIndex>-1) and (not Eof(f)) then begin

if RadioGroup2.ItemIndex = Nvern-1 then ball:=ball+1; //Если выбранный вариант соответствует

RadioGroup2.Items.Clear; //номеру верного ответа то балл прибавляется

Repeat //и очищается поле для следующего вопроса

if (s[1]='-') then begin

delete(s,1,1);

RadioGroup2.Caption:=s;

end

else if s[1]='\*' then begin

delete(s,1,1);

Nvern:=StrToInt(s);

end

else RadioGroup2.Items.Add(s);

readln(f,s);

Label1.Caption:=s;

until (s[1]='-') or (Eof(f));

end

//Если конец файла достигнут, значит вопросы закончились

Else if Eof(f) then begin

delete(s,1,1);

Nvern:=StrToInt(s);

if RadioGroup2.ItemIndex = Nvern-1 then ball:=ball+1;

Label1.Caption:=IntToStr(ball); //Вывод количества баллов

CloseFile(f);

Button1.Enabled:=False; //кнопка становится недоступной в завершении проекта

end;

end;

procedure TForm16.RadioGroup1Click(Sender: TObject);

begin

RadioGroup1.Enabled:=false; //Выбор варианта становится недоступен

RadioGroup2.Enabled:=true; //Доступным становится поле с вопросом

Button1.Enabled:=true; //Кнопка Далее

case RadioGroup1.ItemIndex of //В зависимости от выбранного варианта переменная f

0: AssignFile( f ,'variant1.txt');//связывается с разными файлами

1: AssignFile( f ,'variant1.txt');

end;

reset(f); //Открываем файл для чтения

readln(f,s); //Считываем первую строку из файла

ball:=0; //изначально количество баллов 0

repeat

if (s[1]='-') then begin //Если первый символ строки ‘-‘ значит это вопрос

delete(s,1,1);

RadioGroup2.Caption:=s;

end

else if s[1]='\*' then begin //Если перв символ ‘\*’ значит это номер верного ответа

delete(s,1,1);

Nvern:=StrToInt(s);

end

else RadioGroup2.Items.Add(s); //Иначе это вариант ответа

readln(f,s); //Считываем следующую строку из файла

until (s[1]='-') or (Eof(f)); //Считывание и отправление вариантов ответов в RadiGroup до тех пор

// пока не достигнут следующий вопрос или конец файла

end;

end.