**Введение**

На курсовое проектирование была поставлена задача, на тему: «Разработка квест-игры «Собери Беларусь в своём сердце».

Цель курсового проекта активизация познавательных навыков через познавательную деятельность.

Создаваемое приложение будет рассчитано для учащихся которые проявляют интерес к изучению родного края.

Приведём 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 задачи.

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 “Списке используемых источников” будет приведен список используемой литературы, нормативно- техническую и другую документацию.

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

Схема работы системы будет представлена в графической части.

**1. Анализ задачи**

**1.1 Постановка задачи**

Темой данного курсового проекта является разработка квест-игры «Собери Беларусь в своём сердце»

Периодичность использования данного программного продукта неограниченна. Пользователь может в любое время установить, а также удалить приложение после полного изучения материала по данным темам и прохождения теста.

Цель данного курсового проекта – разработать программный продукт, который будет содержать в себе элементы знаний о родном крае.

За основу программы возьмем из соответствующих источников (сайтов, статей) наиболее полную, достоверную и точную информацию. При реализации проекта необходимо его сделать простым и максимально доступным для пользователей.

Квест-игра должна располагать своим интерфейсом и функционалом к пользователю, чтобы ему было комфортно.

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.

Разрабатываемый программный продукт должен позволять выполнять следующие действия:

* обеспечить быстрый поиск необходимой информации;
* прохождение теста для закрепления знаний;
* просмотр справки;
* просмотр практических заданий;

В разрабатываемой программе будут использоваться три вида данных: входная, выходная и постоянная информация.

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, картинки и др.

Программный продукт предоставляет функционал для следующего ряд пользователей: администратор – организовывает работу программой, гость.

**1.2 Инструменты разработки**

Для разработки данного проекта будет выбрана среда Delphi 11, так как

это самое удобная и доступная среда разработки на данный момент. Delphi 11-язык –программирования, относящийся к классу RAD- (Rapid Application Development – «Средство быстрой разработки приложений») средств CASE –тeхнологии. Delphi 11 сделал разработку приложений дляWindows быстрым иприятным процессом. Теперь разрабатывать сложные и интересные проекты можно только одним человеком, использующим Delphi 11

Интерфейс Windows обеспечивает полное перенесение CASE-технологии в интегральную систему поддержки работ по созданию прикладной системы на всех фазах жизненного цикла работы и проектирования системы.

Delphi 11 обладает широким набором возможностей, начиная от проектировщика форм и заканчивая поддержкой всех форматов популярных баз данных. Среда устраняет необходимость программировать такие компоненты Windows общего назначения, как метки, программы и даже диалоговые панели. Работая в Windows, можно видеть одинаковые «объекты» во многих разнообразных приложениях. Диалоговые панели (например, Choose File и Save File) являются примерами многократно-используемых компонентов, встроенных непосредственно в Delphi 11, который позволяет приспособить эти компоненты к имеющийся задаче, чтобы они работали именно так, как требуется создаваемому приложению. Также здесь имеются предварительно-определенные визуальные и не визуальные объекты.

Три основные части разработки интерфейса следующие: проектирование панели, проектирование диалога и представление окон. Для общего пользовательского доступа также должны учитываться условия применения архитектуры прикладных систем.

Сегодня появилась реальная возможность с помощью моделирования на современных многофункциональных средствах обработки и отображения информации таких как Delphi 11 конкретизировать тип и характеристики используемых информационных моделей, выявить основные особенности будущей деятельности операторов, сформулировать требования к параметрам аппаратно-программных средств интерфейса взаимодействия и т.д. Delphi 11 позволяет создать различные виды программ: консольные приложения, оконные приложения, приложения для работы с Интернетом и базами данных. То есть, Delphi 11 является не только средствами для работы с языком программирования Паскаль, но дополнительные инструменты, призванные для максимального упрощения и ускорения создание приложений.

К дополнительным инструментам можно отнести визуальный редактор форм, благодаря которому можно с легкостью создать полноценную программу, и другие визуальные составляющие разработки программного обеспечения. С Delphi вам не нужно вручную просчитывать расположение каждого элемента интерфейса пользователя, поэтому при разработке программы значительно экономится время.

Выгоды от проектирования в среде Windows с помощью Delphi 11:

* устраняется необходимость в повторном вводе данных;
* обеспечивается согласованность проекта и его реализации;
* увеличивается производительность разработки и переносимость программ.

Ни одно серьезное программное обеспечение не обходится без модуля справочной информации и руководства пользователя. Это придает программе законченный вид и показывает заботу о пользователе. DrExplain – легкий в использовании и функциональный инструмент, упрощающий создание справочных файлов Windows, печать справочных руководств и документации в целом. Программа имеет интуитивно понятный интерфейс. Все созданные проекты можно сохранить в различных форматах:HTML Help, Winhelp и MS Help 2.0 / Visual Studio Help, Browser-based Help, PDF и Word RTF, а также печатной документации при использовании одного и того же проекта. В основном окне программы содержатся оглавление (в виде древовидного списка) и текстовый редактор. Это дает возможность легко ориентироваться в оглавлении, редактировать или перемещать разделы справки без каких-либо проблем. Также утилита позволяет конвертировать help-файлы из одного формата в другой. Кроме приложений для работы с текстом в данном продукте содержатся утилиты для создания скриншотов и редактирования графических файлов.

Для создания инсталлятора будет использоваться мощное и удобное средство - Smart Install Maker. Программа обладает удобным и интуитивно понятным интерфейсом, а также полным набором необходимых функций для создания профессиональных инсталляторов с минимальным размером, высокой степенью сжатия файлов и приятным интерфейсом.

Помимо стандартного минимума, Smart Install Maker позволяет редактировать системный реестр и INI-файлы, создавать программные ярлыки, запускать ассоциируемые и исполняемые файлы, регистрировать новые шрифты и ActiveX компоненты, отображать тексты информации и лицензионного соглашения. Также, с помощью этой утилиты, можно создать мультиязыковые инсталляторы с поддержкой более 20-ти популярных языков мира.

Easy Paint tool SAI - это популярная среди художников программа для рисования (в частности для проработки интерфейса и игры «одевалка»). Удобна своей простотой и в то же время большим функционалом. Программа очень сильно распространена в использовании во всем мире и предпочитается рисующими людьми как один из самых удобных инструментов для создания рисунков. В нем можно сохранить файлы картинок во множестве вариантов (.png, .jpg, .bmp и так далее).

Microsoft Word 2019 – редактор текста для написания документации.

Разработка ведется на ноутбуке HP . У данного ноутбука следующие

параметры:

* процессор Ryzen 7;
* объем ОЗУ 16 гб;
* объем места на HDD – 512 гб ;
* видеоподсистема 1024х768 точек с глубиной цвета 16 Bit;
* ОС – Windows 10 домашняя.

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

**1.3 Требования к приложению**

На этапе исследования предметной области был установлен целый ряд требований, которые предъявляются к разрабатываемой программе.

По этой причине особенно актуальной становится разработка программного электронного продукта, который способствовал бы и давал возможность вспомнить и повторить информацию из теоретического курса. Чтобы эта информация лучше усваивалась необходимо подобрать правильный интерфейс.

Требования к интерфейсу: в связи с частым использованием программы она должна быть с приятной цветовой гаммой и понятной для пользователя. Следовательно, каждое окно должно иметь ясную визуальную иерархию своих элементов. Фрагменты текста должны располагаться на экране так, чтобы пользователя было просто и понятно принимать информацию.

Пользователь не должен испытывать какого-либо дискомфорта в плане восприятия информация, отображённой на экране. Объекты (рисунки и символы) не должны быть слишком мелкие. Все окна приложения по возможности должны помещаться на экран полностью, так как использование в процессе работы полос прокруток достаточно неудобно.

На одной форме нельзя допускать избытка и нагромождения данных. Формы должны быть эффектно оформлены согласно тематике разрабатываемого проекта.

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

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой через графический интерфейс не должны влиять на конечный результат.

**2. 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. Подходящей структурой данных для хранения элементов является: динамический одномерный массив.

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 приложение. Главной задачей будет являться: повторение изученного материала и подготовка к квалификационному экзамену.

Для реализации задач будут использоваться процедуры. С помощью процедуры будет осуществляться переход на текстовый файл в формате .docx.

Материал и тест будет генерироваться на основе выбора пункта в объекте TRadioGroup. Далее после того, как будет прочитан теоретический материал и выполнена практическая часть, можно будет пройти тест для проверки своих знаний и получить оценку своих знаний.

**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 разработает данную программу, какие компоненты будут использованы и какие методы будут автоматизированы. Во-первых, особое внимание следует уделить интерфейсу. Разработчик должен так организовать внешний вид своей программы, чтобы пользователь понял, что от него требуется.

Для организации эффективной работы пользователя нужно создать целостное приложение данной предметной области, в которой все компоненты приложения будут сгруппированы по функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс пользователя. Приложение должно позволить пользователю решать задачи, затрачивая значительно меньше усилий, чем при работе с разрозненными объектами.

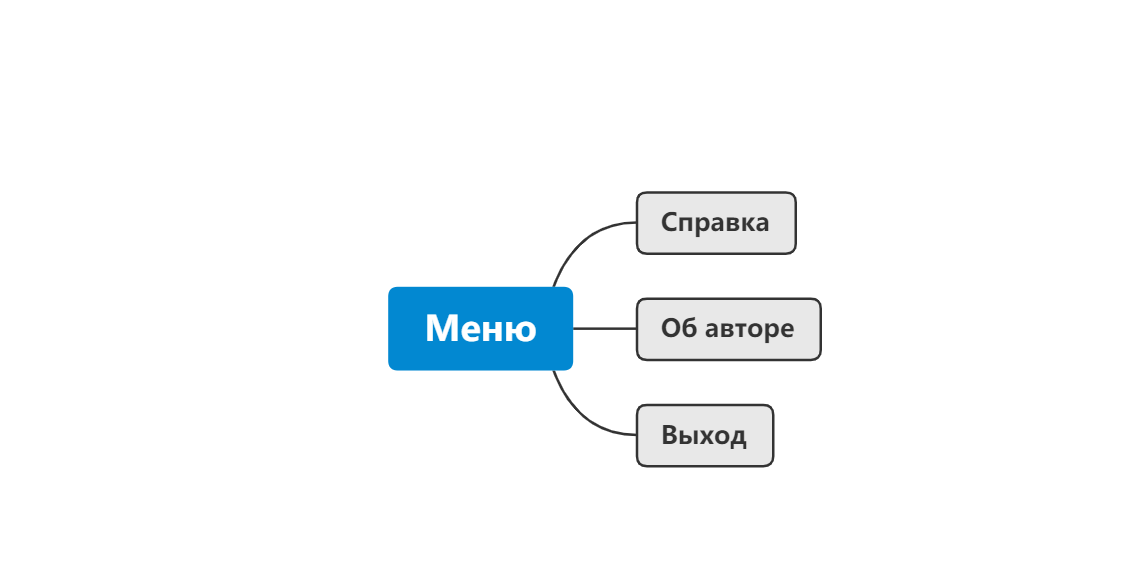
Ниже на рисунке 1 представлена система меню и организация навигации между окнами программы:

Рисунок 1 – Схема навигация между окнами программы

**3. Реализация**

**3.1 Реализация проекта**

Данный курсовой проект содержит 9 модулей. Далее рассмотрим назначение каждого модуля:

Unit 3- является заставкой в курсовом проекте;

Unit 1- содержит авторизацию для прохождения квест-игры;

Unit 2- содержит практические задания ;

Unit 6- «Правда или ложь»;

Unit 7-«Ребус»

Unit 16-«Блиц-опрос»;

Unit 18-«Кто я?»

Unit 29-«Угадай мелодию»

Unit 30-содержит результаты всех заданий.

**3.1.1 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 |
| 1 procedure  TForm3.Button1Click(Sender: TObject); | Unit 3 | Button | | Переход на следующую форму |
| 2 procedure TForm3.N2Click(Sender: TObject); | Unit 3 | Main | | Переход на форму “об авторе” |
| 3 procedure TForm3.N3Click(Sender: TObject); | Unit 3 | Main | | Закрывает квест-игру |
| 4 procedure TForm1. Button1Click (Sender: TObject); | Unit 1 | Button | | Переход на форму с авторизацией |
| 5 procedure TForm2.Image1Click(Sender: TObject); | Unit 2 | TImage | Переход на форму  «Правда или ложь» | |
| 6 procedure TForm2.Image4Click(Sender: TObject); (Sender: TObject); | Unit 2 | TImage | Переход на форму «Ребус» | |
| 7 procedure TForm2.Image5Click(Sender: TObject); | Unit 2 | TImage | Переход на форму «Блиц-опрос» | |
| 8 procedure TForm2.Image5Click(Sender: TObject); (Sender: TObject); | Unit 2 | TImage | Переход на форму «Кто я?» | |

|  |  |  |  |
| --- | --- | --- | --- |
| Продолжение таблицы 1 | | | |
| 1 | 2 | 3 | 4 |
| 9 procedure TForm2.Image6Click(Sender: TObject); (Sender: TObject); | Unit 2 | TImage | Переход на форму «Угадай мелодию» |
| 10 procedure TForm2.Button1Click(Sender: TObject); | Unit 6 | Button | Переход на следующее задание |
| 11 procedure TForm7.Button1Click(Sender: TObject); | Unit 7 | Button | Переход на следующее задание |
| 12 procedure TForm16.Button1Click(Sender: TObject); | Unit 16 | Button | Переход на следующее задание |
| 13 procedure TForm16.Button2Click(Sender: TObject); | Unit 16 | Button | Переход на форму с заданиями |
| 14 procedure TForm18.Button1Click(Sender: TObject); | Unit 18 | Button | Переход на следующее задание |
| 15 procedure TForm29.Button1Click(Sender: TObject); | Unit 29 | Button | Переход на следующее задание и на форму с результатми |

**3.1.2 Описание использованных компонентов**

Описание использованных для разработки приложения компонентов приводится в таблице 2.

Таблица 2- Использованные компоненты

|  |  |  |
| --- | --- | --- |
| Компонент | На какой форме  расположен | Назначение |
| TMainMenu | Form3; | Используется для создания главного меню проекта |
| TImage | Form3, Form1, Form5  Form6 ,Form7, Form16, Form18, Form29, Form30 | Используется как фон |
| TLabel | Form3, Form1, Form5  Form6 ,Form7, Form16, Form18, Form29, Form30 | Отображение надписей на форме |
| TButton | Form2, Form3, Form4, Form5, Form6, Form7 | Используется для взаимодействия  пользователя с программой: для начала теста, закрытие программы, перехода на другую формы и т.д. |
| MediaPlayer | Form29 | Используется для хранения теоретического материала |
| TRadioGroup | Form16,29 | Используется для выбора варианта ответа в тесте |
|  |  |  |
| Продолжение таблицы 2 | | |
| 1 | 2 | 3 |
| TEdit | Form 7, Form18 | Используется для ввода информации пользователя |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

|  |  |
| --- | --- |
| Имя файла | Назначение |
| 1 | 2 |
| Project3.exe | Исполняемый файл проекта, используется для запуска программы на выполнение. |
| Project3.dproj. | Файл проекта, связывает все файлы из которых состоит приложение. |
| Unit3.pas | Файл программного модуля главной формы |
| Unit1.pas | Файл программного модуля с авторизацией |
| Unit2.pas | Файл программного модуля заданий |
| Unit6.pas | Файл программного модуля задания «Правда или ложь» |
| Unit7.pas | Файл программного модуля задания «Ребус» |
| Unit16.pas | Файл программного модуля задания «Блиц-опрос» |
| Unit18. pas | Файл программного модуля задания «Кто я?» |
| Unit29. pas | Файл программного модуля задания «Угадай мелодию» |
| Unit30. pas | Файл программного модуля c результатами |
| Unit3.dfm | Форма с авторизацией |
| Unit1.dfm | Форма с описанием программы |
| Unit2.dfm | Форма с тестом |
| Unit7.dfm | Тест по повторенному материалу | |
| Unit16.dfm | Установочный пакет программы | |
| Unit18.dfm | Теоретический материал | |
| Unit30.dfm | Практический материал | |
| 1-20.jpg | Изображение | |

**4. Тестирование**

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

Отчет о результатах тестирования представлен в таблице 4

Таблица 4- Отчет о результатах тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Идентификатор теста | Тест | Ожидаемый результат | Физический результат | Результат тестирование |
| 1 | Проверка кнопки «Начать» | Переход на форму с авторизацией | Осуществляется переход на форму с авторизацией | Выполнено |
| 2 | Проверка кнопки-  «Далее» | Переход к следующему  заданию, | Осуществляется переход к следующему заданию | Выполнено |
| 3 | Проверка кнопки «Правда» | Переход на следующий вопрос | Осуществляется переход на следующий вопрос | Выполнено |
| 4 | Проверка кнопки  «Ложь» | Переход на следующий вопрос | Осуществляется переход на следующий вопрос | Выполнено |
| 6 | Проверка пунктов выбора варианта ответа в тесте | Открытие формы с теоретическим материалом | Открытие формы с теоретическим материалом | Выполнено |
| 7 | Проверка мелодии | При нажатии на вариант ответа включается определённая мелодия | Осуществляется вариант ответа и включается мелодия | Выполнено |
| 8 | Проверка главного меню на кнопку «Закрыть» | Закрытие квест-игры | Осуществляется закрытие квест-игры | Выполнено |
| 9 | Проверка главного меня на кнопку «Описание программы» | Открытие формы с описанием программы | Открытие формы с описанием программы | Выполнено |
| 10 | Проверка перехода по картинке | При нажатии на картинку осуществляется переход на другую форму | Осуществляется при нажатии на картинку переход на другую форму | Выполнено |

При разработке программного продукта было решено множество проблем, например, не осуществлялся переход к следующему изображению по нажатию стрелки или открывался доступ к последнему квесту, когда был пройден только первый, поэтому в будущем пользователь не столкнется с данными проблемами.

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

**5 Применение**

**5.1 Общие сведения о программном продукте**

Цель данного проекта заключается в обобщении знаний и подготовки к квалификационному экзамену.

Создаваемое приложение будет рассчитано на учащихся, проходивших профессиональную подготовку рабочих по профессии «Оператор ЭВМ ».

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 AMD Ryzen 7 4700U with Radeon Graphics, 2000 МГц, ядер: 8, логических процессоров: 8
* объем ОЗУ 32Гб;
* графический адаптер AMD Radeon Graphics Processor (0x1636), Advanced Micro Devices, Inc.-совместимый;
* операционная система Windows 10.

**5.2 Инсталляция**

Для того, чтобы установить программу необходимо запустить файл Установщик.exe. Появится окно установки приложения “ Электронное средство обучения для учащихся профессиональной подготовки рабочих по профессии «Оператор ЭВМ ”.

Затем достаточно следовать приведенной инструкции установки приложения.

**5.3 Выполнение программы**

**5.3.1 Запуск программы**

Данную программу можно запустить различными способами. Первым из них является запуск с помощью ярлыка на рабочем столе. Необходимо дважды щелкнуть левой кнопкой мыши на ярлыке с названием «Собери Беларусь в своём сердцем. Вторым способом является запуск из каталога, в который устанавливалось приложение (по умолчанию C:\Program Files (x86)\ Квест-игра «Собери Беларусь в своём сердце»

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**

При запуске квест-игры Пользователя встречает главное меню.

Рисунок 2 – Заставка

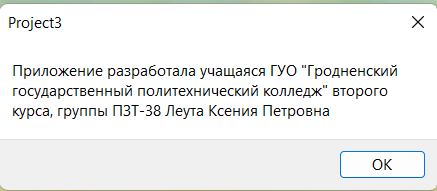
На главном экране находятся 1 основная кнопка: "Начать. Помимо этой кнопки есть 2 вкладки: "Справка" и "Выход". Выбрав вкладку "Справка", пользователь увидит кнопки "Об авторе" и "О приложении". Нажав кнопку "О приложнии" пользователю представится справка о приложении, которую Вы сейчас читаете. Нажав кнопку "Об авторе" откроется окно в котором вы знакомитесь с разработчиком квест-игры

Рисунок 3-Об авторе

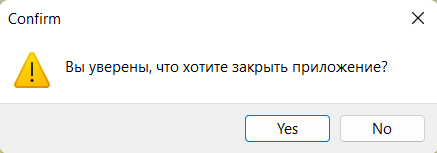
При нажатии кнопки "Выход" появится диалоговое окно с уточнением, действительно ли Вы хотите закрыть приложение.

Рисунок 4- Выход

При нажатии кнопки "Начать" производится переход на следующую форму с авторизацией.

Рисунок 5-Авторизация

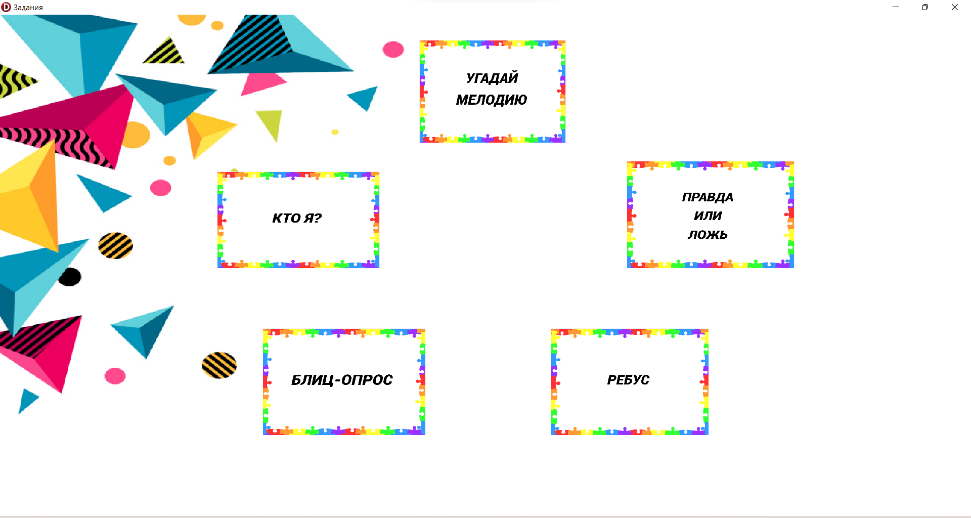
****При нажатии кнопки «Далее» производится переход на форму с выбором заданий.

Рисунок 6-Форма с заданиями

При нажатии на определённую картинку будет открываться форма с определёнными заданиями.

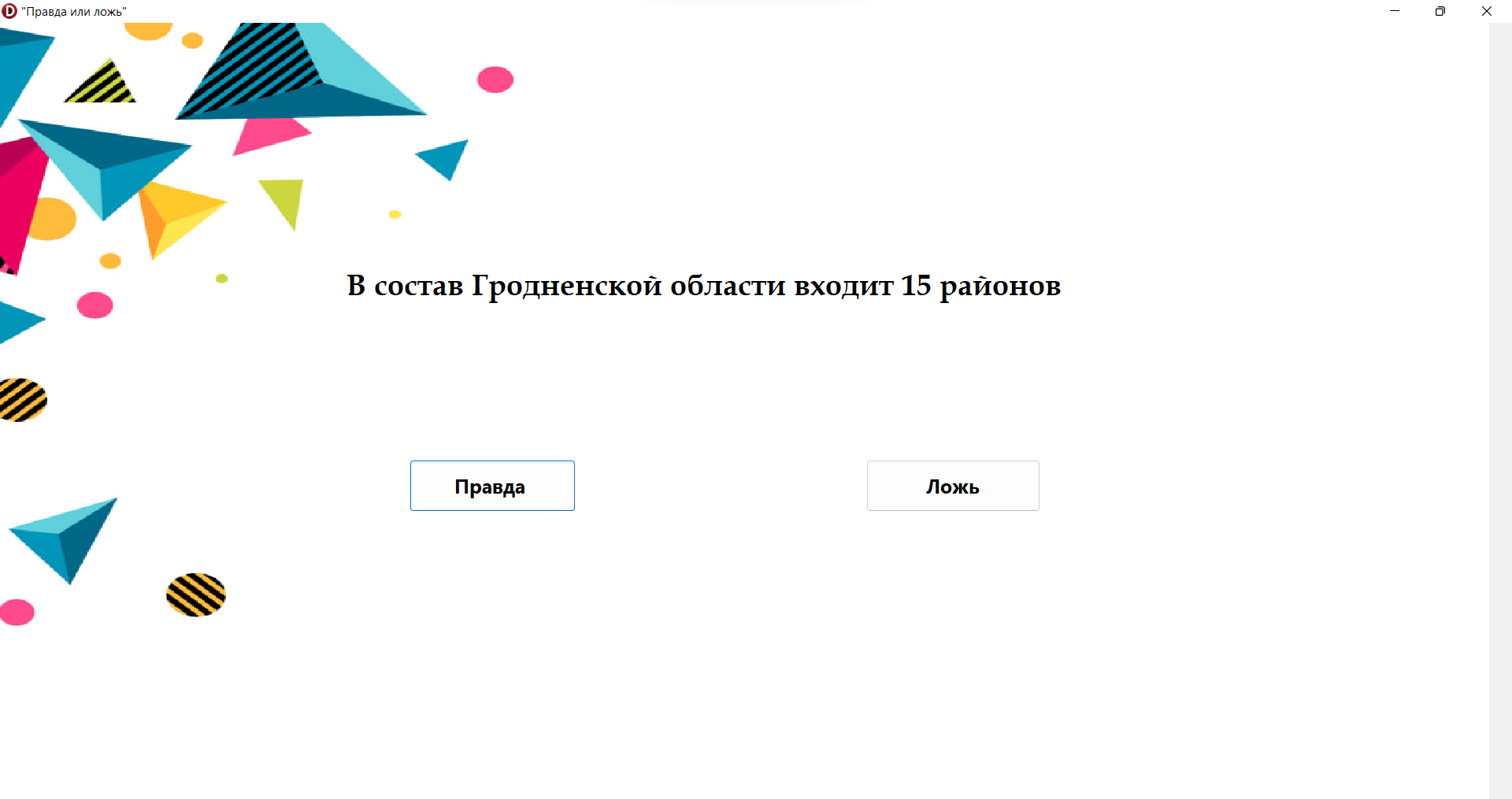
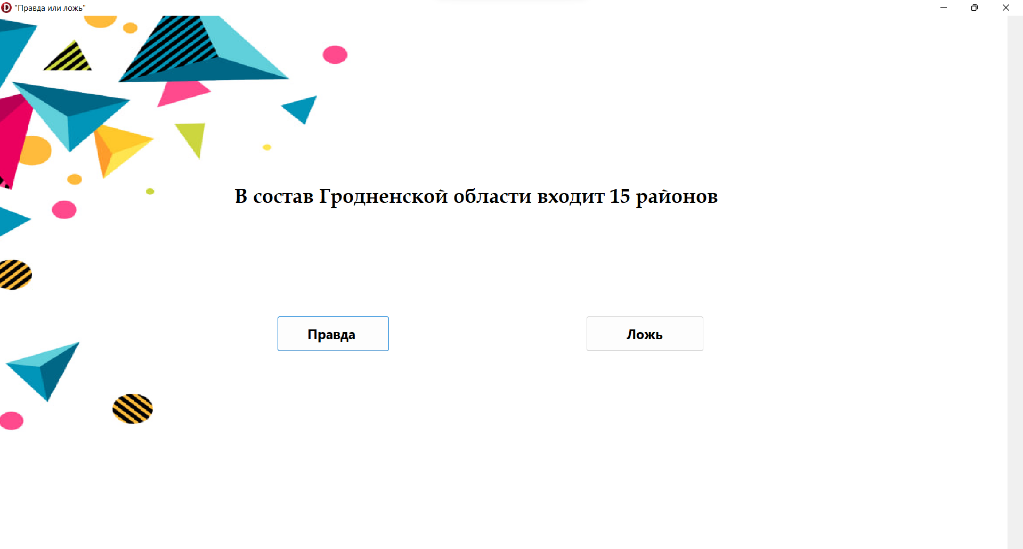


Рисунок 7-Форма с заданием «Правда или ложь»

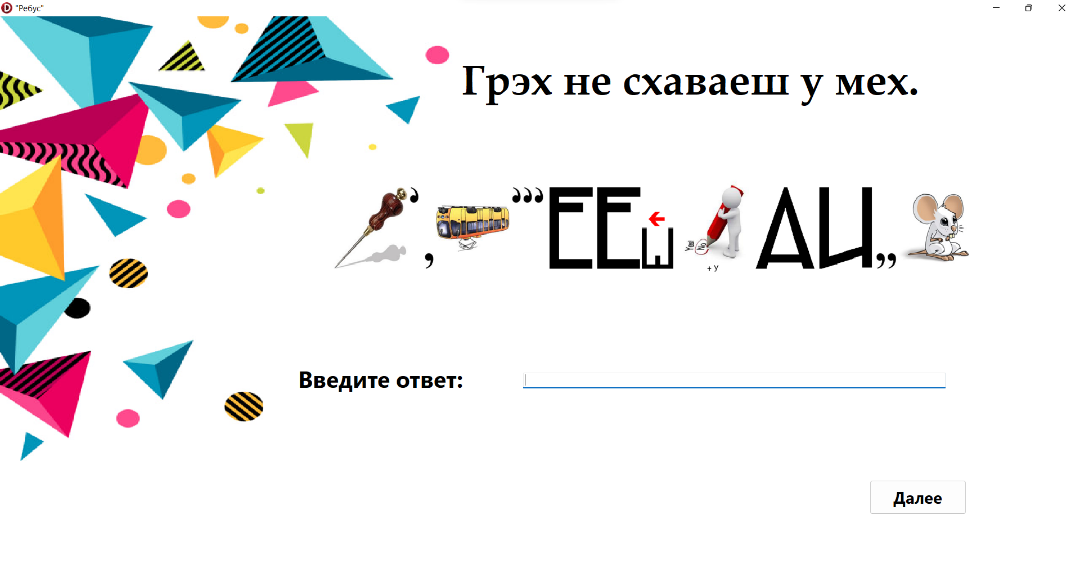
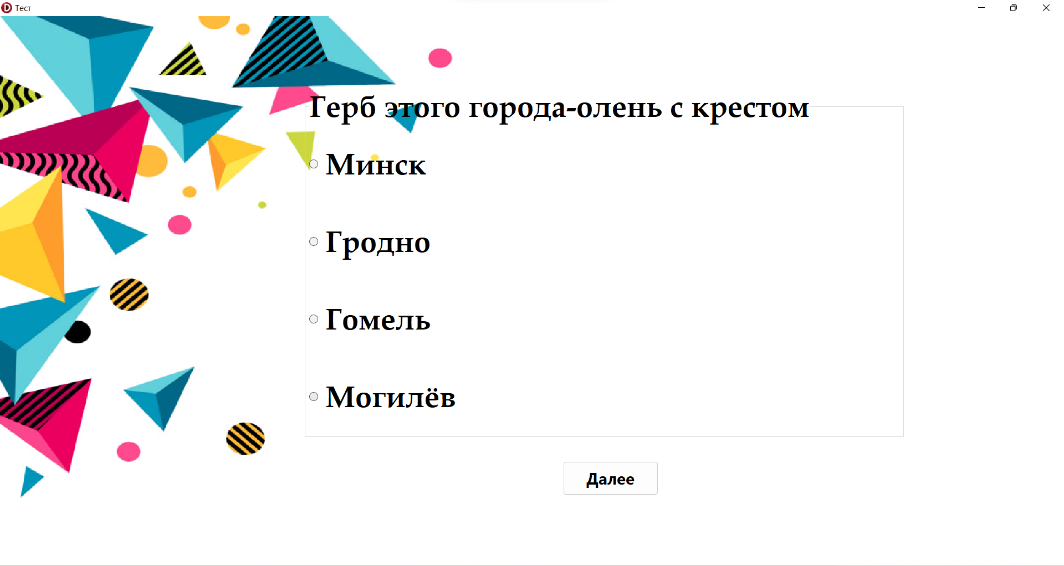
При нажатии на копку «правда» или «ложь» будет осуществляться переход на следующее задание в форме и переход на форму с выбором заданий.

Рисунок 8-Форма с заданием «Ребус»

При вводе ответа и нажатии кнопки «Далее» будет осуществляться переход на следующее задание в этой форме и переход на форму с выбором заданий.

Рисунок 9-Форма с заданием «Блиц-опрос»

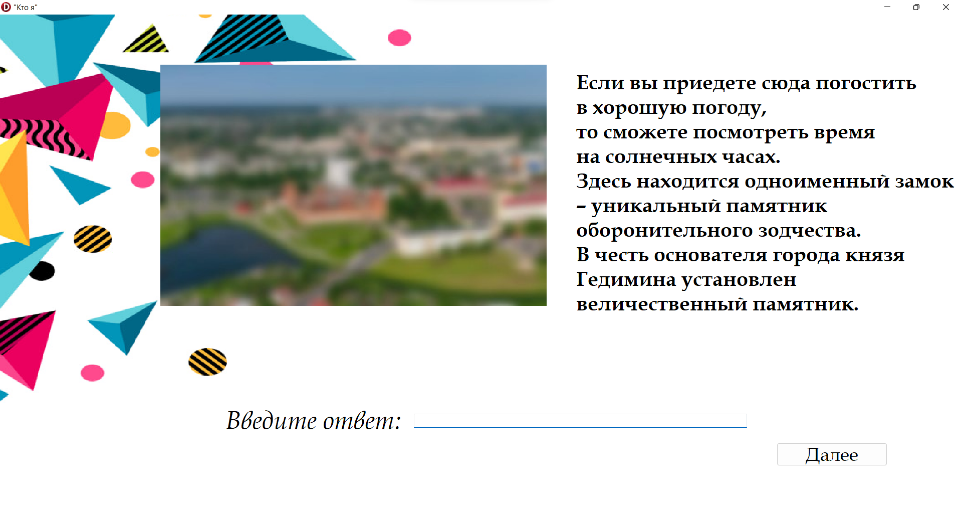
При выборе варианта ответа и нажатии кнопки «Далее» будет осуществлять переход на следующее задание в этой форме и переход на форму с выбором заданий.

Рисунок 10-Форма с заданием «Кто я?»

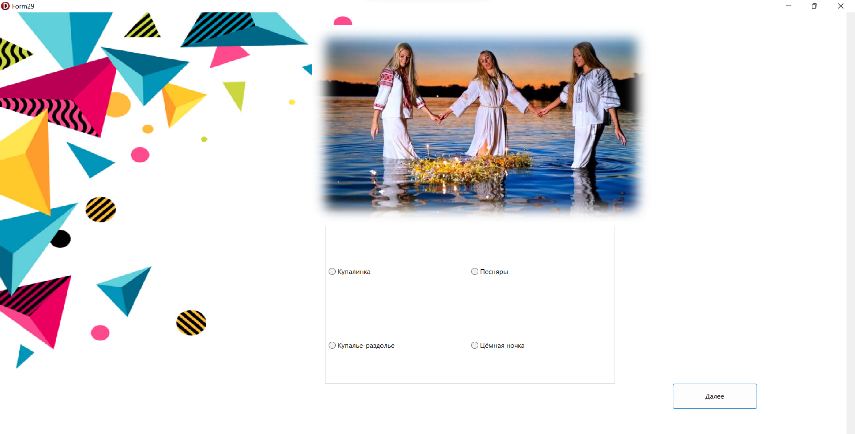
При вводе ответа и нажатии кнопки «Далее» будет осуществляться переход на следующее задание в этой форме и переход на форму с выбором заданий.

Рисунок 11-Форма с заданием «Угадай мелодию»

При выборе варианта ответа и нажатии кнопки «Далее» будет осуществлять переход на следующее задание в этой форме и переход на форму с результатами.

Рисунок 12-Форма с результатами

**Заключение**

Целью данного проекта заключалось обобщение знаний о родном крае.

В поставленной задаче был реализован простой и понятный пользовательский интерфейс.

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

В процессе разработки программного продукта я научился создавать динамические компоненты на форме, проработал их взаимодействие, закрепил умение создания собственных процедур, научился обрабатывать все исключительные ситуации.

Исходя из этого, можно сделать вывод, что программа реализована успешно.

**7. Список использованных источников**

1. Симонович С.В., Евсеев Г.А., Алексеев А.Г. Специальная информатика: Учебное пособие. - М.: АСТ-ПРЕСС: Инфорком-Пресс, 2001. - 480 с.
2. Архангельский, А. Я. Delphi 7. Справочное пособие. - Москва: Бином-Пресс, 2014. - 1024 с.
3. Вощинская, Г.Э. Разработка компонентов в DELPHI. - Воронеж: ИПЦ ВГУ, 2007. - 57 с.
4. Культин, Н. Основы программирования в Delphi 7. - Санкт-Петербург: Питер, 2009. - 640 с.
5. Архангельский, А. Я. Delphi 7. Справочное пособие. - Москва: Бином-Пресс, 2014. - 1024 с.
6. Культин, Н. Delphi 6. Программирование на Object Pascal / Н. Культин. - М.: БХВ-Петербург, 2012. - 528 c.

**Приложение А**

Листинг программы

unit Unit3;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Imaging.jpeg, Vcl.ExtCtrls,

Vcl.StdCtrls, Vcl.Imaging.pngimage, Vcl.Menus, ShellAPI;

type

TForm3 = class(TForm)

Image2: TImage;

Button1: TButton;

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure N1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

uses

unit1;

{$R \*.dfm}

procedure TForm3.Button1Click(Sender: TObject);

begin

Form3.Hide;

Form1.Show;

end;

procedure TForm3.FormCreate(Sender: TObject);

begin

WindowState:=wsMaximized;

end;

procedure TForm3.N1Click(Sender: TObject);

begin

ShellExecute(0,Pchar('Open'), Pchar('C:\2 курс\ОАИП\Курсовой проект\Курсовой проект\NewProject.chm'), nil,nil, SW\_SHOW);

end;

procedure TForm3.N2Click(Sender: TObject);

begin

if MessageDlg('Вы уверены, что хотите закрыть приложение?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes

then close;

end;

procedure TForm3.N3Click(Sender: TObject);

begin

Showmessage('Приложение разработала учащаяся ГУО "Гродненский государственный политехнический колледж" второго курса, группы ПЗТ-38 Леута Ксения Петровна')

end;

unit Unit1;

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;

type

TForm1 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

Label2: TLabel;

Button1: TButton;

Image1: TImage;

Label3: TLabel;

Edit2: TEdit;

Image2: TImage;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses

unit2;

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

Form1.Hide;

Form2.Show;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

WindowState:=wsMaximized;

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.ExtCtrls, Vcl.Imaging.jpeg,

Vcl.StdCtrls, Vcl.Imaging.pngimage;

type

TForm2 = class(TForm)

Image1: TImage;

Image3: TImage;

Image2: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

procedure Button1Click(Sender: TObject);

procedure Image3Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Image4Click(Sender: TObject);

procedure Image5Click(Sender: TObject);

procedure Image6Click(Sender: TObject);

procedure Image2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses

unit6, Unit7, unit16, unit18,unit29;

{$R \*.dfm}

procedure TForm2.Button1Click(Sender: TObject);

begin

Form2.Hide;

Form6.Show;

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

WindowState:=wsMaximized;

end;

procedure TForm2.Image2Click(Sender: TObject);

begin

form2.Hide;

form7.show

end;

procedure TForm2.Image3Click(Sender: TObject);

begin

Form2.Hide;

Form6.Show;

image3.Enabled := false;

end;

procedure TForm2.Image4Click(Sender: TObject);

begin

Form2.Hide;

Form16.Show;

image4.Enabled := false;

form16.Button2.Click;

end;

procedure TForm2.Image5Click(Sender: TObject);

begin

Form2.Hide;

Form18.Show;

image5.Enabled := false;

end;

procedure TForm2.Image6Click(Sender: TObject);

begin

Form2.Hide;

Form29.Show;

image6.Enabled := false;

end;

end.

unit Unit6;

interface

uses

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

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

Vcl.ExtCtrls;

type

TForm6 = class(TForm)

Image1: TImage;

Label1: TLabel;

Button1: TButton;

Button2: TButton;

Label2: TLabel;

procedure FormCreate(Sender: TObject);

procedure next\_question;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure end\_test;

procedure proverka;

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form6: TForm6;

cur\_quest\_num: integer = 1;

labels: array[1..10] of string;

answers: array[1..10] of integer;

t,f, i, q:integer;

a:array [1..15] of integer;

implementation

uses

unit2;

{$R \*.dfm}

procedure TForm6.FormCreate(Sender: TObject);

begin

WindowState:=wsMaximized;

labels[1] := 'В состав Гродненской области входит 15 районов';

labels[2] := 'В Гродненской области более 120 достопримечательностей';

labels[3] := 'В Гродненской области расположены '+#10+'около 1000 промышленных предприятий.';

labels[4] := 'Островец является самым молодым городом страны.';

labels[5] := 'Самым древним городом Гродненской области является Волковыск.';

labels[6] := 'По территории Гродненской области протекает 85 рек.';

labels[7] := 'На панно, которое украшает пожарную башню в Гродно,'+#10+'изображена девушка-спасатель, лицо которой'+#10+'очень напоминает всем известную героиню картины да Винчи';

labels[8] := 'Самая старая действующая аптека Беларуси находится в Гродно.';

labels[9] := 'В Щучинском районе растет пробковый лес.';

labels[10] :='Неподалёку от Слонима находится «Бездонное озеро»,';

a[1]:=0;

a[2]:=1;

a[3]:=0;

a[4]:=1;

a[5]:=1;

a[6]:=0;

a[7]:=1;

a[8]:=1;

a[9]:=1;

a[10]:=1;

end;

Procedure TForm6.proverka;

begin

for i:=1 to 10 do

if answers[i]=a[i] then begin q:=q+1;Label2.Caption:=inttostr(q); end;

end;

procedure TForm6.next\_question;

begin

cur\_quest\_num := cur\_quest\_num + 1;

if cur\_quest\_num=11 then begin

end\_test; proverka; end

else Label1.Caption:=labels[cur\_quest\_num]

end;

procedure TForm6.Button1Click(Sender: TObject);

begin

answers[cur\_quest\_num] :=1;

next\_question;

end;

procedure TForm6.Button2Click(Sender: TObject);

begin

answers[cur\_quest\_num] :=0;

next\_question;

end;

procedure TForm6.end\_test;

begin

Form6.Hide;

Form2.Show;

end;

end.

unit Unit7;

interface

uses

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

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

Vcl.ExtCtrls, Vcl.Imaging.pngimage;

type

TForm7 = class(TForm)

Label1: TLabel;

Image2: TImage;

Label2: TLabel;

Edit1: TEdit;

Button1: TButton;

Image3: TImage;

Image1: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

Image7: TImage;

Image8: TImage;

Image9: TImage;

Image10: TImage;

Label3: TLabel;

procedure FormCreate(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form7: TForm7;

images: array[1..10] of TImage;

cur\_quest\_num,b: integer;

q1,q2,q3,q4,q5,q6,q7,q8,q9:string;

implementation

uses

unit8, Unit2;

{$R \*.dfm}

procedure TForm7.Button1Click(Sender: TObject);

begin

case cur\_quest\_num of

1: begin q1:=edit1.Text; if q1='Шила в мешке не утаишь' then begin edit1.Text:='';inc(b);end; end;

2: begin q2:=edit1.Text; if q2='Два сапога пара' then begin edit1.Text:='';inc(b);end; end;

3: begin q3:=Edit1.Text; if q3='В тихом омуте черти водятся' then begin edit1.Text:='';inc(b);end; end;

4: begin q4:=Edit1.Text; if q4='С волками жить – по-волчьи выть' then begin edit1.Text:='';inc(b);end; end;

5: begin q5:=Edit1.Text; if q5='Снявши голову, по волосам не плачут' then begin edit1.Text:='';inc(b);end; end;

6: begin q6:=Edit1.Text; if q6='Мягко стелет, да жёстко спать.' then begin edit1.Text:='';inc(b);end; end;

7: begin q7:=Edit1.Text; if q7='Яблоко от яблони недалеко падает' then begin edit1.Text:='';inc(b);end; end;

8: begin q8:=Edit1.Text; if q8='Взялся за гуж – не говори, что не дюж.' then begin edit1.Text:='';inc(b); end; end;

end;

//Label3.Caption:=inttostr(b);

images[cur\_quest\_num].Visible := False;

cur\_quest\_num := cur\_quest\_num + 1;

images[cur\_quest\_num].Visible := True;

if cur\_quest\_num=9 then begin

form7.Hide;

form2.show;

end;

end;

procedure TForm7.FormCreate(Sender: TObject);

begin

cur\_quest\_num := 1;

WindowState:=wsMaximized;

images[1]:=Image2;

images[2]:=Image3;

images[3]:=Image4;

images[4]:=Image5;

images[5]:=Image6;

images[6]:=Image7;

images[7]:=Image8;

images[8]:=Image9;

images[9]:=Image10;

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.Imaging.jpeg,

Vcl.ExtCtrls, unit2;

type

TForm16 = class(TForm)

Image1: TImage;

Button1: TButton;

RadioGroup1: TRadioGroup;

Button2: TButton;

Label1: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form16: TForm16;

f: text;

s: string;

Nvern, ball, i, x: integer;

implementation

uses

unit3;

{$R \*.dfm}

procedure TForm16.Button1Click(Sender: TObject);

begin

//Если выбран вариант ответа и не достигнут конец файла

if (RadioGroup1.ItemIndex>-1) and (not Eof(f)) then begin

if RadioGroup1.ItemIndex = Nvern-1 then ball:=ball+1; //Если выбранный вариант соответствует

RadioGroup1.Items.Clear; //номеру верного ответа то балл прибавляется

Repeat //и очищается поле для следующего вопроса

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

delete(s,1,1);

RadioGroup1.Caption:=s;

end

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

delete(s,1,1);

Nvern:=StrToInt(s);

end

else RadioGroup1.Items.Add(s);

readln(f,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 RadioGroup1.ItemIndex = Nvern-1 then ball:=ball+1;

Label1.Caption:=inttostr(ball);

CloseFile(f);

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

Form16.Hide;

form2.Show;

end;

end;

procedure TForm16.Button2Click(Sender: TObject);

begin

Button2.Visible:=False;

Button1.Visible:=true;

RadioGroup1.Visible:=True;

AssignFile( f ,'text.txt');//связывается с разными файлами

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);

RadioGroup1.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 RadioGroup1.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.

unit Unit18;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Imaging.jpeg, Vcl.ExtCtrls,

Vcl.StdCtrls;

type

TForm18 = class(TForm)

Image1: TImage;

Image2: TImage;

Label1: TLabel;

Edit2: TEdit;

Label2: TLabel;

Button1: TButton;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

Image7: TImage;

Label3: TLabel;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form18: TForm18;

images: array[1..10] of TImage;

cur\_quest\_num,b: integer;

labels: array[1..10] of string;

answers: array[1..10] of boolean;

q1,q2,q3,q4,q5,q6,q7:string;

implementation

uses

unit2;

{$R \*.dfm}

procedure TForm18.Button1Click(Sender: TObject);

begin

case cur\_quest\_num of

1: begin q1:=edit2.Text; if q1='Лида' then begin edit2.Text:='';inc(b);end; end;

2: begin q2:=edit2.Text; if q2='Слоним' then begin edit2.Text:='';inc(b);end; end;

3: begin q3:=Edit2.Text; if q3='Волковыск' then begin edit2.Text:='';inc(b);end; end;

4: begin q4:=Edit2.Text; if q4='Мосты' then begin edit2.Text:='';inc(b);end; end;

5: begin q5:=Edit2.Text; if q5='Щучин' then begin edit2.Text:='';inc(b);end; end;

end;

//Label3.Caption:=inttostr(b);

label2.Caption:=labels[cur\_quest\_num];

images[cur\_quest\_num].Visible := False;

cur\_quest\_num := cur\_quest\_num + 1;

images[cur\_quest\_num].Visible := True;

if cur\_quest\_num=7 then begin

Form18.Hide;

Form2.Show;

end;

end;

procedure TForm18.FormCreate(Sender: TObject);

begin

label2.Caption:='Если вы приедете сюда погостить'+#10+'в хорошую погоду,'+#10+'то сможете посмотреть время'+#10+'на солнечных часах. '+#10+'Здесь находится одноименный замок '+#10+'– уникальный памятник '+#10+'оборонительного зодчества. '+#10+'В честь основателя города князя'+#10+'Гедимина установлен'+#10+'величественный памятник.';

cur\_quest\_num := 2;

images[1]:=Image1;

images[2]:=Image2;

images[3]:=Image3;

images[4]:=Image4;

images[5]:=Image5;

images[6]:=Image6;

images[7]:=Image7;

labels[2] := 'Город возник как пограничное'+#10+'укрепление,'+#10+'даже название происходит от'+#10+'древнеславянского слова,'+#10+'означающего «преграда».'+#10+'Особое место – это пригород,'+#10+'усадьба рода Пусловских.'+#10+'Здесь крутейшая атмосфера'+#10+'в парковом массиве с прудами.'+#10+'Рядом в 10 километрах находится'+#10+'известнейший Жировичский'+#10+'православный монастырь.';

labels[3] := 'Туристам можно порекомендовать'+#10+' посетить уникальный военно'+#10+'-исторический музей'+#10+'имени П.И.Багратиона '+#10+'с богатейшей экспозицией '+#10+'по истории войны 1812 года.'+#10+'На гербе города изображён волк,'+#10+'смотрящий назад.';

labels[4] := 'Одно время город назывался'+#10+'Фабричный – благодаря'+#10+'фанерной фабрике братьев'+#10+'Конопацких.'+#10+'В городе находится'+#10+'самый короткий проспект в стране'+#10+'— проспект Мира, длиной'+#10+'немногим более километра, и'+#10+'железнодорожный вокзал — одно'+#10+'из старейших зданий в городе.';

labels[5] := 'Жители города гордятся башней'+#10+'с часами,украшающей'+#10+'центральную площадь, которую'+#10+'они прозвали «Биг-Беном».'+#10'Здесь находится памятник'+#10+'Элоизе Пашкевич (Тётке),'+#10+'знаменитой писательнице,'+#10+' поэтессе, общественной'+#10+'деятельнице и журналистке,'+#10+'уроженке этих мест.';

end;

end.

unit Unit29;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Imaging.jpeg, Vcl.ExtCtrls,

Vcl.Imaging.pngimage, Vcl.StdCtrls, Vcl.MPlayer;

type

TForm29 = class(TForm)

Image1: TImage;

Button1: TButton;

MediaPlayer1: TMediaPlayer;

Image2: TImage;

RadioGroup1: TRadioGroup;

Label1: TLabel;

procedure Button1Click(Sender: TObject);

procedure FormActivate(Sender: TObject);

var

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form29: TForm29;

f: text;

s: string;

Nvern, ball, i, x: integer;

// images: array[1..5] of TImage;

cur\_quest\_num: integer;

// answers: array[1..5] of byte;

implementation

uses

unit30;

{$R \*.dfm}

procedure TForm29.Button1Click(Sender: TObject);

begin

inc(cur\_quest\_num);

if cur\_quest\_num=5 then begin

MediaPlayer1.Stop;

Form29.Hide;

Form30.Show;

end

else begin

Image2.Picture.LoadFromFile(IntToStr(cur\_quest\_num)+'.jpg');

MediaPlayer1.Filename:='audio\'+IntToStr(cur\_quest\_num)+'.mp3';

MediaPlayer1.Open;

MediaPlayer1.Play;

end;

//Если выбран вариант ответа и не достигнут конец файла

if (RadioGroup1.ItemIndex>-1) and (not Eof(f)) then begin

if RadioGroup1.ItemIndex = Nvern-1 then ball:=ball+1; //Если выбранный вариант соответствует

RadioGroup1.Items.Clear; //номеру верного ответа то балл прибавляется

Repeat //и очищается поле для следующего вопроса

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

delete(s,1,1);

RadioGroup1.Caption:=s;

end

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

delete(s,1,1);

Nvern:=StrToInt(s);

end

else RadioGroup1.Items.Add(s);

readln(f,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 RadioGroup1.ItemIndex = Nvern-1 then ball:=ball+1;

Label1.Caption:=inttostr(ball);

CloseFile(f);

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

end;

// Form29.Caption:=IntToStr(cur\_quest\_num);

// images[cur\_quest\_num].Visible := False;

// inc(cur\_quest\_num);

// if cur\_quest\_num=6 then begin

// Form29.Hide;

// Form30.Show;

// MediaPlayer1.stop;

// end

// else begin

// images[cur\_quest\_num].Visible := True;

// MediaPlayer1.Filename:='audio\'+IntToStr(cur\_quest\_num)+'.mp3';

// MediaPlayer1.Open;

// MediaPlayer1.Play;

// end;

end;

procedure TForm29.FormActivate(Sender: TObject);

//begin

//cur\_quest\_num := 1;

//images[1]:=Image2;

//images[2]:=Image3;

//images[3]:=Image4;

//images[4]:=Image5;

//images[5]:=Image6;

////form29.Close();

//end;

begin

Image2.Picture.LoadFromFile('1(1).jpg');

MediaPlayer1.FileName:='audio/1(1).mp3';

MediaPlayer1.Open;

MediaPlayer1.Play;

Button1.Visible:=true;

RadioGroup1.Visible:=True;

AssignFile( f ,'muzz.txt');//связывается с разными файлами

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);

RadioGroup1.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 RadioGroup1.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.

unit Unit30;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Imaging.jpeg, Vcl.ExtCtrls,

Vcl.StdCtrls, Vcl.Imaging.pngimage;

type

TForm30 = class(TForm)

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Image2: TImage;

procedure FormActivate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form30: TForm30;

implementation

{$R \*.dfm}

uses Unit6, Unit7, Unit16, Unit18, Unit29;

procedure TForm30.FormActivate(Sender: TObject);

var

otveti:integer;

begin

otveti:=q+ball+ball1+b+c;

Label1.Caption:='Правильных ответов: '+inttostr(otveti);

labels[4] := 'Если вы набрали правильных ответов:'+#10+' более 35 баллов - вы прекрасно знаете нашу страну';

labels[5] := '25-35 – хорошо,но есть ещё области знания, готовые вам покориться.'

end;

end.