**Введение**

На курсовое проектирование была поставлена задача – разработать приложение на тему «Разработка программного продукта «Электронная энциклопедия моды»».

Цель проекта заключается в разработке приложения, при помощи которого пользователь может изучить разные стили в одежде, проверить свои знания тестом, а также в свободном режиме сыграть в игру «одевалка». Данная программа подходит для людей младших-средних классов, так как заинтересует их своей информативностью и интерактивностью.

Также, для работы с программой выход в интернет не нужен: она работает с файлами, содержащими все необходимое для пользования.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нем вы можете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической сущности. Так же в данном 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 в 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. Здесть 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 прогаммой, начиная от запуска до выходы.

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

В «Списке использованных источников» будет приведен список используемых при разработке источников.

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

Схема работы системы будет представлена в графической части.

1. **Анализ задачи**
   1. **Постановка задачи**

Темой данного курсового проекта является «Электронная энциклопедия моды».

Трудно представить, как люди могут жить без одежды. Ещё с древних времен статус человека могли определить только взглянув на его обличие. Цвет, длина, покрой – всё это говорило о стоящей перед тобой персоне. Кто-то побогаче мог позволить себе одежду подлиннее, более ярких, «богатых» цветов, когда бедняки и нищие ходили в лахмотьях и коротких убраниях. У всевозможных культур и народов было различное представление об одежде, но едино одно: она неотделима от повседневности человека.

Со временем все преобразовывалось: из шкур, обычных туник и повязок на поясницах люди переходили к платьям, вертюгалям и доспехам, а после одежда и вовсе начинает потихоньку преобретать современный вид.

В период времени, начиная с 50-ых годов прошлого столетия и до наших дней, в мире появилось множество различных группировок, субкультур, а также кругов лиц по интересам. И конечно же, в первую очередь, чтобы отличать «своего» от «чужого», люди помечали себя определенной стилистикой в одежде.

Многие ребята в современном поколении достаточно большое внимание уделяют своему имиджу, «прикиду» и так далее. Даже бессознательно: все мы стараемся подобрать верхнюю одежду под нижнюю, обувь под сумку. Выглядеть стильно и уникально в наше время – значит показать своё внутреннее «я». Толи это будет удобная, комфортная футболка и штаны с кучей карманов, толи большое количество вещей с множеством цепей и аксекссуаров на ней же. Возможно для кого-то носить открытую одежду – настоящее счастье. А кто-то предпочитает укутаться от лишних глаз в слоях непроглядной ткани.

В наше время люди свободны и имеют возможность выбирать то, как они могут одеваться. Нет фактически никаких запретов и ограничений на законодательном уровне в большинстве странах. Можно смело эксперементировать с образами и не бояться осуждения окружающих: общество максимально лояльно, толерантно, а в отдельных случаях может и похвалить за смелый образ на улице. Даже на самых популярных показах выставляются шокирующие образы, которые вызывают и радость и ужас – но всегда это яркая эмоция, которая оставляет отпечаток в душе человека.

Однако как же выбрать то, что тебе действительно будет по душе?

Изучение энциклопедии моды по разным стилям может понадобиться многим обычным пользователям. В частности, с изучением и определением собственного стиля в одежде, а также наслаждением в нахождении режиме «свободный режим».

Аналогов для такого приложения можно привести не так уж и много. В основном в сети интернет находятся различного рода статьи по даннной тематике, сочинения других людей. В более редких случаях – ознакомительный материал с примерами и проверками на внимательность читателя ввиде теста и вывода результата прохождения на экран. Собранный же воедино материал с интерактивом и обучающим элементом встречается единично на просторах интернет-магазинах для различного рода приложений.

Периодичность использования данного программного продукта неограничена. Пользователь может в любое время установить, а также удалить приложение после полного изучения материала по данной теме и прохождения тестов.

Цель данного курсового проекта – разработать программный продукт, который будет сочетать в себе элементы обучения, проверки усвоенного материала и развлечения для людей, интересующихся модой, стилем ушедших эпох, а также разработкой игры «одевалка».

За основу программы следует взять из соответствующих источников (сайтов, статей, энциклопедий) наиболее полную, достоверную и точную информацию. При реализации проекта разработчик должен стараться сделать его простым и максимально доступным для пользователей. Приложение должно располагать своим интерфейсом и функционалом к пользователю, чтобы ему было комфортно

Данный проект должен стать общедоступным для всех пользователей. В поставленной задаче наобходимо 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;

- просмотр статей;

- прохождение тестов для закрепления знаний;

- просмотр справки;

- просмотр пункта «о разработчике»;

- возможность сыграть в игру «одевалка».

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, статьи на тему стилей одежды в различные годы и\или эпохи, свободный режим игры, где можно одеть персонажа на свой вкус и расслабиться.

Постоянной информацией в проекте будут являться текстовые файлы, картинки, медиафайлы и др.

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

* 1. **Инструменты разработки**

Для разработки данного проекта будет выбрана среда Delphi 7, так как это самое удобная и доступная среда разработки на данный момент. Delphi 7 - язык –рограммирования, относящийся к классу RAD- (Rapid Application Development – «Средство быстрой разработки приложений») средств CASE – технологии. Delphi 7 сделал разработку приложений для windows быстрым и приятным процессом. Теперь разрабатывать сложные и интересные проекты можно только одним человеком, использующим Delphi 7.

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 7 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 7, который позволяет приспособить эти компоненты к имеющийся задаче, чтобы они работали именно так, как требуется создаваемому приложению. Также здесь имеются предварительно-определенные визуальные и невизуальные 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 7 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 7 позволяет создать различные виды программ: консольные

приложения, оконные приложения, приложения для работы с Интернетом и базами данных. То есть, Delphi 7 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 7:

- 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).

Также есть возможность сохранять изображение как файл фотошопа, заносить все в псоелднюю указанную программу и работать со слоями проекта, ничего не потеряв.

**Adobe Photoshop CS6 x64** – редактор для редактирования изображений.

Необходим для оформления некоторых элементов интерфейса. Также использовался и для создания элементов игры «одевалки». Сложен, но удобен своей многофункциольнальностью. От простого поворота фотографии и добавления текстов до работы с векторами и 3д объектами.

**Microsoft Word 2016** – редактор текста для написания документации.

**Microsoft Power Point 2016** – программа для создания презентации.

**Блокнот** – для создавания html страниц.

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

- процессор Intel Core i5;

- объем ОЗУ 4 гб;

- объем места на HDD – 500 гб ;

- видеоподсистема 1024х768 точек с глубиной цвета 16 Bit;

- ОС – Windows 10.

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

* 1. **Требования к приложению**

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

При моделировании форм следует учесть такой момент: основная проблема состоит в том, что нынешняя молодежь достаточно поверхностно ознакомлена с существующими стилями и образами в моде. По этой причине особенно актуальной становится разработка программного электронного продукта, который способствовал бы и давал возможность вспомнить, изучить историю моды. 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 (не крахом) 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 (крахом) 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. ПП должен иметь средства фиксации всех действий в процессе изготовления программного продукта. Это необходимо для восстановления состояния процесса разработки на любом его этапе (при комплексной отладке).

**2 Проектирование задачи**

**2.1 Организация данных**

Проектирование задачи – это 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 данных, используемых в программе, является вводимая пользователем информация (входные данные). Входной информацией в разрабатываемой программе будут являться данные о прохождении теста пользователем, комбинации картинок одежды в «свободном режиме». При прохождении теста пользователь выбирает определенный ответ из заданного количества вариантов. Сами варианты ответов хранятся в текстовых файлах и достаются и читаются оттуда через функцию RadioGroup. При захождении в «свободный режим» эксплуататор имеет возможность комбинировать предложенное количество вариантов одежды и одеть стоящую на холсте куклу. Данные об одежде хранятся и показываются через одномерны массив.

Вторым видом являются данные, которые будут введены разработчиком на этапе реализации задачи (выходная информация). Сюда можно отнести изображения (иконки, изображения в статье, изображения в «свободном режиме»), хранящиеся в html страницах, а также в файлах самого приложения. В программе также хранятся картинки для игры «одевалка» в файле с кодом. Выходной информацией также является выдаваемый результат за прохождение теста – то, насколько пользователь усвоил материал из статей. Результат подсчитывается в циклах проверки ответов и выводится на экран.

Условно-постоянной информацией является текст в html страницах, который храниться и открывается непосредственно из них. Также ею является музыка, играющая на фоне, пока пользователь использует приложение.

**2.2 Процессы**

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно должна заниматься разрабатываемая курсовая программа.

Главные ее задачи:

- демонстрация статей на тему «стили одежды»;

- прохождение тестов на проверку знаний;

- возможность пользоваться «Свободным режимом».

Ниже приведены функциональные модели разработки главных задач. Первая – прохождение тестов на проверку знаний (Рисунок 1):

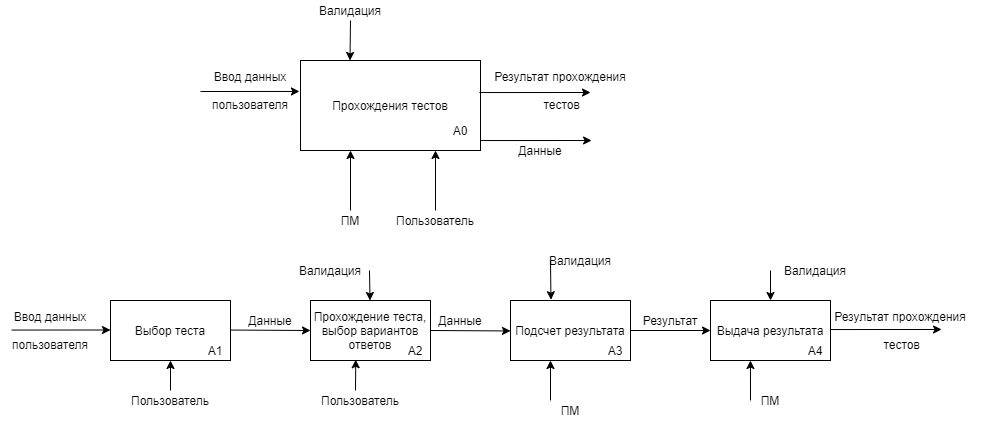


Рисунок 1 - Функциональная модель процесса прохождения тестов

Вторая – демонстрация статей на тему «стили одежды» (Рисунок 2):



Рисунок 2 - Функциональная модель процесса скроллинга статей

Третья – использование «свободного режима» (Рисунок 3):

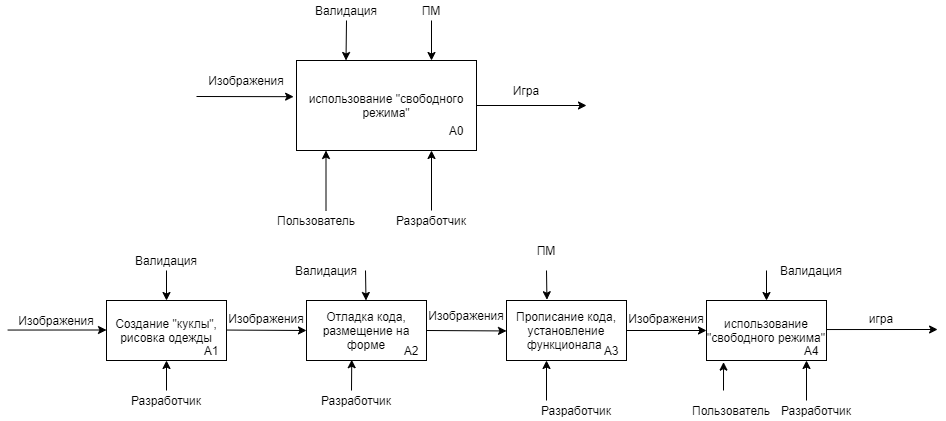


Рисунок 3 - Функциональная модель процесса «свободного режима»

**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 представлена система меню и организация навигации между окнами программы (Рисунок 4):

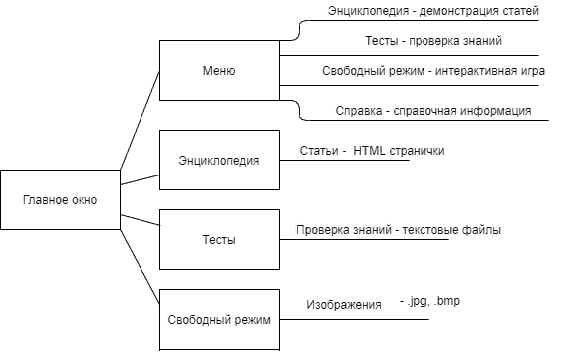


Рисунок 4 - Навигация между окнами программы

Прототип – это наглядная модель пользовательского интерфейса. В сущности это «черновик», созданный на основе представления разработчика о потребностях пользователя. Итоговое отображение программы может отличаться от прототипа. Ниже можно наблюдать прототипы:

- Логотипа (Рисунок 5);

- Заставки (Рисунок 6);

- Главного окна (Рисунок 7);

- Окно, содержащее статьи об стилях одежды (Рисунок 8);

- Окно, содержащее тест и выводящее результаты (Рисунок 9);

- Окно, содержащее себе интерактивную игру «одевалка» (Рисунок 10).



Рисунок 5 – логотип

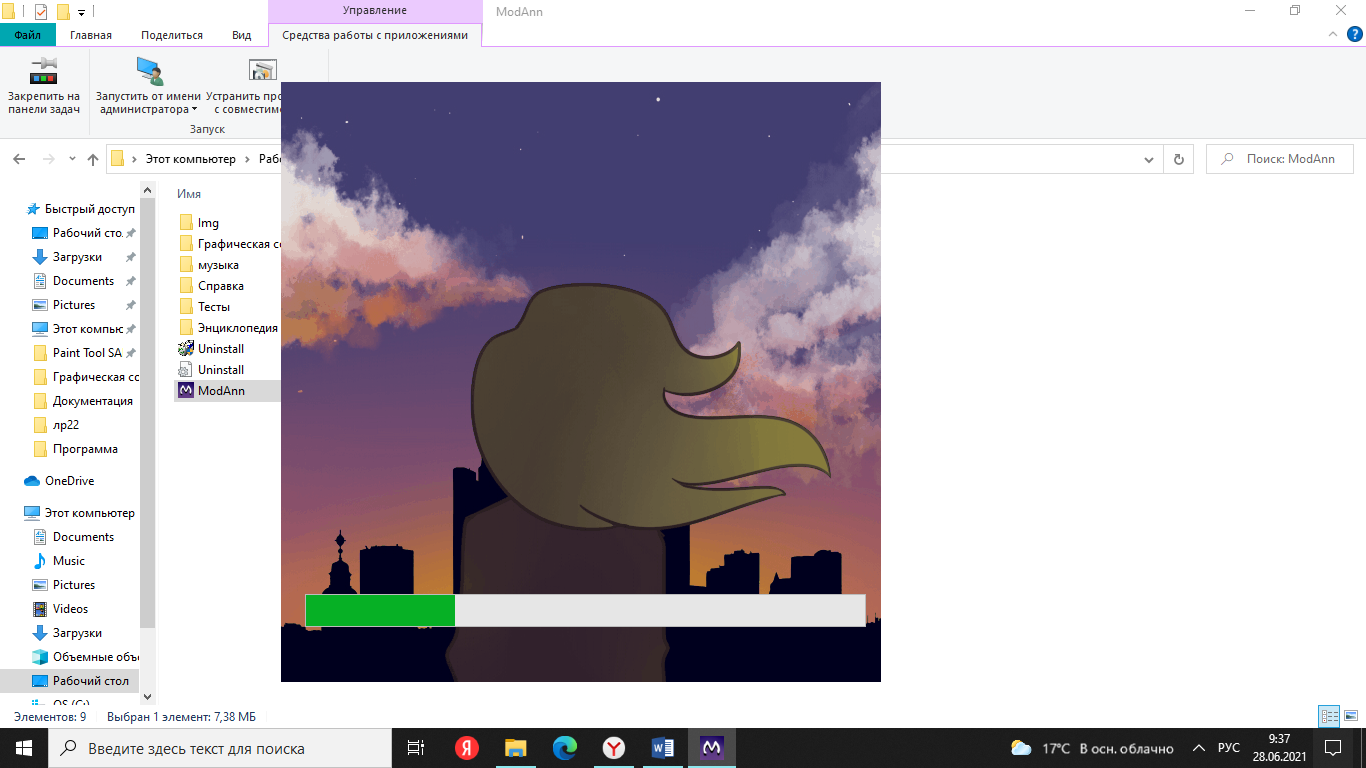


Рисунок 6 - заставка

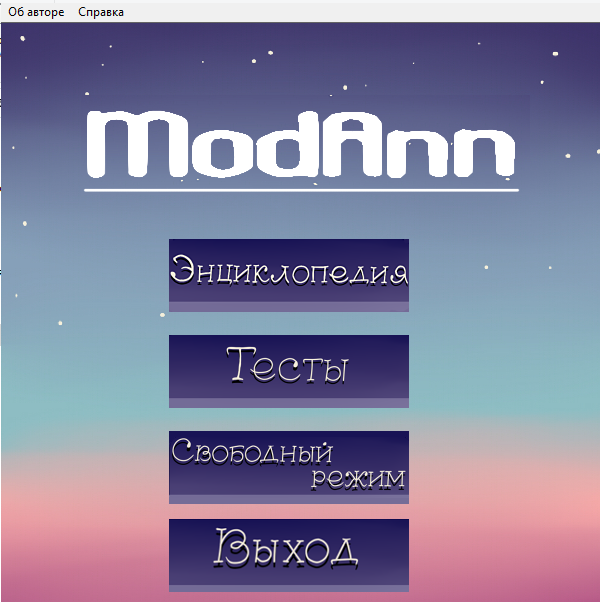


Рисунок 7 - Главное окно

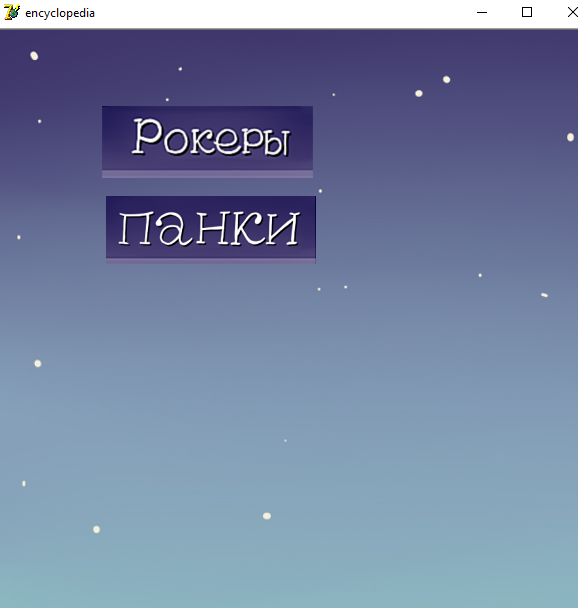


Рисунок 8 - Окно, содержащее статьи об стилях одежды

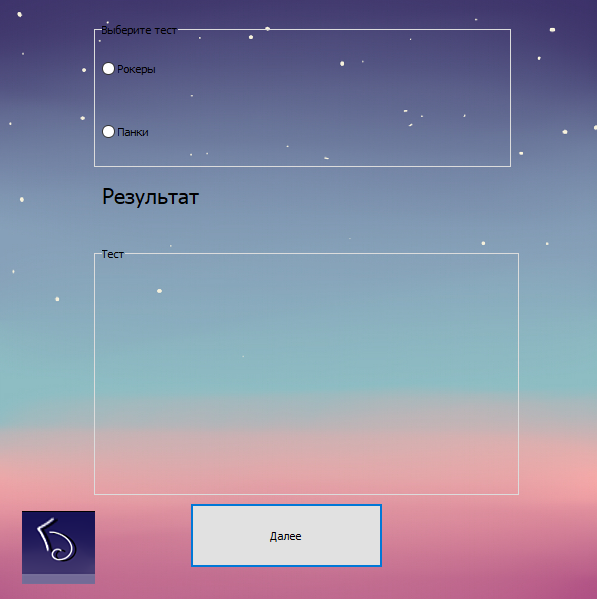


Рисунок 9 - Окно, содержащее тест и выводящее результаты

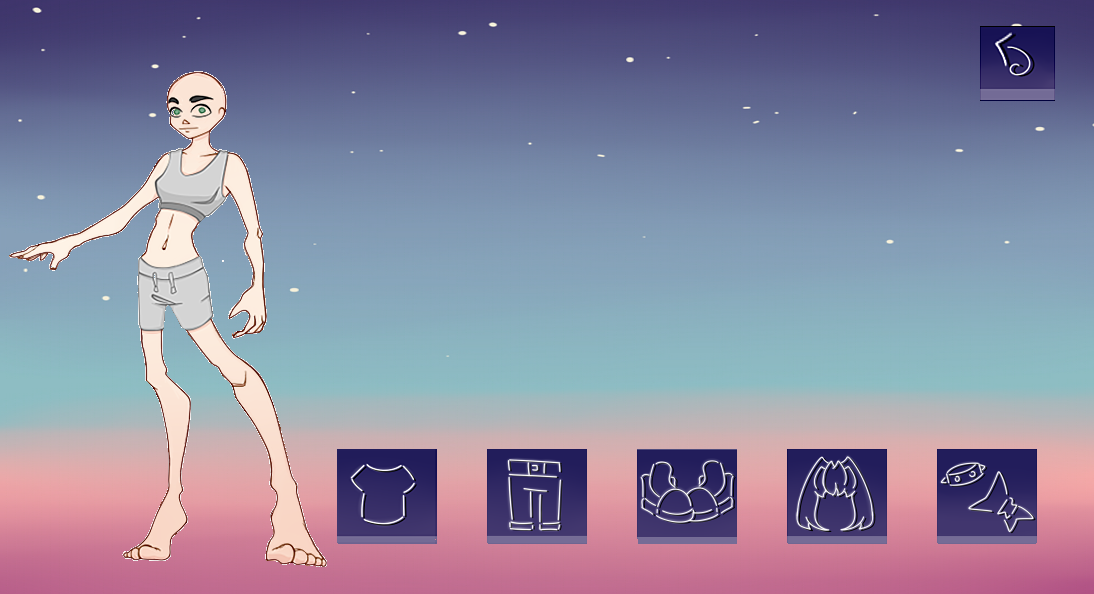


Рисунок 10 – Свободный режим

**3 Реализация**

**3.1 Реализация проекта**

**3.1.1 Структура программы**

Для реализации данного курсового проекта понадобилось создать 6 модулей и справка, где описана информация о приложении. Далее будет описано назначение для каждого из них.

Модуль Unit 1 - является заставкой для приложения. Подгружает форму, где находится главное меню, а само закрывается.

Модуль Unit 2 - главное меню проекта. С его помощью можно перейти на остальные формы, а именно: энциклопедия, тесты, “свободный режим”, “об авторе”, справка.

Модуль Unit 3 - содержит в себе энциклопедию моды. В ней можно переходить от статьи к статье и вернуться на главную форму.

Модуль Unit 4 - в нем реализован одноразовый тест, на который можно перейти с главного меню. Нажав на кнопку «назад», можно вернуться в главное меню.

Модуль Unit 5 – на этой форме реализован “свободный режим”, в котором пользователь может одеть куклу в одежду из предложенных. На эту форму можно перейти через вторую, а также вернуться в главное меню обратно.

Модуль 6 - содержит в себе информацию “об авторе”, где разработчик оставил желаемую информацию о себе, а также о своем проекте. Открывается через MainMenu. Можно вернуться на главную форму, если нажать крестик на форме и заркыть модуль.

Справка находится в том же разделе, что и информация «об авторе». Служит для ознакомления пользователей с программный продуктом в случае возникновения сложностей с эксплуатацией.

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

Описание разработанных процедур и функций приводится в таблице 1.

Таблица 1 – Процедуры и функции

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Имя процедуры (функции) | В каком модуле находится | За каким управлением элемента закреплена | Назначение |
| 1 | 2 | 3 | 4 | 5 |
| 1 | procedure TForm1.Timer1Timer(Sender: TObject); | Unit1.pas | Timer1 | Делать имитацию загрузки приложения, а также открывать форму с главным меню. |
| 2 | procedure TForm1.FormCreate(Sender: TObject); | Unit1.pas | Image1 | Загружать гифку на фон |
| 3 | procedure TForm2.Image3Click(Sender: TObject); | Unit2.pas | Image3 | Выполнять функцию кнопки, служит для перехода на «энциклопедию» |
| 4 | procedure TForm2.Image6Click(Sender: TObject); | Unit2.pas | Image6 | Закрывать приложение |
| 5 | procedure TForm2.Image4Click(Sender: TObject); | Unit2.pas | Image4 | Открывать форму с тестами |
| 6 | procedure TForm2.Image5Click(Sender: TObject); | Unit2.pas | Image5 | Открывать форму со свободным режимом |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 7 | procedure TForm2.N1Click(Sender: TObject); | Unit2.pas | MainMenu1 | Открывать форму «об авторе» |
| 8 | procedure TForm2.N2Click(Sender: TObject); | Unit2.pas | MainMenu1 | Открыть форму «справка» |
| 9 | procedure TForm3.FormCreate(Sender: TObject); | Unit3.pas | WebBrowser1 | Открывать html страницы как «энциклопедию» |
| 10 | procedure TForm4.Button1Click(Sender: TObject); | Unit4.pas | Button1 | Начинать прохождение теста и получать результат |
| 11 | procedure TForm4.RadioGroup1Click(Sender: TObject); | Unit4.pas | RadioGroup1 | Выборка теста и подсчет результата |
| 12 | procedure TForm4.Image2Click(Sender: TObject); | Unit4.pas | Image1 | Возвращение в главное меню |
| 13 | procedure TForm5.FormCreate(Sender: TObject); | Unit5.pas | FormCreate | Загрузка в массивы картинки |
| 14 | procedure TForm5.Timer1Timer(Sender: TObject); | Unit5.pas | Timer1 | Загрузка фона, прорисовка куклы и кнопок |
| 15 | procedure TForm5.FormMouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | Unit5.pas | FormMouseDown | Разработана функция кнопок |
| 16 | Procedure TForm5.FormCanResize(Sender: TObject; var NewWidth, NewHeight: Integer; var Resize: Boolean); | Unit5.pas | FormCanResize | Контроль размера формы |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 17 | procedure TForm5.Image1Click(Sender: TObject); | Unit5.pas | Image1 | Возвращает в главное меню |

**3.1.3 Описание использованных компанентов**

Описание использованных при разработке приложения компонентов приводится к таблице 2.

Таблица 2 – Использованные компоненты

|  |  |  |  |
| --- | --- | --- | --- |
| № | Компонент | На какой форме расположен | Назначение |
| 1 | 2 | 3 | 4 |
| 1 | TMainMenu | Form2 | Используется для просмотра форм «об авторе» и «справка» |
| 2 | TButton | Form4 | Используется для выбора следующего вопроса в тесте |
| 3 | TImage | Form1, Form2, Form 4, Form 5, Form 6 | Служит как фон для приложения, заставка, а также как кнопки и реализация «свобожного режима» |
| 4 | TTimer | Form1, Form5 | Реализована заставка, а также в нем загружаются картинки на форму в «свободном режиме» |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | WebBrowser1 | Form3 | Открывает html-страницы, где содержится энциклопедия |
| 6 | TRadioGroup | Form5 | Хранится выбор прохождения теста, а также сами вопросы |
| 7 | TFormMouseDown | Form5 | Реализация имитации кнопок для выбора одежды |
| 8 | TFormCanResize | Form5 | Контролировать размер формы |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

|  |  |  |
| --- | --- | --- |
| № | Имя Файла | Назначение |
| 1 | 2 | 3 |
| 1 | ModAnn.exe | Исполняемый файл проекта, используется для запуска файла на выполнение. |
| 2 | ModAnn.dpr | Файл проекта, связывает все файлы из которых состоит приложение. |
| 3 | Unit1.pas | Файл программного модуля формы с заставкой. |
| 4 | Unit2.pas | Файл программного модуля формы главного меню. |
| 5 | Unit3.pas | Файл программного модуля формы с энциклопедией. |
| 6 | Unit4.pas | Файл программного модуля формы с тестами. |
| 7 | Unit5.pas | Файл программного модуля формы со «свободным режимом». |
| 8 | Unit6.pas | Файл программного модуля формы «об авторе». |
| 9 | Unit7.pas | Файл программного модуля формы «справка». |
| 10 | SetUp.exe | Установочный файл. |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 11 | t1.txt, t2.txt | Текстовые файлы с тестами |
| 12 | Папка «графическая составляющая» | В ней хранятся картинки для главного меню, а также других форм |
| 13 | Папка «энциклопедия» | Хранит html-файлы для открытия их в энциклопедии приложении |
| 14 | Папка «img» | Хранятся картинки для «свободного режима» |
| 15 | Папка «справка» | Хранится справка |
| 16 | Unit1.dfm | Код в форме 1 |
| 17 | Unit2.dfm | Код в форме 2 |
| 18 | Unit3.dfm | Код в форме 3 |
| 19 | Unit4.dfm | Код в форме 4 |
| 20 | Unit5.dfm | Код в форме 5 |
| 21 | Unit6.dfm | Код в форме 6 |
| 22 | Unit7.dfm | Код в форме 7 |
| 23 | Unit8.dfm | Код в форме 8 |
| 24 | Unit9.dfm | Код в форме 9 |
| 25 | Unit10.dfm | Код в форме 10 |

**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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Проверка загрузки гифки в заставке | Загрузится выбранная гифка | Загрузится выбранная гифка | Выполнено |
| 2 | Проверка работы полосы загрузки | Отображение загрузки | Отображение загрузки | Выполнено |
| 3 | Проверка главного меню | Отображение всех «кнопок» | Отображение всех «кнопок» | Выполнено |
| 4 | Переход по кнопкам в главном меню | Переход на другую форму соотвественно своей «кнопке» | Переход на другую форму соотвественно своей «кнопке» | Выполнено |
| 5 | Проверка пункта меню «энциклопедия» | Открытие списка статей | Открытие списка статей | Выполнено |
| 6 | Скроллинг статей | Исправное открытие html страниц | Исправное открытие html страниц | Выполнено |
| 7 | Работа кнопок в статьях | Переход между страницами | Переход между страницами | Выполнено |

Продолжение таблицы 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 8 | Работа кнопки «назад» | Переход назад на главное меню | Переход назад на главное меню | Выполнено |
| 9 | Кнопка «тесты» | Переход на форму с тестами | Переход на форму с тестами | Выполнено |
| 10 | Выбор в прохождении теста | Открытие нужного текстового фалйа | Открытие нужного текстового фалйа | Выполнено |
| 11 | Прохождение теста | Исправное выполнение выбора варианта ответа | Исправное выполнение выбора варианта ответа | Выполнено |
| 12 | Проверка вывода ответа | Вывод ответа | Вывод ответа | Выполнено |
| 13 | Кнопка «назад» | Выход в главное меню | Выход в главное меню | Выполнено |
| 14 | Кнопка «свободный режим» | Открытие формы «свободный режим» | Открытие формы «свободный режим» | Выполнено |
| 15 | Работа формы | Открытие всех изображений на форме | Открытие всех изображений на форме | Выполнено |
| 16 | Работа обаботки кнопок | Исправное показывание одежды на кукле | Исправное показывание одежды на кукле | Выполнено |
| 17 | Кнопка «назад» | Выход в главное меню | Выход в главное меню | Выполнено |

Продолжение таблицы 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 18 | Кнопка «об авторе» | Исправно открывает форму | Исправно открывает форму | Выполнено |
| 19 | Форма «об авторе» | Исправно показывает все изображения на форме | Исправно показывает все изображения на форме | Выполнено |
| 20 | Кнопка «справка» | Исправно открывает справку приложения | Исправно открывает справку приложения | Выполнено |
| 21 | Работа справки | Исправно функционирует | Исправно функционирует | Выполнено |
| 22 | Медиаплеер | Музыка исправно идет на фоне работающей программы | Музыка исправно идет на фоне работающей программы | Выполнено |
| 23 | Кнопка «выход» | Осуществляет выход из программного продукта | Осуществляет выход из программного продукта | Выполнено |
| 24 | Ошибка «сode 8: недостаточно средств памяти для выполнения операции» | Устранение ошибки | Устранение ошибки не осуществимо | Не выполнено |

В результате выполнения тестирования программного продукта «Энциклопедия моды» не все ошибки были исправлены, однако разработчик будет учитывать её в будущем.

**5 Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Разработанный программный продукт является учебно-развлекательным приложением «Электронная энциклопедия моды». Данный программный продукт предназначен для пользователей, интересующимися различными стилями в моде, а также любителям игр жанра «аркада».

Быстродействие любого программного приложения во многом зависит от характеристики выбранного персонального компьютера: рабочей частоты процессора, частоты обработки видеокарты, объема оперативной памяти, а также на жестком диске, и так далее. Несмотря на все реализованные задачи в в данном программном продукте, он легко запускается и функционирует на любых вычислительных машинах.

Тестирование проводилась на различных электроннно-вычислительных машинах, но даже на системе следующего типа работать было комфортно и приятно:

- Процессор Intel i3;

- Объем озу 2 ГБ;

- Видеосистема: 1024\*768 точек с глубиной цвета 16bit;

- Операционная система Windows XP.

**5.2 Инсталляция**

Для того, чтобы установить программу, необходимо запустить файл Setup.exe. Появится окно игрового-развлекательного приложения ModAnn (рисунок 11).

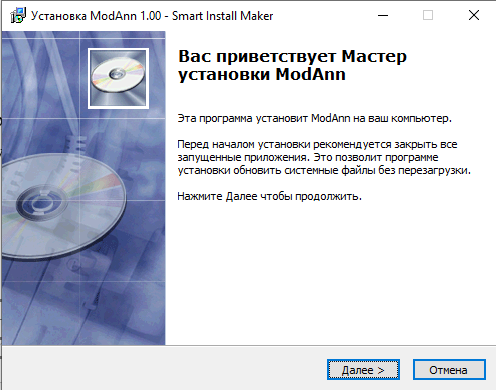


Рисунок 11 – Установка игрового приложения ModAnn

После нажатия на кнопку «Далее» появляется окно выбора каталога установили (рисунок 12). Нажав на кнопку «обзор», вы можете поменять место для сохранения файла.

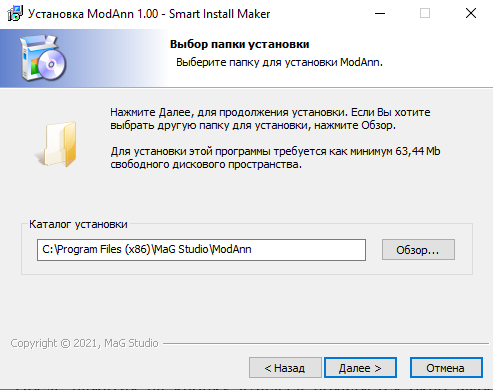


Рисунок 12 – выбор каталога установки

После нажатия на кнопку «Далее» появится скриншот «Все готово» (рисунок 13). Выбрав все необходимое для установки программного продукта, вам будет предложено нажать на кнопку «установить». После её нажатия пойдет загрузка файлов (рисунок 14). Итоговым станет окно «Ваше приложение установлено и готово к работе» (рисунок 15).

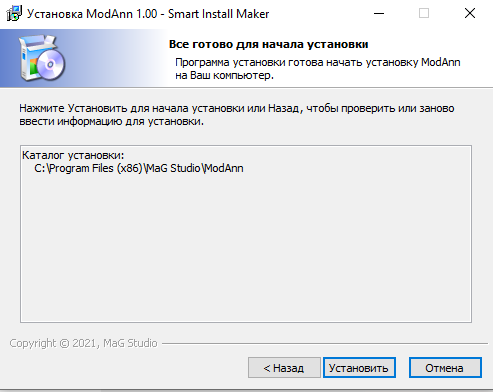


Рисунок 13 – Финальная установка

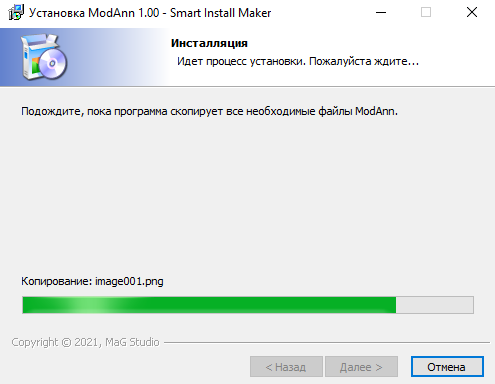


Рисунок 14 – загрузка приложения

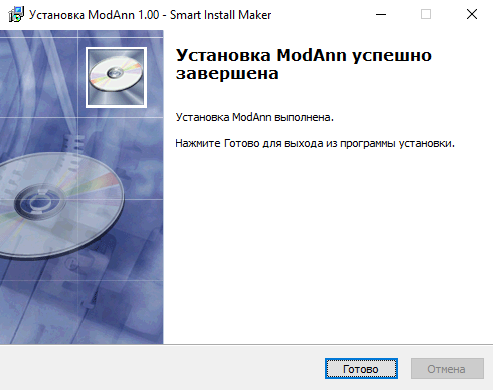


Рисунок 15 – Все готово

**5.3 Выполнения программы**

**5.3.1 Запуск программы**

Данную программу можно запустить различными способами.

Первым из них является запуск с помощью ярлыка на рабочем столе. Необхожимо дважды клинуть правой кнопкой мышки на ярлыке под названием ModAnn (рисунок 16).



Рисунок 16 – Ярлык приложения ModAnn

Вторым способом является запуск через Пуск. Необходимо нажать Пуск\Все программы\ModAnn\ModAnn.

Третьим способ является запуск из каталога, в которое устанавливалось приложение (По умолчанию С:\Program Files\ModAnn).

**5.3.2 Инструкция по работе с программой**

После запуска программы на экране появляется заставка (рисунок 17).

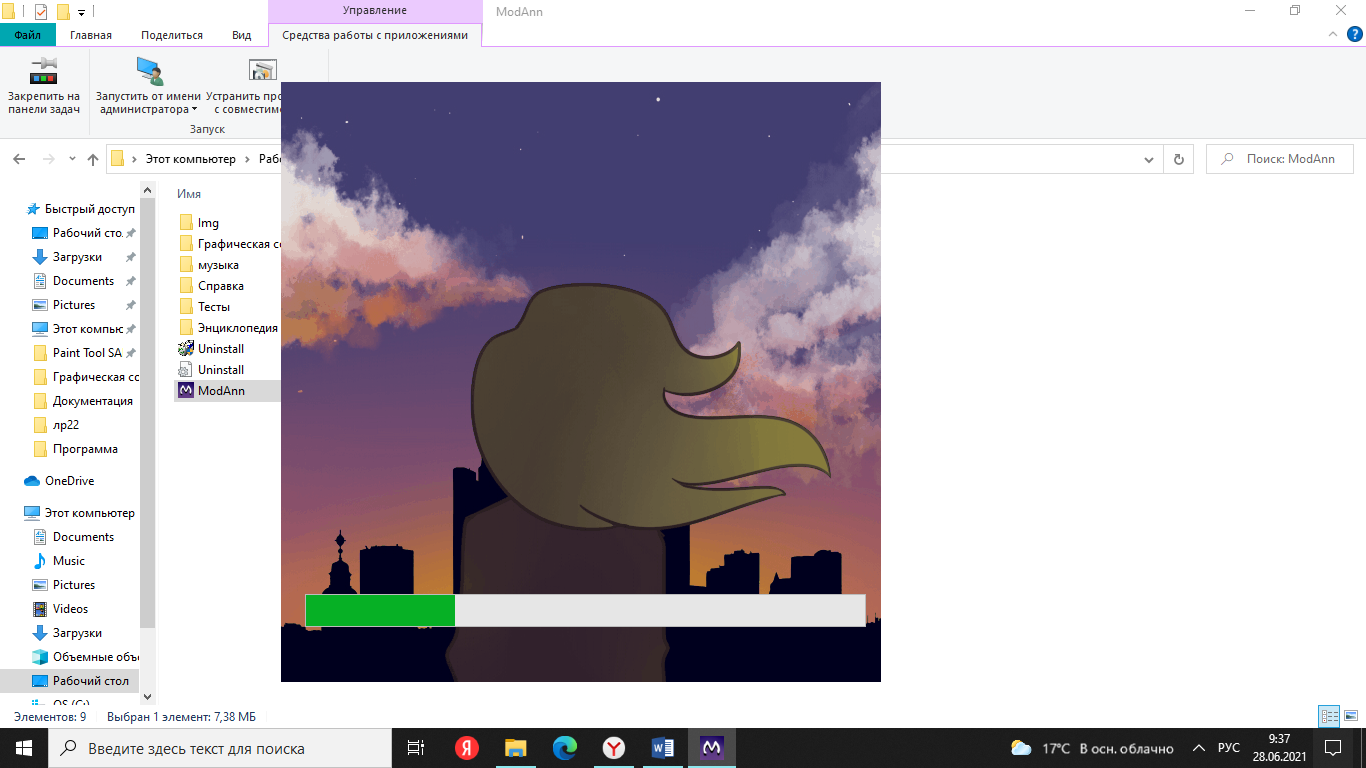


Рисунок 17 – заставка приложения

После заставки появляется главная форма (рисунок 18) откуда польователь может уже перейти в любую часть самого приложения.

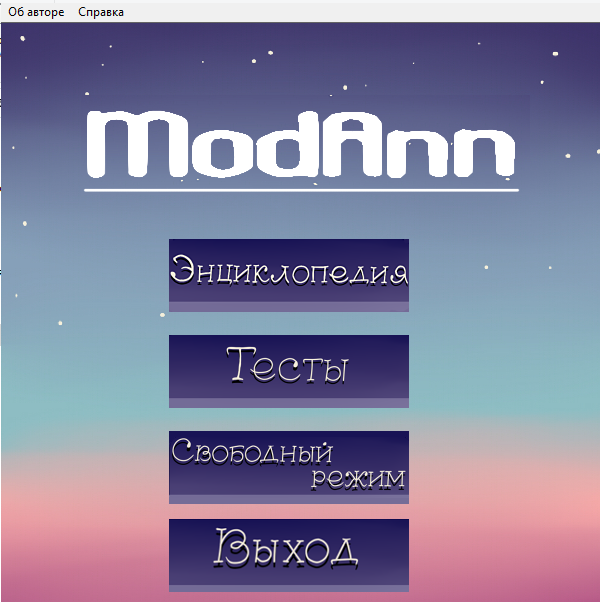


Рисунок 18 – главное меню

Далее у пользователя появляется возможность выбора того, что он хочет делать в этом приложении.

При нажатии на кнопку «энциклопедия» пользователя перенесет на html-страницы, где предоставлена информация об стилях в одежде. На главное форме эниклопедии есть кнопки для перехода на различные статьи (рисунок 19). Далее можно, после прочтения и ознакомления со статьей, перейти на следующую\предыдущую страницу или выйти в главное меню энциклопедии (рисунок 20).

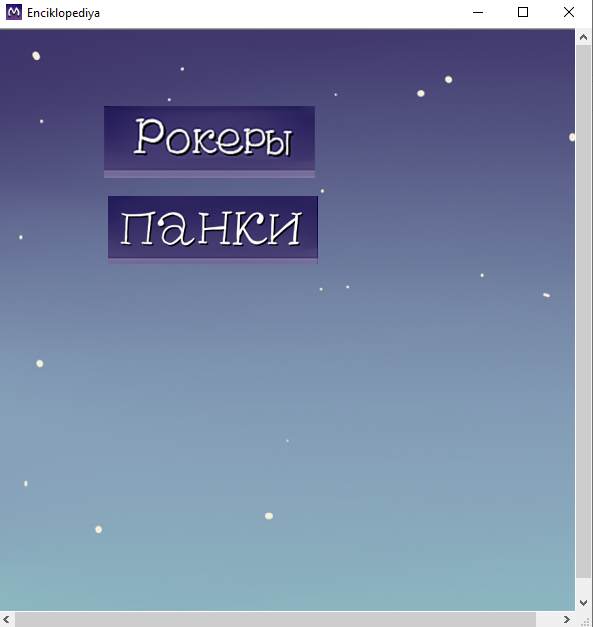


Рисунок 19 – главное меню энциклопедии

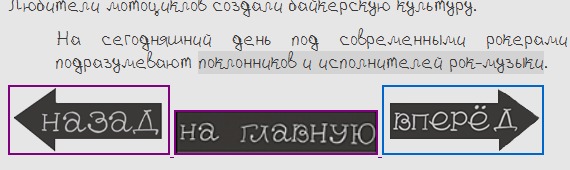


Рисунок 20 – переход на другие статьи

Далее существует форма с тестами. Для их перехода нужно нажать на кнопку «тесты». В появившемся окне можно выбрать любой из предложенных тестов, а затем ответить на вопросы (рисунок 21). Для выхода в главное меню нужно нажать на кнопку «назад». Для повторного прохождения следует перезапустить приложение.

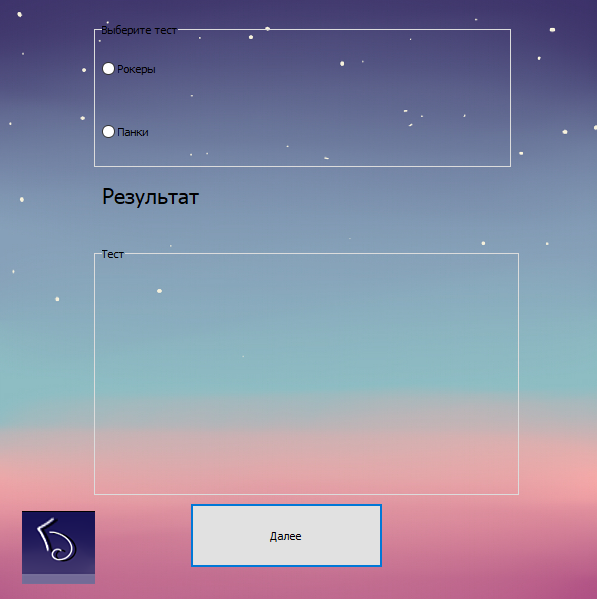


Рисунок 21 – Тесты

«Свободный режим» доступен после нажатия на кнопку «свободный режим». В нем пользователь может на свой вкус одеть куклу (рисунок 22). Для выхода из этого окна используется кнопка «назад».

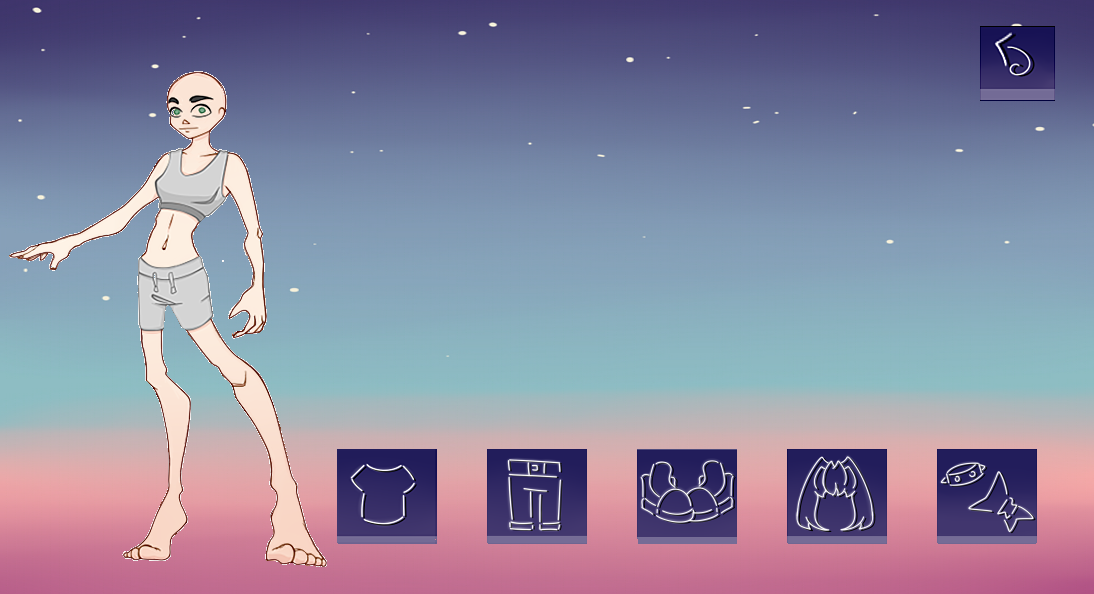


Рисунок 22 – Свободный режим

Для открытия окна «об авторе» нужно открыть пунк меню «об авторе» (рисунок 23). Там появится окно с приветствием от автора. Для выхода нужно нажать кнопку «назад».

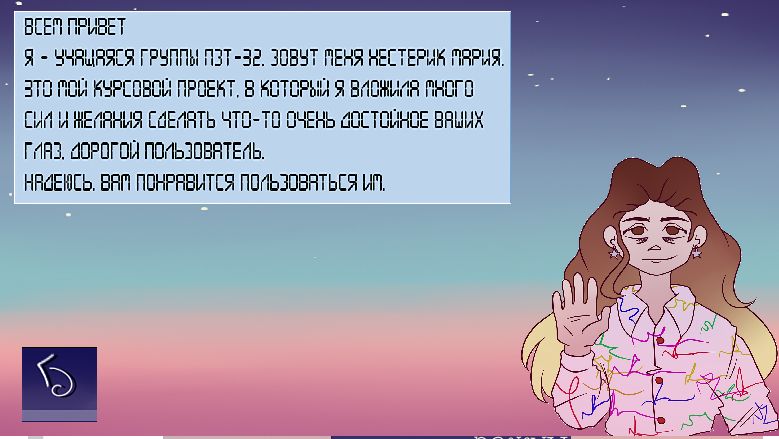


Рисунок 23 – Об авторе

**5.3.3 Завершение работы с программой**

Для выхода из приложения нужно нажать «Выход» в главном меню (рисунок 24).

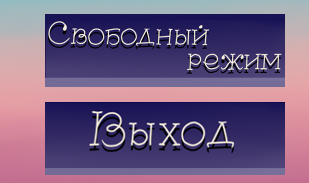


Рисунок 24 – Выход

**5.4 Использование системы справочной информации**

Справочную систему можно запустить с помощью команды меню «Справка» (рисунок 25).

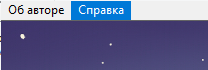


Рисунок 25 – Справка

Появится окно для ознакомления со справочной системой (рисунок 26). В ней можно ознакомится с использованием приложения «ModAnn». Нажимая на нужный пункт содержания, находящегося в левой части экрана, в правой части появляется информация относящаяся к этому пунтку. Выйти из справки можно, нажав на крестик в правом верхнем углу.

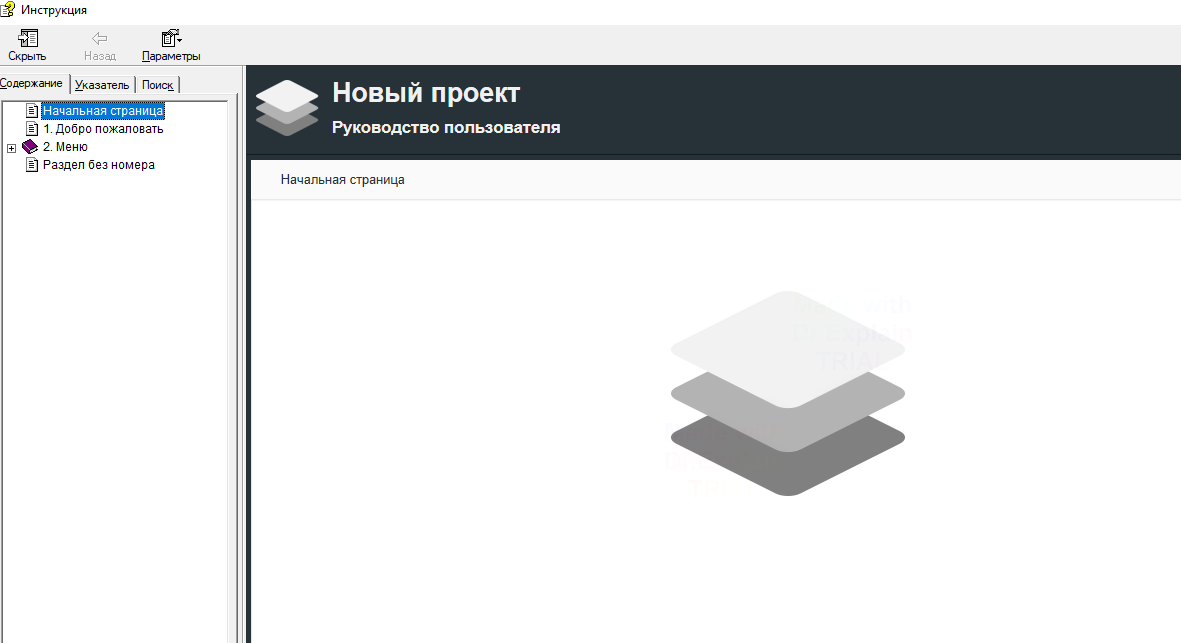


Рисунок 26 - Справка

**Заключение**

Целью данной курсовой работы была разработка программного продукта «Электронная энциклопедия моды».

В поставленной задаче был реализован простой и понятный для всех пользователях интерфейс.

Поставленная задача была выполнена не на все сто процентов. В получившемся программном продукте все ещё есть недочеты и выскакивающие ошибки, однако в будущем они будут избегаться.

Степерь соответствия задания не соответствует ожидаемому результату. В ходе разработки приложения выскакивает ошибка, над которой ведется тщательная работа на данный момент.

Однако продукт ожидает перспективное будущее: в нем можно модифицировать разделы и добавлять все новую и новую информацию. Возможность появления нового функционала и креативных идей от разработчика также является хорошим фактором для дальнейшего развития этого проекта.

Итог: продукт нуждается в доработке.

**Список использованных источников**

1 Баженова, Ю.И. «Delphi 7. Самоучитель программиста.» - М.: Изд-во КУДИЦ-«Образ ISBN: 5-93378-072-3», 2003. – 448 с.

1. Рок-музыка [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Рок-музыка>. Дата доступа 05.06.2021
2. Панк [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Панк>. Дата доступа 09.06.2021
3. Видеоролик «одевалка» [Электронный ресурс] – Режим доступа: <https://www.youtube.com/watch?v=d9ZrSnlLSl4&t=2385s>. Дата доступа 10.06.2021

**­­Приложение А**

**Листинг программы**

**unit Unit1; //Модуль заставки**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, XPMan, ExtCtrls, Menus, jpeg, ComCtrls, GIFImage;

type

TForm1 = class(TForm)

Timer1: TTimer;

ProgressBar1: TProgressBar;

Image1: TImage;

procedure Timer1Timer(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses

Unit2;

{$R \*.dfm}

procedure TForm1.Timer1Timer(Sender: TObject); // Полоса загрузки

begin

ProgressBar1.Position:=ProgressBar1.Position+27; // Сама полоса

If (ProgressBar1.Position=ProgressBar1.Max) Then

begin

form2.Show; // Показать главное меню

Form1.Hide; //Скрыть форму заставки

Timer1.Enabled:=False;

end;

end;

procedure TForm1.FormCreate(Sender: TObject); // Заставка

begin

image1.Picture.LoadFromFile('Графическая составляющая\заставка.gif');

end;

end.

**unit Unit2; // Форма главного меню**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, jpeg, ExtCtrls, Menus, ShellAPI, MMSystem;

type

TForm2 = class(TForm)

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

procedure Image3Click(Sender: TObject);

procedure Image6Click(Sender: TObject);

procedure Image4Click(Sender: TObject);

procedure Image5Click(Sender: TObject);

procedure N1Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit3, Unit4, Unit5, Unit6;

{$R \*.dfm}

procedure TForm2.Image3Click(Sender: TObject); // Открытие энциклопедии

begin

Form3.Show;

end;

procedure TForm2.Image6Click(Sender: TObject); // Закрытие всего проекта

begin

Halt;

end;

procedure TForm2.Image4Click(Sender: TObject); // Открытие теста

begin

Form4.Show;

end;

procedure TForm2.Image5Click(Sender: TObject); // Открытие "свободного режима"

begin

Form5.Show;

end;

procedure TForm2.N1Click(Sender: TObject); // Открытие формы "об авторе"

begin

form6.Show;

end;

procedure TForm2.N2Click(Sender: TObject); // Открытие справки

begin

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

end;

procedure TForm2.FormCreate(Sender: TObject); // Открытие фоновой музыки

begin

PlaySound (PChar('музыка\lofi32.wav'),0,SND\_ASYNC Or SND\_LOOP);

end;

end.

**unit Unit3; //форма с энциклопедией**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, OleCtrls, SHDocVw;

type

TForm3 = class(TForm)

WebBrowser1: TWebBrowser;

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

{$R \*.dfm}

procedure TForm3.FormCreate(Sender: TObject); // Открытие html страниц

begin

WebBrowser1.Navigate(extractfilepath(paramstr(0))+'Энциклопедия\Main\_page.htm');

end;

end.

**unit Unit4; // Форма с тестом**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ExtCtrls, jpeg;

type

TForm4 = class(TForm)

Image1: TImage;

RadioGroup1: TRadioGroup;

Label1: TLabel;

RadioGroup2: TRadioGroup;

Button1: TButton;

Image2: TImage;

procedure Button1Click(Sender: TObject);

procedure RadioGroup1Click(Sender: TObject);

procedure Image2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

f: text;

s: string;

Nvern, ball: integer;

implementation

uses Unit2;

{$R \*.dfm}

procedure TForm4.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 ответа верный

RadioGroup2.Items.Clear; //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);

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:='Вы ответили на '+IntToStr(ball)+' из 4 вопросов'; //Вывод кол-ва баллов

CloseFile(f);

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

end;

end;

procedure TForm4.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 ,'Тесты\t.txt');//связывается с разными файлами

1: AssignFile( f ,'Тесты\t2.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;

procedure TForm4.Image2Click(Sender: TObject);

begin

form4.Hide;

form2.show;

end;

end.

**unit Unit5; // Открытие "свободного режима"**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, jpeg, ExtCtrls;

type

TForm5 = class(TForm)

Timer1: TTimer;

Image1: TImage;

procedure FormCreate(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

procedure FormMouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure FormCanResize(Sender: TObject; var NewWidth,

NewHeight: Integer; var Resize: Boolean);

procedure Image1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

Buf:TBitmap;

path:string;

Img: array [0..21] of TBitmap;

KnImg: array [1..5] of TBitmap;

clothes:array [1..20] of Boolean;

Vverx: array [1..5] of integer;

implementation

uses Unit2;

{$R \*.dfm}

procedure TForm5.FormCreate(Sender: TObject); // Загрузка изображений

var i:integer;

begin

Path:=ExtractfileDir(Application.exeName); // указание путя

Buf:= TBitmap.Create;

Buf.Width:=1100; // Масштабирование формы

Buf.Height:=600;

for i:= 0 to 20 do

begin

Img[i]:=TBitmap.Create;

Img[i].Transparent:=true;

Img[i].LoadFromFile(path+'\Img\'+inttostr(i)+'.bmp'); // Занесение одежды в массив

end;

for i:= 1 to 5 do

begin

KnImg[i]:=TBitmap.Create;

KnImg[i].Transparent:=true;

KnImg[i].LoadFromFile(path+'\Img\kn'+inttostr(i)+'.bmp'); // Занесение кнопок в массив

end;

for i:= 1 to 21 do // Прозрачная одежда при первом запуске

clothes[i]:=false;

Vverx[1]:=1;

Vverx[2]:=5;

Vverx[3]:=9;

Vverx[4]:=13;

Vverx[5]:=17;

end;

procedure TForm5.Timer1Timer(Sender: TObject); // Загрузка фона, куклы

var i:integer;

begin

Buf:=TBitMap.Create;

Buf.LoadFromFile('Графическая составляющая\fon.bmp');

Buf.Canvas.Draw(0,0,Img[0]);

for i:= 1 to 5 do

Buf.Canvas.Draw(i\*150+190,450,KnImg[i]);

for i:= 1 to 20 do

if clothes[i]=true then Buf.Canvas.Draw(0,0,Img[i]);

Form5.Canvas.Draw(0,0,Buf);

end;

procedure TForm5.FormMouseDown(Sender: TObject; Button: TMouseButton; // Реализация кнопок

Shift: TShiftState; X, Y: Integer);

begin

Form5.caption:=inttostr(X);

if (X>=340) and (X<=440) and (Y>=450) and (Y<=545) then // Кнопка с верхней одеждой

begin

clothes[Vverx[1]]:=false;

inc(Vverx[1]);

if Vverx[1]>4 then Vverx[1]:=1;

clothes[Vverx[1]]:=true;

end;

if (X>=490) and (X<=540) and (Y>=450) and (Y<=545) then // Кнопка с нижней одеждой

begin

clothes[Vverx[2]]:=false;

inc(Vverx[2]);

if Vverx[2]>8 then Vverx[2]:=5;

clothes[Vverx[2]]:=true;

end;

if (X>=640) and (X<=740) and (Y>=450) and (Y<=545) then // Кнопка с обувью

begin

clothes[Vverx[3]]:=false;

inc(Vverx[3]);

if Vverx[3]>12 then Vverx[3]:=9;

clothes[Vverx[3]]:=true;

end;

if (X>=790) and (X<=890) and (Y>=450) and (Y<=545) then // Кнопка с прической

begin

clothes[Vverx[4]]:=false;

inc(Vverx[4]);

if Vverx[4]>16 then Vverx[4]:=13;

clothes[Vverx[4]]:=true;

end;

if (X>=940) and (X<=1040) and (Y>=450) and (Y<=545) then // Кнопка с аксекссуарами

begin

clothes[Vverx[5]]:=false;

inc(Vverx[5]);

if Vverx[5]>20 then Vverx[5]:=17;

clothes[Vverx[5]]:=true;

end;

end;

procedure TForm5.FormCanResize(Sender: TObject; var NewWidth,

NewHeight: Integer; var Resize: Boolean);

begin

Resize := False;

end;

procedure TForm5.Image1Click(Sender: TObject);

begin

form5.Hide;

form2.show;

end;

end.

unit Unit6; // Открытие формы «об авторе»

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls;

type

TForm6 = class(TForm)

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

procedure Image4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form6: TForm6;

implementation

{$R \*.dfm}

procedure TForm6.Image4Click(Sender: TObject); // Кнопка «назад»

begin

form6.Hide;

end;

end.

­