**Введение**

На курсовое проектирование была поставлена задача, разработки программы на тему: Разработка познавательного приложения «Плавание»

Цель курсового проекта заключается в знакомстве массового пользователя с видами плавания, спортсменами, которые поставили рекорды, нормативами Республики Беларусь по плаванию, а также интересные факты о плавании.

Создаваемая программа будет рассчитана на любого рода пользователей. Применить данную программу смогут не только люди, которые занимаются профессиональным спортом, но и просто люди, заинтересовавшиеся в данной теме.

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 ею 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 создаётся 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 познавательного приложения «Плавание»

Плавание - это один из самых полезных видов спорта, который не только укрепляет здоровье, но и приносит удовольствие. Плавание также является одним из немногих видов спорта, который подходит для людей всех возрастов и уровней физической подготовки. Оно помогает улучшить работу сердечно-сосудистой системы, укрепить мышцы и суставы, а также повысить выносливость.

Кроме того, плавание - это отличный способ расслабиться и снять стресс. Во время плавания человек находится в воде, которая создает ощущение легкости и свободы движений. Это помогает снять напряжение и улучшить настроение.

Одним из главных преимуществ плавания является то, что оно является низкоударным видом спорта. Это означает, что он не нагружает суставы и кости так жестко, как другие виды физической активности. Поэтому плавание может быть особенно полезным для людей с проблемами со здоровьем или травмами.

А также, плавание является отличным способом улучшить свое кардиоваскулярное здоровье. Во время плавания сердце работает более интенсивно, чтобы обеспечить кровью все органы и мышцы тела. Это помогает укрепить сердечную мышцу и улучшить ее функционирование.

Еще одним преимуществом плавания является его доступность. Для занятий этим видом спорта не требуется дорогостоящее оборудование или специальная подготовка. Достаточно иметь доступ к бассейну или открытой воде. Однако, как и любой другой вид спорта, плавание может быть опасным без должной осторожности и знаний правил безопасности. Перед началом занятий необходимо проконсультироваться с врачом и инструктором по плаванию.

Кроме того, плавание может быть очень приятным занятием. Во время плавания мы чувствуем легкость и свободу движений, что помогает расслабиться и снять стресс. Также плавание - отличный способ провести время.

Изучение познавательного приложения «Плавание» может понадобиться  многим обычным пользователям, которые заинтересованы в этом спорте и хотят узнать больше информации о нем, также приложение поможет улучшить технику плавания и дать пользователю мотивацию.

Аналогов для такого приложения можно привести не так уж и много. В  основном в сети интернет находятся различного рода статьи по даннной  тематике. В более редких случаях – ознакомительный  материал с примерами. Собранный же воедино материал со стилями плавания, спортсменами, которые ставят мировые рекорды и интерактивом встречается единично на просторах  интернета.

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 сочетать в себе обучающую информацию, проверку усвоенного  материала и развлечения для людей.

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.

Приложение должно 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 в теме плавания.

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.

Canva - это графический дизайнер, который позволяет пользователям создавать различные типы дизайна, включая логотипы, постеры, презентации, социальные медиа-посты, визитки и многое другое. Canva имеет простой и интуитивно понятный интерфейс, который позволяет пользователям легко создавать профессионально выглядящие дизайны без необходимости иметь опыт в графическом дизайне.

Aseprite - это графический редактор, который специализируется на создании пиксельной графики и анимации. Он предназначен для художников, которые работают с пиксельной графикой, такой как спрайты для видеоигр, и позволяет создавать анимацию кадр за кадром.

Microsoft Word 2016 – редактор текста для написания документации.

Разработка ведется на ноутбуке Lenovo. У данного ноутбука следующие

параметры:

* процессор Intel Core i5;
* объем ОЗУ 8 гб;
* объем места на SSD – 256 гб ;
* видеоподсистема 1024х768 точек с глубиной цвета 16 Bit;
* ОС – Windows 10 Pro.

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.

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: изучение информации по теме «Плавание».

Для реализации задач будут использоваться процедуры и функции. С помощью процедуры будет перемешиваться картинка в игре «Пазл».

Тест будет генерироваться на основе выбора пункта в объекте TRadioGroup. Вопросы теста будут подгружаться из txt файла.

**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:

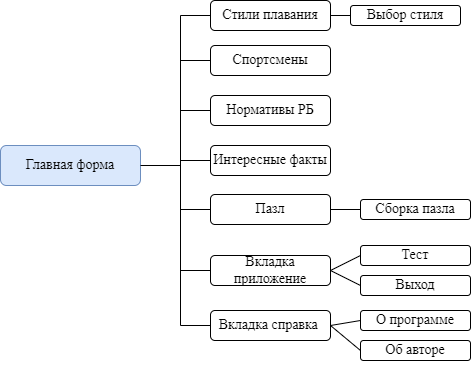


Рисунок 1 – Структура навигации по проекту

**3. Реализация**

**3.1 Реализация проекта**

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

Unit 1 - является заставкой в курсовом проекте;

Unit 2 - знакомство с Джоном;

Unit 3 - является главной формой;

Unit 4 - содержит информацию о стилях плавания;

Unit 5 - содержит информацию о спортсменах;

Unit 6 - содержит информацию о нормативах Республики Беларусь по плаванию;

Unit 7 - содержит информацию об интересных фактах;

Unit 8 - тестовые задания;

Unit 9 - содержит игру «Пазл».

**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 Tzastavka.Timer1Timer(Sender: TObject); | | Unit 1 | | Timer1 | | Переход на следующую форму по истечению времени | |
| 2 procedure Tznakomstvo\_with\_Jon.Image3MouseDown(Sender: TObject;  Button: TMouseButton; Shift: TShiftState; X, Y: Integer); | | Unit 2 | | Image3 | | Переход на следующую форму | |
| 3 procedure Tglavnoe\_menu.Image2MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | | Unit 3 | | Image2 | | Переход на форму со стилями плавания | |
| Продолжение таблицы 1 | |  | |  | |  | |
| 1 | | 2 | | 3 | | 4 | |
| 4 procedure Tglavnoe\_menu.Image3MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | | Unit 3 | | Image3 | | Переход на форму со спортсменами | |
| 5 procedure Tglavnoe\_menu.Image4MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | | Unit 3 | | Image4 | | Переход на форму с нормативами Республики Беларусь | |
| 6 procedure Tglavnoe\_menu.Image5MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | | Unit 3 | | Image5 | | Переход на форму с интересными фактами | |
| 7 procedure Tglavnoe\_menu.Image7MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | | Unit 3 | | Image7 | | Переход на форму с игрой «Пазл» | |
| 8 procedure Tglavnoe\_menu.N2Click(Sender: TObject); | | Unit 3 | | MainMenu1 | | Переход на форму с тестовыми заданиями и итоговым ответом | |
| 9 procedure Tglavnoe\_menu.N4Click(Sender: TObject); | | Unit 3 | | MainMenu1 | | Закрытие программы | |
| 10 procedure Tglavnoe\_menu.N5Click(Sender: TObject); | | Unit 3 | | MainMenu1 | | Переход на справочную систему | |
| 11 procedure Tstili\_plavaniya.Image2MouseDown(Sender: TObject;  Button: TMouseButton; Shift: TShiftState; X, Y: Integer); | | Unit 4 | | Image2 | | Отображение веб-ресурса «Кроль» | |
| 12 procedure Tstili\_plavaniya.Image3MouseDown(Sender: TObject;  Button: TMouseButton; Shift: TShiftState; X, Y: Integer); | | Unit 4 | | Image3 | | Отображение веб-ресурса «Брасс» | |
|  | |  | |  | |  | |
|  | |  | |  | |  | |
| Продолжение таблицы 1 | |  | |  | |  | |
| 1 | | 2 | | 3 | | 4 | |
| 13 procedure Tstili\_plavaniya.Image4MouseDown(Sender: TObject;  Button: TMouseButton; Shift: TShiftState; X, Y: Integer); | | Unit 4 | | Image4 | | Отображение веб-ресурса «Баттерфляй» | |
| 14 procedure Tstili\_plavaniya.Image5MouseDown(Sender: TObject;  Button: TMouseButton; Shift: TShiftState; X, Y: Integer); | | Unit 4 | | Image5 | | Отображение веб-ресурса «На спине» | |
| 15 procedure Tstili\_plavaniya.N2Click(Sender: TObject); | | Unit 4 | | MainMenu1 | | Переход на главное меню | |
| 16 procedure Tstili\_plavaniya.N4Click(Sender: TObject); | | Unit 4 | | MainMenu1 | | Выход из программы | |
| 17 procedure Tsportsmen.FormCreate(Sender: TObject); | | Unit 5 | | - | | Отображение веб-ресурсов | |
| 18 procedure Tsportsmen.N2Click(Sender: TObject); | | Unit 5 | | MainMenu1 | | Переход на главное меню | |
| 19 procedure Tsportsmen.N4Click(Sender: TObject); | | Unit 5 | | MainMenu1 | | Закрытие программы | |
| 20 procedure Tnormativy.FormCreate(Sender: TObject); | | Unit 6 | | - | | Отображение веб-ресурсов | |
| 21 procedure Tnormativy.N2Click(Sender: TObject); | | Unit 6 | | MainMenu1 | | Переход на главную форму | |
| 22 procedure Tnormativy.N4Click(Sender: TObject); | | Unit 6 | | MainMenu1 | | Закрытие программы | |
| 23 procedure Tinterest\_facts.FormCreate(Sender: TObject); | | Unit 7 | | - | | Очищение Memo1 | |
| 24 procedure Tinterest\_facts.Image1MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | | Unit 7 | | Image1 | | Вывод рандомного интересного факта | |
| 25 procedure Tinterest\_facts.N2Click(Sender: TObject); | | Unit 7 | | MainMenu1 | | Переход на главную форму | |
| 26 procedure Tinterest\_facts.N4Click(Sender: TObject); | | Unit 7 | | MainMenu1 | | Закрытие программы | |

|  |  |  |  |
| --- | --- | --- | --- |
| Продолжение таблицы 1 |  |  |  |
| 1 | 2 | 3 | 4 |
| 27 procedure Ttest.FormCreate(Sender: TObject); | Unit 8 | - | Текст - невидимый |
| 28 procedure Ttest.Image1MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | Unit 8 | Image1 | Прохождение теста |
| 29 procedure Ttest.N2Click(Sender: TObject); | Unit 8 | MainMenu1 | Переход на главную форму |
| 30 procedure Ttest.N4Click(Sender: TObject); | Unit 8 | MainMenu1 | Закрытие программы |
| 31 procedure Ttest.RadioGroup1Click(Sender: TObject); | Unit 8 | RadioGroup1 | Выбор варианта |
| 32 procedure Tpazl.FormCreate(Sender: TObject); | Unit 9 | - | Загрузка картинки |
| 33 procedure Tpazl.Image2MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | Unit 9 | Image2 | Выводится исходный рисунок |
| 34 procedure Tpazl.Image3MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | Unit 9 | Image3 | Устанавливаются координаты и границы фрагментов |
| 35 procedure Tpazl.Image5MouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer); | Unit 9 | Image5 | Проверяется, правильно ли собран пазл |
| 36 procedure Tpazl.N2Click(Sender: TObject); | Unit 9 | MainMenu1 | Переход на главную форму |
| 37 procedure Tpazl.N4Click(Sender: TObject); | Unit 9 | MainMenu1 | Закрытие программы |
| 38 procedure Tpazl.PaintBox2MouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer); | Unit 9 | PaintBox2 | Перемещение фрагментов |

|  |  |  |  |
| --- | --- | --- | --- |
| Продолжение таблицы 1 |  |  |  |
| 1 | 2 | 3 | 4 |
| 39 procedure XYtoUV(X, Y: Integer; var u, v: Integer); | Unit 9 | - | Находится номер строки и столбца |

**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 |
| 1 | 2 | 3 |
| TTimer | zastavka | Используется для перехода на другую форму через определенный промежуток времени |
| TImage | zastavka, znakomstvo\_with\_Jon, glavnoe\_menu, stili\_plavaniya, sportsmen, normativy, interest\_facts, test, pazl | Используется как фон, как отображение графического изображения, как кнопки |
| TLabel | znakomstvo\_with\_Jon, glavnoe\_menu, stili\_plavaniya, interest\_facts, test | Отображение надписей на форме |
| TPanel | glavnoe\_menu | Используется для группировки компонентов |
| TMainMenu | glavnoe\_menu, stili\_plavaniya, sportsmen, normativy, interest\_facts, test, pazl | Используется для создания меню проекта |
| TMenuItem | glavnoe\_menu, stili\_plavaniya, sportsmen, normativy, interest\_facts, test, pazl | Используется как подменю(элемент меню) |
| Продолжение таблицы 2 |  |  |
| 1 | 2 | 3 |
| TPageControl | sportsmen, normativy | Используется как многостраничная форма |
| TTabSheet | sportsmen, normativy | Используются как закладки для многостраничной формы |
| TMemo | interest\_facts | Используется как текстовый редактор |
| TRadioGroup | test | Используется для выбора одного из нескольких взаимоисключающих вариантов |
| TPaintBox | pazl | Используется для создания пользовательских элементов управления и графических изображений |
| TMediaPlayer | pazl | Используется для воспроизведения аудио и видео файлов в приложении |
| TOpenDialog | pazl | Используется для выбора файлов, которые будут использоваться в приложении |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

|  |  |
| --- | --- |
| Имя файла | Назначение |
| 1 | 2 |
| Project1.exe | Исполняемый файл проекта, используется для запуска программы на выполнение. |
| Project1.dproj | Файл проекта, связывает все файлы из которых состоит приложение. |
|  |  |
|  |  |
| Продолжение таблицы 3 | |
| 1 | 2 |
| Unit1.pas | Файл программного модуля заставки |
| Unit2.pas | Файл программного модуля знакомства с Джоном |
| Unit3.pas | Файл программного модуля главной формы |
| Unit4.pas | Файл программного модуля описания стилей плавания |
| Unit5.pas | Файл программного модуля информации о спорстменах |
| Unit6.pas | Файл программного модуля информации о нормативых РБ |
| Unit7.pas | Файл программного модуля информации об интересных фактах |
| Unit8.pas | Файл программного модуля для прохождения теста |
| Unit9.pas | Файл программного модуля игры «Пазл» |
| Unit1.dfm | Форма с заставкой |
| Unit2.dfm | Форма знакомства с Джоном |
| Unit3.dfm | Главная форма |
| Unit4.dfm | Форма со стилями плавания |
| Unit5.dfm | Форма со спортсменами |
| Unit6.dfm | Форма с нормативами РБ |
| Unit7.dfm | Форма с интересными фактами о плавании |
| Unit8.dfm | Форма с тестом |
| Unit9.dfm | Форма с игрой «Пазл» |
| 0.txt | Интересный факт |
| 1.txt | Интересный факт |
| 2.txt | Интересный факт |
| 3.txt | Интересный факт |
| 4.txt | Интересный факт |
| 5.txt | Интересный факт |
| 6.txt | Интересный факт |
| 7.txt | Интересный факт |
| 8.txt | Интересный факт |
| 9.txt | Интересный факт |
| 10.txt | Интересный факт |
| 11.txt | Интересный факт |
| 12.txt | Интересный факт |
| 13.txt | Интересный факт |
| 14.txt | Интересный факт |
| fail.mp3 | Музыка для результата |
| win.mp3 | Музыка для результата |
| Продолжение таблицы 3 | |
| 1 | 2 |
| ж\_25.htm | Теоретический материал |
| ж\_50.htm | Теоретический материал |
| м\_25.htm | Теоретический материал |
| м\_50.htm | Теоретический материал |
| юноши.htm | Теоретический материал |
| Евгений Цуркин.htm | Теоретический материал |
| Илья Шиманович.htm | Теоретический материал |
| Баттерфляй.htm | Теоретический материал |
| Брасс.htm | Теоретический материал |
| Кроль.htm | Теоретический материал |
| На\_спине.htm | Теоретический материал |
| help.chm | Справочная информация |
| setup.exe | Установочный пакет программы |
| variant1.txt | Тест |
| variant2.txt | Тест |
| Приспособление.bmp | Изображение |
| 1.ico | Иконка |

**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 | 2 | | 3 | 4 | 5 |
| 1 | Проверка входа  пользователя в приложение | | Открытие заставки | Открытие заставки | Выполнено |
| 2 | Проверка кнопки  «Продолжить» | | Открытие главной формы | Открытие главной формы | Выполнено |
| 3 | Проверка кнопки «Стили плавания» | | Открытие формы со стилями плавания | Открытие формы со стилями плавания | Выполнено |
| 4 | Проверка кнопки «Спортсмены» | | Открытие формы со спортсменами | Открытие формы со спортсменами | Выполнено |
| 5 | Проверка кнопки «Нормативы РБ» | | Открытие формы с нормативами РБ | Открытие формы с нормативами РБ | Выполнено |
| 6 | Проверка кнопки «Интересные факты» | | Открытие формы с интересными фактами | Открытие формы с интересными фактами | Выполнено |
| 7 | Проверка кнопки «Пазл» | | Открытие формы с игрой «Пазл» | Открытие формы с игрой «Пазл» | Выполнено |
| 8 | Проверка главного меню на кнопку «Тест» | | Открытие формы с тестом | Открытие формы с тестом | Выполнено |
| 9 | Проверка главного меню на кнопку «Выход» | | Закрытие приложения | Закрытие приложения | Выполнено |
|  |  | |  |  |  |
| Продолжение таблицы 4 | |  |  |  |  |
| 1 | 2 | | 3 | 4 | 5 |
| 11 | Проверка главного меню на кнопку «Меню» | | Открытие главной формы | Открытие главной формы | Выполнено |
| 12 | Проверка кнопки  «Кроль» | | Загрузка веб-ресурса | Загрузка веб-ресурса | Выполнено |
| 13 | Проверка кнопки  «Брасс» | | Загрузка веб-ресурса | Загрузка веб-ресурса | Выполнено |
| 14 | Проверка кнопки  «Баттерфляй» | | Загрузка веб-ресурса | Загрузка веб-ресурса | Выполнено |
| 15 | Проверка кнопки  «На спине» | | Загрузка веб-ресурса | Загрузка веб-ресурса | Выполнено |
| 16 | Проверка кнопки  «Узнать» | | Загрузка текстовых файлов | Загрузка текстовых файлов | Выполнено |
| 17 | Проверка кнопки  «Далее» | | Переход на следующий вопрос | Переход на следующий вопрос | Выполнено |
| 18 | Проверка кнопки  «Показать картинку» | | Загрузка картинки | Загрузка картинки | Выполнено |
| 19 | Проверка кнопки  «Перемешать» | | Перемешка картинки | Перемешка картинки | Выполнено |
| 20 | Проверка кнопки  «Проверить» | | Проверка правильности сборки | Проверка правильности сборки | Выполнено |

При разработке программного продукта было решено множество проблем, например, неправильно отображались веб-ресурсы, справочная система открывалась некорректно.

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

**5. Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Цель данного проекта заключается в знакомстве массового пользователя с видами плавания, спортсменами, которые поставили рекорды, нормативами Республики Беларусь по плаванию, а также интересные факты о плавании.

Создаваемое программа будет рассчитана на любого рода пользователей. Применить данную программу смогут не только люди, которые занимаются профессиональным спортом, но и просто люди, заинтересовавшиеся в данной теме.

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

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

* процессор Intel Core i5 1235U 1.3 ГГц (4.4 ГГц, в режиме Turbo)
* объем ОЗУ 8Гб;
* операционная система Windows 10 Pro.

**5.2 Инсталляция**

Для того, чтобы установить программу необходимо запустить файл Setup.exe. Появится окно установки приложения «Плавание».

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 «Плавание»

Вторым способом является запуск из каталога, в который устанавливалось приложение (по умолчанию C:\Program Files (x86)\Плавание).

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

**5.3.2 Инструкции по работе с программой**

После запуска приложения на экране нас встречает заставка, которая представлена на рисунке 2.



Рисунок 2 - Заставка

При истечении времени, заставка закрывается и появляется новая форма – знакомство с Джоном. Форма изображена на рисунке 3.

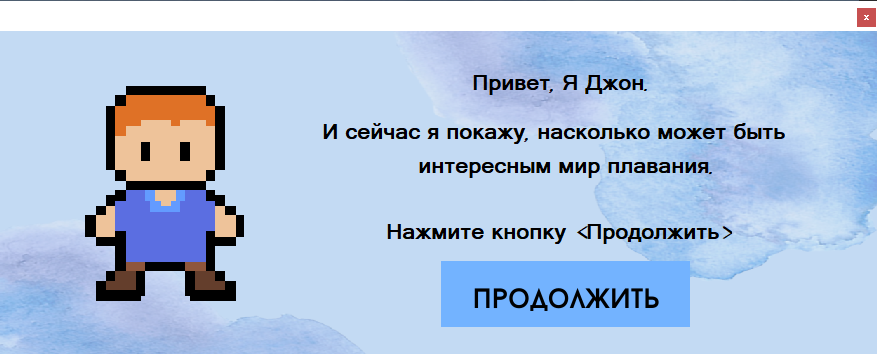


Рисунок 3 - Знакомство с Джоном

При нажатии на кнопку «Продолжить» будет осуществлен переход на главную форму, которая проиллюстрирована на рисунке 4.

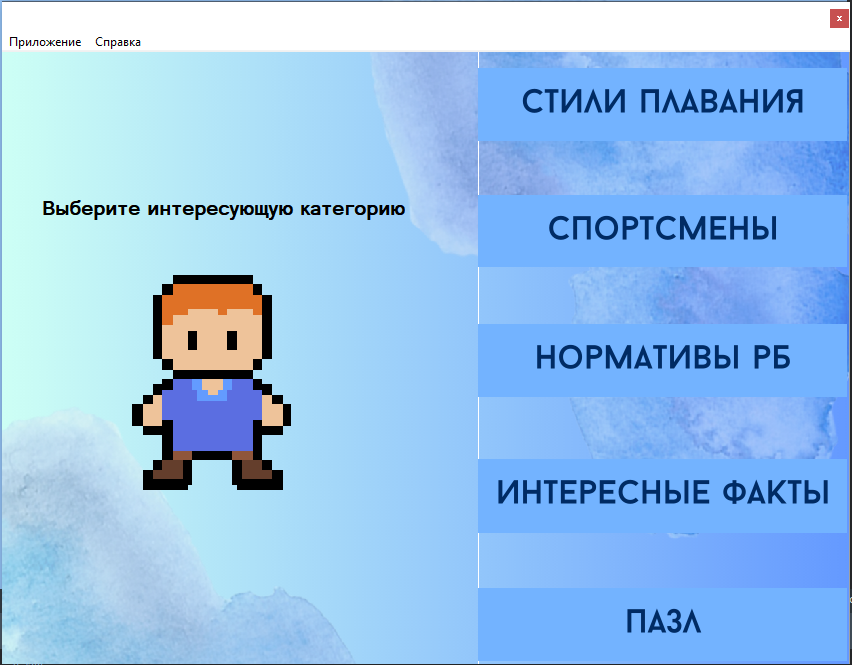


Рисунок 4 - Главная форма

При нажатии на кнопку “Приложение” в главном меню, можно увидеть кнопку «Тест» при нажатии на которую происходит переход на новую форму, которая запечатлена на рисунке 5.

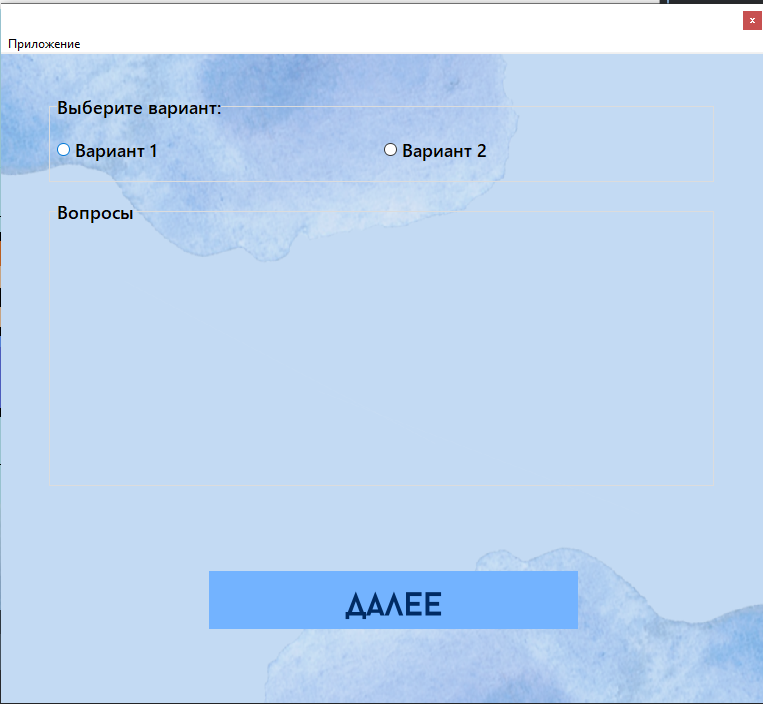


Рисунок 5 - Тест

Вернуться назад можно через главное меню. Вернувшись назад на главную форму, нажав кнопку «Стили плавания» появляется форма, изображённая на рисунке 6.

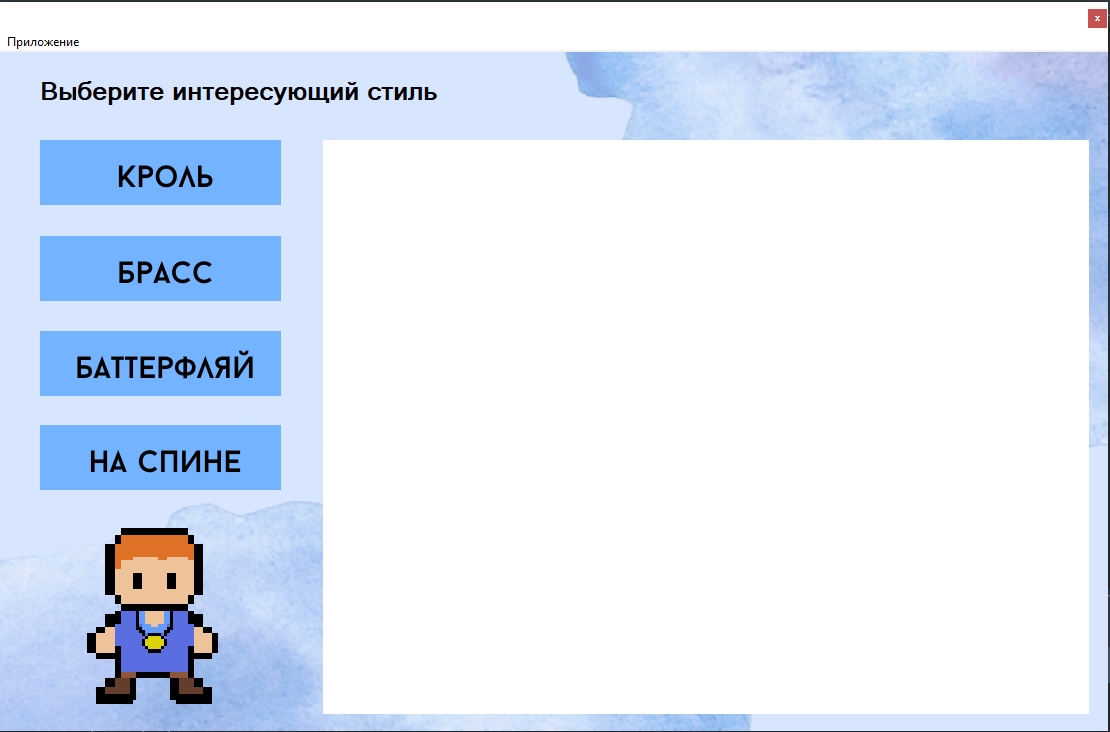


Рисунок 6 - Стили плавания

Возвратившись обратно через главное меню, нажав клавишу «Спортсмены» возникает форма, представленная в рисунке 7.



Рисунок 7 - Спортсмены

Вернувшись на главную форму через главное меню, нажав клавишу «Нормативы РБ» возникает форма, представленная в рисунке 8.

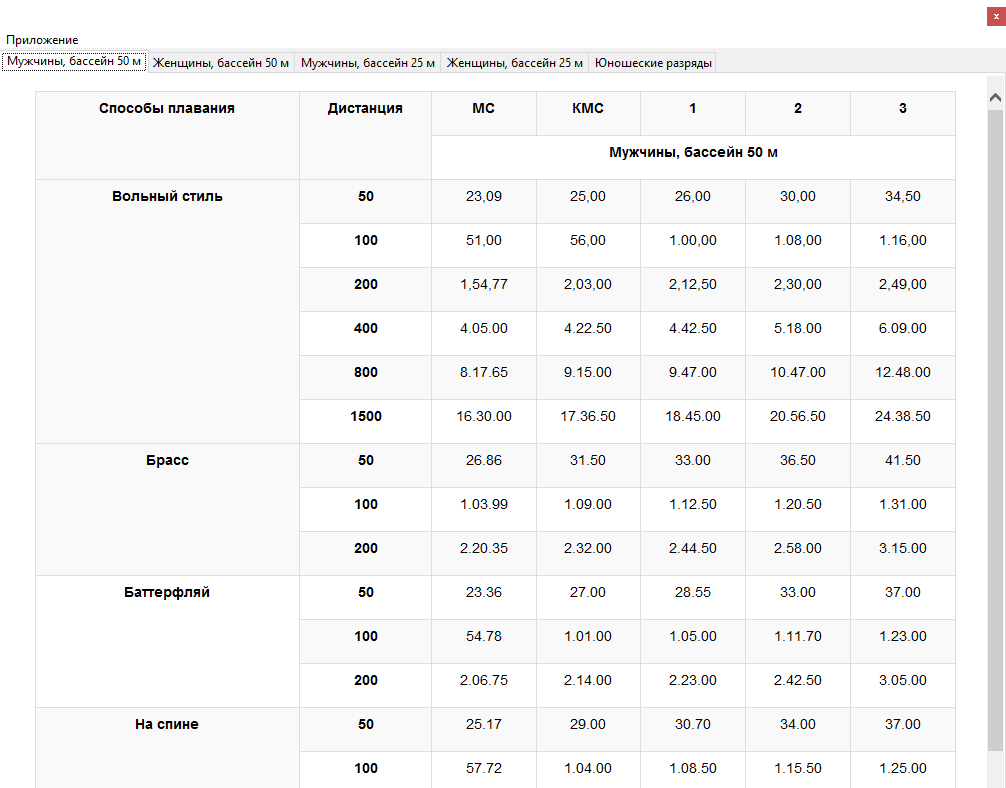


Рисунок 8 - Нормативы РБ

Возвратившись на главную форму, при нажатии на клавишу «Интересные факты» возникает форма, представленная в рисунке 9.

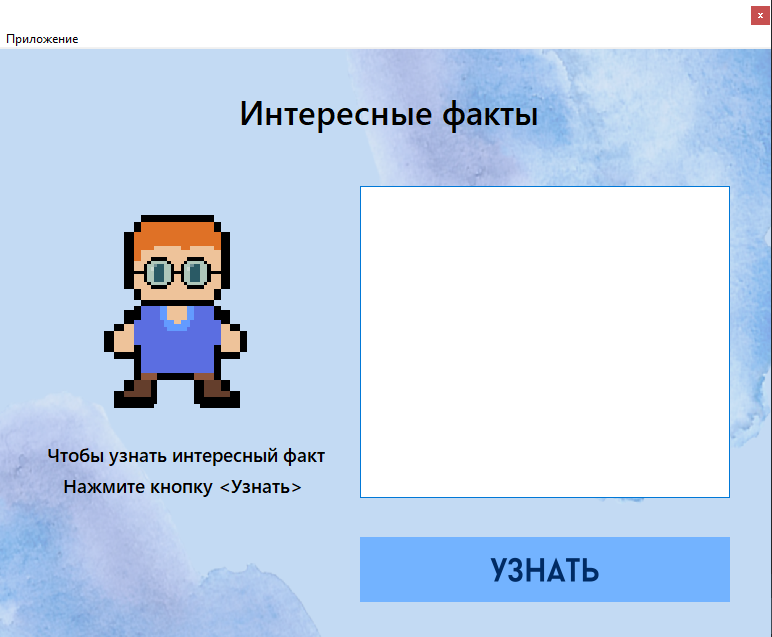


Рисунок 9 - Интересные факты

Вернувшись через главное меню на главную форму, нажав клавишу «Пазл» возникает форма, представленная в рисунке 10.

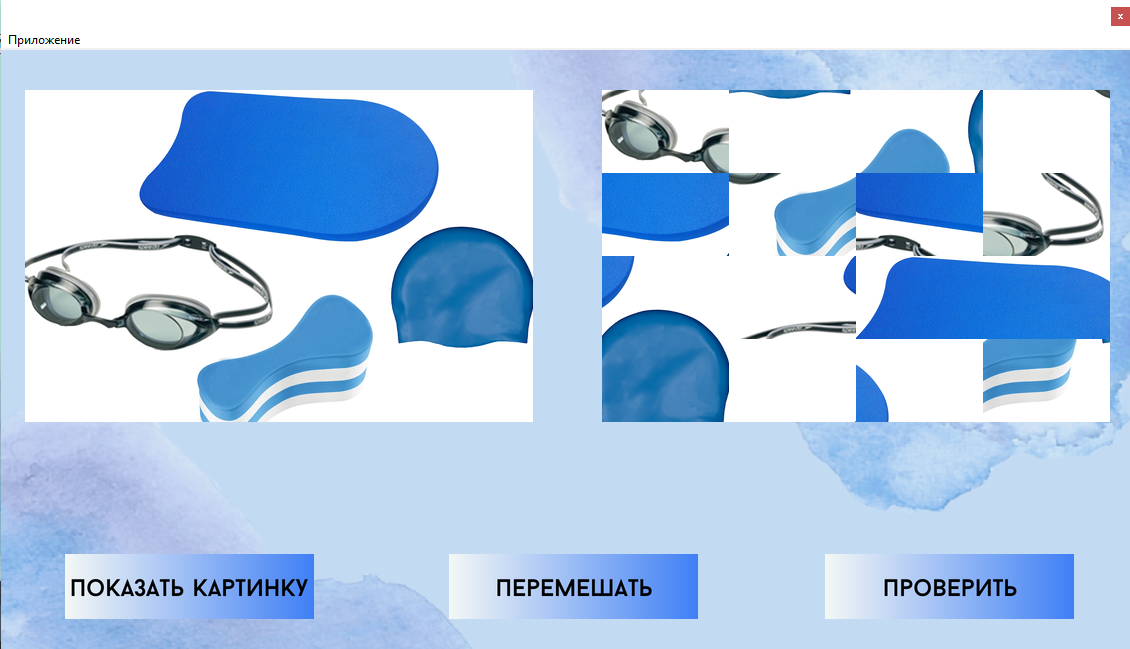


Рисунок 10 - Пазл

**5.3.3 Завершение работы с программой**

Завершить работу с программой можно несколькими способами:

1. Нужно нажать на крестик в углу формы.

2. Также можно закрыть игровое приложение нажав пункт Приложение – Выход в главном меню, который представлен на рисунке 11.

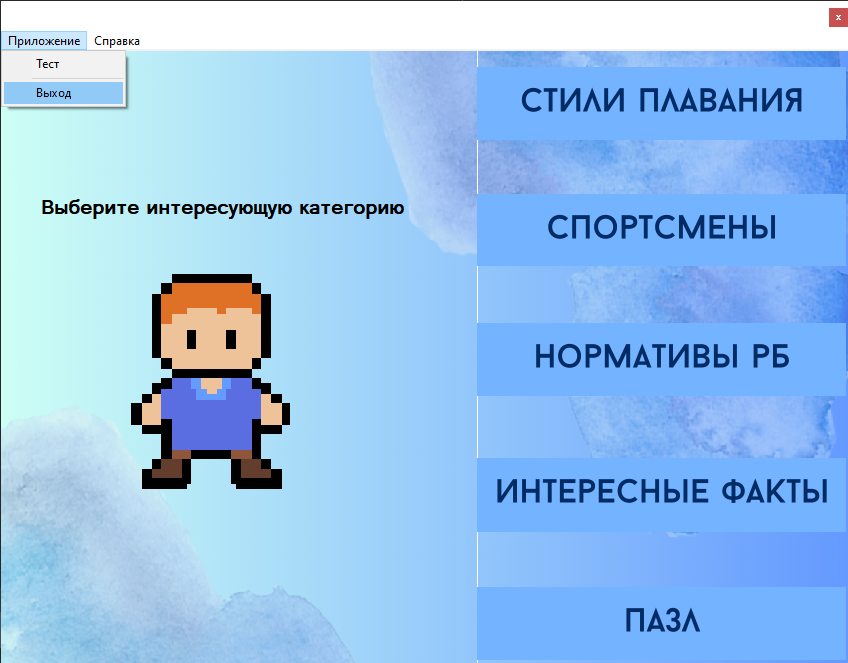


Рисунок 11 - Выход из приложения

**5.4 Использование системы справочной информации**

Справочную систему можно запустить с помощью пункта меню «Справка», изображено на рисунке 12.

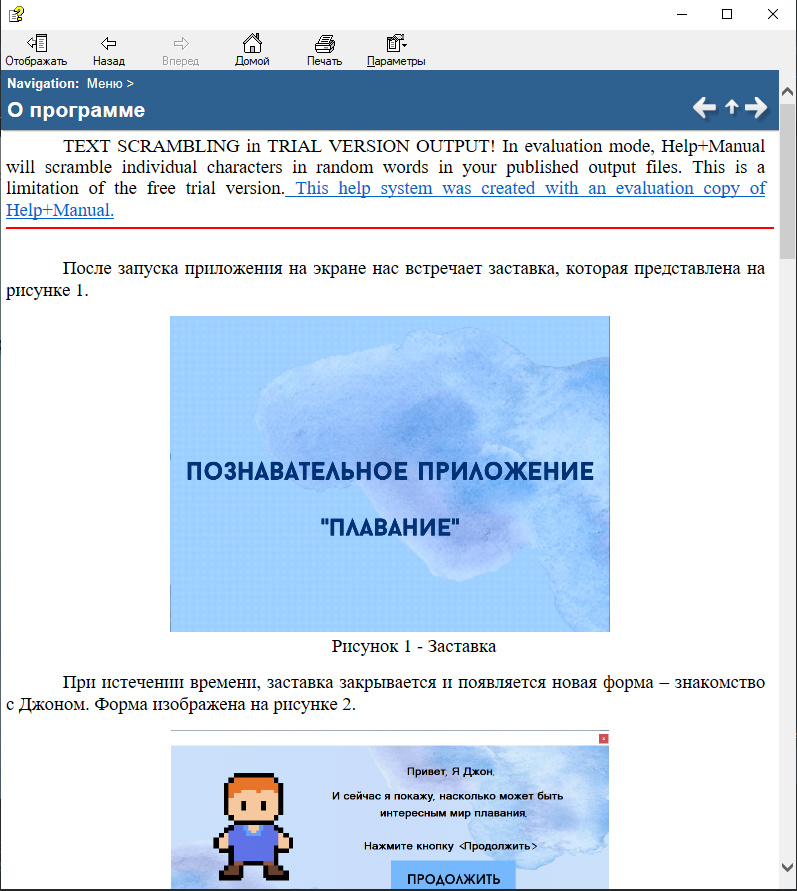


Рисунок 12 - Запуск справки

**Заключение**

Цель данного проекта заключалось в знакомстве пользователя с темой плавания.

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 научилась 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.

**Список использованных источников**

1. Разрядные нормы по плаванию для спортсменов. [Электронный ресурс]. – Режим доступа: https://swimming.by – Дата доступа: 06.06.2023.
2. 60 интересных фактов о спортивном плавании. [Электронный ресурс]. – Режим доступа: https://obshe.net – Дата доступа: 11.06.2023.
3. Спортивные стили плавания: от кроля до баттерфляя. [Электронный ресурс]. – Режим доступа: https://iplav.com – Дата доступа: 13.06.2023.
4. Архангельский, А. Я. Delphi 7. Справочное пособие. - Москва: Бином-Пресс, 2014. - 1024 с

**Приложение А**

Листинг программы

unit Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.Imaging.pngimage;

type

Tzastavka = class(TForm)

Timer1: TTimer;

Image1: TImage;

procedure Timer1Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

zastavka: Tzastavka;

implementation

{$R \*.dfm}

uses Unit2;

procedure Tzastavka.Timer1Timer(Sender: TObject);

begin

Timer1.interval := 0;

// при истечении времени заставка закроется и откроется новая форма

znakomstvo\_with\_Jon.show;

zastavka.hide;

end;

end.

unit Unit2;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

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

Vcl.ExtCtrls;

type

Tznakomstvo\_with\_Jon = class(TForm)

Image1: TImage;

Image3: TImage;

Image2: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

procedure Image3MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

private

{ Private declarations }

public

{ Public declarations }

end;

var

znakomstvo\_with\_Jon: Tznakomstvo\_with\_Jon;

implementation

{$R \*.dfm}

uses Unit3;

procedure Tznakomstvo\_with\_Jon.Image3MouseDown(Sender: TObject;

Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

begin

// при нажатии на кнопку форма закроется и откроется главная форма

znakomstvo\_with\_Jon.hide;

glavnoe\_menu.show

end;

end.

unit Unit3;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

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

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

type

Tglavnoe\_menu = class(TForm)

Panel1: TPanel;

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Image7: TImage;

Image9: TImage;

Label1: TLabel;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

procedure Image2MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Image3MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Image4MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Image5MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Image7MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure N4Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure N5Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

glavnoe\_menu: Tglavnoe\_menu;

implementation

{$R \*.dfm}

uses Unit2, unit4, Unit5, Unit6, Unit7, Unit8, Unit9;

procedure Tglavnoe\_menu.Image2MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

// при нажатии на кнопку главная форма закроется и откроется форма со стилями плавания

glavnoe\_menu.close;

stili\_plavaniya.show

end;

procedure Tglavnoe\_menu.Image3MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

// при нажатии на кнопку главная форма закроется и откроется форма со спортсменами

sportsmen.show;

glavnoe\_menu.hide

end;

procedure Tglavnoe\_menu.Image4MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

// при нажатии на кнопку главная форма закроется и откроется форма с нормативами

begin

normativy.show;

glavnoe\_menu.hide

end;

procedure Tglavnoe\_menu.Image5MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

// при нажатии на кнопку главная форма закроется и откроется форма с интересными фактами

interest\_facts.show;

glavnoe\_menu.hide

end;

procedure Tglavnoe\_menu.Image7MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

// при нажатии на кнопку главная форма закроется и откроется форма с игрой

glavnoe\_menu.hide;

pazl.show;

end;

procedure Tglavnoe\_menu.N2Click(Sender: TObject);

begin

// при нажатии на кнопку главная форма закроется и откроется форма с тестом

glavnoe\_menu.hide;

test.show;

end;

procedure Tglavnoe\_menu.N4Click(Sender: TObject);

begin

// при нажатии на кнопку приложение закроется

close;

end;

procedure Tglavnoe\_menu.N5Click(Sender: TObject);

begin

// прописываем путь для открытия справочной системы

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

end;

end.

unit Unit4;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.OleCtrls, SHDocVw,

Vcl.Imaging.pngimage, Vcl.ExtCtrls, Vcl.Menus;

type

Tstili\_plavaniya = class(TForm)

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

WebBrowser1: TWebBrowser;

Image1: TImage;

Label1: TLabel;

Image6: TImage;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

procedure Image2MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Image3MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Image4MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Image5MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure N4Click(Sender: TObject);

procedure N2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

stili\_plavaniya: Tstili\_plavaniya;

implementation

{$R \*.dfm}

uses Unit3, Unit5, Unit6, Unit7, Unit9, Unit8;

procedure Tstili\_plavaniya.Image2MouseDown(Sender: TObject;

Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

begin

// при нажатии на кнопку отобразиться данный веб-ресурс

WebBrowser1.navigate(ExtractFilePath(ParamStr(0)) + 'Стили\Кроль.htm');

end;

procedure Tstili\_plavaniya.Image3MouseDown(Sender: TObject;

Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

begin

// при нажатии на кнопку отобразиться данный веб-ресурс

WebBrowser1.navigate(ExtractFilePath(ParamStr(0)) + 'Стили\Брасс.htm');

end;

procedure Tstili\_plavaniya.Image4MouseDown(Sender: TObject;

Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

begin

// при нажатии на кнопку отобразиться данный веб-ресурс

WebBrowser1.navigate(ExtractFilePath(ParamStr(0)) + 'Стили\Баттерфляй.htm');

end;

procedure Tstili\_plavaniya.Image5MouseDown(Sender: TObject;

Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

begin

// при нажатии на кнопку отобразиться данный веб-ресурс

WebBrowser1.navigate(ExtractFilePath(ParamStr(0)) + 'Стили\На\_спине.htm');

end;

procedure Tstili\_plavaniya.N2Click(Sender: TObject);

begin

// при нажатии на кнопку закроется данная форма и откроется главная форма

stili\_plavaniya.hide;

glavnoe\_menu.show;

end;

procedure Tstili\_plavaniya.N4Click(Sender: TObject);

begin

// при нажатии на кнопку программа завершится

close;

end;

end.

unit Unit5;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

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

Vcl.Imaging.jpeg, Vcl.StdCtrls, Vcl.Menus, Vcl.ComCtrls, Vcl.OleCtrls,

SHDocVw;

type

Tsportsmen = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

TabSheet3: TTabSheet;

WebBrowser1: TWebBrowser;

WebBrowser2: TWebBrowser;

WebBrowser3: TWebBrowser;

TabSheet4: TTabSheet;

WebBrowser4: TWebBrowser;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

procedure FormCreate(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N2Click(Sender: TObject);

var

private

{ Private declarations }

public

{ Public declarations }

end;

var

sportsmen: Tsportsmen;

implementation

{$R \*.dfm}

uses Unit8, Unit3, Unit4, Unit6, Unit7, Unit9;

procedure Tsportsmen.FormCreate(Sender: TObject);

begin

// при открытии формы будут отображаться следующие веб-ресурсы

WebBrowser1.navigate(ExtractFilePath(ParamStr(0)) +

'Спортсмены\Алина Змушко.htm');

WebBrowser2.navigate(ExtractFilePath(ParamStr(0)) +

'Спортсмены\Анастасия Шкурдай.htm');

WebBrowser3.navigate(ExtractFilePath(ParamStr(0)) +

'Спортсмены\Илья Шиманович.htm');

WebBrowser4.navigate(ExtractFilePath(ParamStr(0)) +

'Спортсмены\Евгений Цуркин.htm');

end;

procedure Tsportsmen.N2Click(Sender: TObject);

begin

// при нажатии на кнопку меню закроется данная форма и откроется главная форма

sportsmen.hide;

glavnoe\_menu.show;

end;

procedure Tsportsmen.N4Click(Sender: TObject);

begin

// при нажатии на кнопку меню программа завершится

close;

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.ComCtrls,

Vcl.OleCtrls, SHDocVw, Vcl.Menus;

type

Tnormativy = class(TForm)

Tab: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

TabSheet3: TTabSheet;

TabSheet4: TTabSheet;

TabSheet5: TTabSheet;

WebBrowser1: TWebBrowser;

WebBrowser2: TWebBrowser;

WebBrowser3: TWebBrowser;

WebBrowser4: TWebBrowser;

WebBrowser5: TWebBrowser;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

procedure FormCreate(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

normativy: Tnormativy;

implementation

{$R \*.dfm}

uses Unit8, Unit3, Unit4, Unit5, Unit7, Unit9;

procedure Tnormativy.FormCreate(Sender: TObject);

begin

// при открытии формы будут отображаться следующие веб-ресурсы

WebBrowser1.navigate(ExtractFilePath(ParamStr(0)) + 'НормативыРБ\м\_50.htm');

WebBrowser2.navigate(ExtractFilePath(ParamStr(0)) + 'НормативыРБ\ж\_50.htm');

WebBrowser3.navigate(ExtractFilePath(ParamStr(0)) + 'НормативыРБ\м\_25.htm');

WebBrowser4.navigate(ExtractFilePath(ParamStr(0)) + 'НормативыРБ\ж\_25.htm');

WebBrowser5.navigate(ExtractFilePath(ParamStr(0)) + 'НормативыРБ\юноши.htm');

end;

procedure Tnormativy.N2Click(Sender: TObject);

begin

// при нажатии на кнопку меню закроется данная форма и откроется главная форма

normativy.hide;

glavnoe\_menu.show;

end;

procedure Tnormativy.N4Click(Sender: TObject);

begin

// при нажатии на кнопку меню программа завершится

close;

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.Menus, Vcl.StdCtrls,

Vcl.Imaging.pngimage, Vcl.ExtCtrls;

type

Tinterest\_facts = class(TForm)

Image1: TImage;

Memo1: TMemo;

Image2: TImage;

Image3: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

procedure FormCreate(Sender: TObject);

procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure N4Click(Sender: TObject);

procedure N2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

interest\_facts: Tinterest\_facts;

n: Integer;

implementation

{$R \*.dfm}

uses Unit8, Unit3, Unit4, Unit5, Unit6, Unit9;

procedure Tinterest\_facts.FormCreate(Sender: TObject);

begin

// очищаем Memo1

Memo1.text := '';

end;

procedure Tinterest\_facts.Image1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

n := random(15); // присваиваем n рандомное значеник

Memo1.Lines.LoadFromFile(ExtractFilePath(ParamStr(0)) + 'Интересные факты\' +

IntToStr(n) + '.txt') // загружаем в Memo1 текстовый файл

end;

procedure Tinterest\_facts.N2Click(Sender: TObject);

begin

// при нажатии на кнопку меню закроется данная форма и откроется главная форма

interest\_facts.hide;

glavnoe\_menu.show;

end;

procedure Tinterest\_facts.N4Click(Sender: TObject);

begin

// при нажатии на кнопку меню программа завершится

close;

end;

end.

unit Unit8;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

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

Vcl.Imaging.jpeg, Vcl.Imaging.pngimage, Vcl.Menus;

type

Ttest = class(TForm)

RadioGroup1: TRadioGroup;

RadioGroup2: TRadioGroup;

Label1: TLabel;

Image2: TImage;

Image1: TImage;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

procedure FormCreate(Sender: TObject);

procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure RadioGroup1Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

test: Ttest;

f: text;

s: string;

Nvern, ball: Integer;

implementation

{$R \*.dfm}

uses Unit3, Unit4, Unit5, Unit6, Unit7, Unit9;

procedure Ttest.FormCreate(Sender: TObject);

begin

// очищаем Label1

Label1.visible := false;

end;

procedure Ttest.Image1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

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

begin

if RadioGroup2.ItemIndex = Nvern - 1 then

ball := ball + 1; // Если выбранный вариант соответствует

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

Repeat // и очищается поле для следующего вопроса

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

begin

delete(s, 1, 1);

RadioGroup2.Caption := s;

end

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

begin

delete(s, 1, 1);

Nvern := StrToInt(s);

end

else

RadioGroup2.Items.Add(s);

readln(f, s);

Label1.Caption := s;

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

end

// Если конец файла достигнут, значит вопросы закончились

Else if Eof(f) then

begin

delete(s, 1, 1);

Nvern := StrToInt(s);

if RadioGroup2.ItemIndex = Nvern - 1 then

ball := ball + 1;

Label1.visible := true;

Label1.Caption := 'Ого, вы набрали целых ' + IntToStr(ball) + '/10 баллов!';

// Вывод количества баллов

CloseFile(f);

Image1.Enabled := false;

// кнопка становится недоступной в завершении проекта

end;

end;

procedure Ttest.N2Click(Sender: TObject);

begin

// при нажатии на кнопку данная форма закроется и откроется главная форма

test.hide;

glavnoe\_menu.show;

end;

procedure Ttest.N4Click(Sender: TObject);

begin

// при нажатии на кнопку меню программа завершится

close;

end;

procedure Ttest.RadioGroup1Click(Sender: TObject);

begin

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

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

Image1.Enabled := true; // Кнопка Далее

case RadioGroup1.ItemIndex of

// В зависимости от выбранного варианта переменная f

0:

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

1:

AssignFile(f, 'variant2.txt');

end;

reset(f); // Открываем файл для чтения

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

ball := 0; // изначально количество баллов 0

repeat

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

begin // Если первый символ строки ‘-‘ значит это вопрос

delete(s, 1, 1);

RadioGroup2.Caption := s;

end

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

begin // Если перв символ ‘\*’ значит это номер верного ответа

delete(s, 1, 1);

Nvern := StrToInt(s);

end

else

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

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

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

// Считывание и отправление вариантов ответов в RadiGroup до тех пор

// пока не достигнут следующий вопрос или конец файла

end;

end.

unit Unit9;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

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

Vcl.Imaging.pngimage, Vcl.MPlayer;

type

Tpazl = class(TForm)

PaintBox1: TPaintBox;

Image1: TImage;

PaintBox2: TPaintBox;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

MediaPlayer1: TMediaPlayer;

OpenDialog1: TOpenDialog;

procedure FormCreate(Sender: TObject);

procedure Image2MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Image3MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure PaintBox2MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure N2Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure Image5MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

private

{ Private declarations }

public

{ Public declarations }

end;

var

pazl: Tpazl;

puzzle: TBitmap; // создаем компонент Puzzle класса TBitmap

rect1, rect2: Trect;

click1, i, j, dx, dy, k, n, X, Y: Integer;

pole: array [1 .. 4, 1 .. 4] of Integer;

implementation

{$R \*.dfm}

uses Unit1, Unit3;

procedure Tpazl.FormCreate(Sender: TObject);

begin

puzzle := TBitmap.Create; // создаем bitmap

puzzle.LoadFromFile(ExtractFilePath(ParamStr(0)) + 'Приспособление.bmp');

// загружаем bitmap из файла

{ устанавливаем размеры картинки и будущего пазла }

PaintBox1.Width := puzzle.Width;

PaintBox1.Height := puzzle.Height;

PaintBox2.Width := puzzle.Width;

PaintBox2.Height := puzzle.Height;

{ картинка будет размером 4\*4 }

dx := puzzle.Width div 4; // размер одного фрагмента обозначим dx и dy

dy := puzzle.Height div 4;

Image1.visible := false;

click1 := 1;

// номер "клика"(первый клик и далее нечетный клик выбирает первый фрагмент, второй – четный, выбирает второй фрагмент и меняет их местами

end;

procedure Tpazl.Image2MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

PaintBox1.Canvas.Draw(0, 0, puzzle); // выводит исходный рисунок в Paintbox1

end;

procedure Tpazl.Image3MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

{ заполняем двумерный массив нyлями. С помощью массива будем определять, помещен ли в соответствующую клетку фрагмент рисунка. Если в массиве 0, то это свободное поле, где еще не поставлен фрагмент. }

for i := 1 to 4 do

for j := 1 to 4 do

pole[i, j] := 0; // Весь массив заполнили нулями

for i := 1 to 4 do

for j := 1 to 4 do

begin // Ищем свободное поле

repeat

k := random(4) + 1;

n := random(4) + 1;

until pole[k, n] = 0; // повторять, пока в массиве не останется 0

pole[k, n] := 1; // заполняем его единицей

X := (j - 1) \* dx;

// координаты фрагмента, который мы скопируем из PaintBox1 в Paintbox2 на случайно выбранное место

Y := (i - 1) \* dy;

rect1 := bounds(X, Y, dx, dy);

// границы этого фрагмента сохраняем в переменной rect1

X := (n - 1) \* dx; // координаты случайно выбранного фрагмента в Paintbox2

Y := (k - 1) \* dy;

rect2 := bounds(X, Y, dx, dy);

// границы этого фрагмента сохраняем в переменной rect2

PaintBox2.Canvas.CopyRect(rect2, PaintBox1.Canvas, rect1);

// копируем фрагмент из Paintbox1 в Paintbox2

end;

with PaintBox2.Canvas do // чертим клетки на Paintbox2

end;

procedure Tpazl.Image5MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

var

tr: boolean;

begin

tr := true;

for i := 1 to 4 do

begin

Y := (i - 1) \* dy;

for j := 1 to 4 do

{ проверка }

begin

X := (j - 1) \* dx;

if PaintBox2.Canvas.pixels[X, Y] <> PaintBox1.Canvas.pixels[X, Y] then

tr := false;

end;

end;

if tr = true then

MediaPlayer1.filename := ExtractFilePath(ParamStr(0)) + 'Музыка\win.mp3'

// если собран верно - включается музыка win.mp3

else

MediaPlayer1.filename := ExtractFilePath(ParamStr(0)) + 'Музыка\fail.mp3';

// если собран неверно - включается музыка fail.mp3

MediaPlayer1.open;

MediaPlayer1.play;

end;

procedure Tpazl.N2Click(Sender: TObject);

begin

// при нажатии на кнопку данная форма закроется и откроется главная форма

pazl.hide;

glavnoe\_menu.show;

end;

procedure Tpazl.N4Click(Sender: TObject);

begin

// при нажатии на кнопку меню программа завершится

close;

end;

procedure Tpazl.PaintBox2MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

var

u, v: Integer;

rect3: Trect;

procedure XYtoUV(X, Y: Integer; var u, v: Integer);

// процедура находит номер строки и столбца, в которых находится фрагмент c координатами Х и Y

begin

u := (X div dx) \* dx;

v := (Y div dy) \* dy;

end;

begin

XYtoUV(X, Y, u, v);

rect2 := bounds(0, 0, dx, dy); // границы фрагмента, скопированного в Image1

if click1 mod 2 = 1 then // если это нечетный "клик"

begin

// копируем фрагмент в Image1

rect1 := bounds(u, v, dx, dy);

Image1.Canvas.CopyRect(rect2, PaintBox2.Canvas, rect1);

end

else // если это четный "клик"

begin

// меняем местами два фрагмента

rect3 := bounds(u, v, dx, dy);

PaintBox2.Canvas.CopyRect(rect1, PaintBox2.Canvas, rect3);

PaintBox2.Canvas.CopyRect(rect3, Image1.Canvas, rect2);

end;

click1 := click1 + 1;

end;

end.