**Введение**

**Введение**

На курсовое проектирование была поставлена задача, разработать программу на тему: Разработка программного продукта «Генеалогическое древо» Цель курсового проекта заключается в восстановлении памяти о родственниках семьи.

Создаваемая программа будет рассчитана на любого рода пользователей. Применить данную программу смогут все люди, заинтересовавшиеся в данной теме.

Далее приведем краткое описание разделов пояснительной записки.

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

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

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

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

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

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

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

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

Схема работы системы будет представлена в графической части.

**1 Анализ задачи**

**1.1 Постановка задачи**

**Наименование задачи:** Разработка программного продукта «Генеалогическое древо» **Цель разработки:** создание проекта - Разработка программного продукта «Генеалогическое древо»

**Назначение:** данное приложение разработается для пользования им в кругу семьи, а также для личного пользования.

**Периодичность использования:** зависит от нужд потребителя, может использоваться ежедневно.

**Источники и способы получения данных:**

Входной информации нет.

К выходной информации относится – формы проекта.

Постоянный информацией в проекте будут являться текстовые файлы, графические изображения и др.

Разрабатываемый программный продукт должен позволять выполнить следующие действия:

- просмотр блоков с родственникам на дереве.

- выход из преложения

- просмотр видеоролика о семье

**1.2 Инструменты разработки**

Для разработки данного проекта будет выбрана среда Delphi (Rad Studio 11.2) так как это одна из мощнейших сред для разработки приложений такого уровня. Delphi — это среда объектно-ориентированного программирования, относящаяся к классу RAD – (Rapid Application Development – «Средство быстрой разработки приложений»), основанная на языке Object Pascal. Используется для разработки и поддержки программного обеспечения.

При разработке данного программного продукта был использован компьютер со следующими характеристиками:

- Процессор: i5 10300h 2.50 GHz;

- ОЗУ: 8Gb;

- Память: HDD 512GB;

- ОС – Windows 10.

Также для разработки программы необходимы:

- Google-браузер – нужен для нахождения информации и картинок;

- Word 2016 – нужен для написания пояснительной записки;

- PowerPoint 2016 – нужен для создания отчётной презентации;

- Photoshop 2021 – создание изображений;

- Smart install maker – Инсталлятор для программы

**1.3 Требования к приложению**

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

При моделировании форм следует учесть такие моменты:

- интуитивно понятный интерфейс и управление;

- небольшое количество информации на игровом экране;

- небольшие окна формы;

- визуально понятный и приятный глазу интерфейс;

- наличие информации о функционале игры;

- небольшие кнопки.

Минимальные системные требования:

- 1.8 GHz процессор;

- 1 GB RAM;

- от 500 MB доступного места на жестком диске.

Рекомендуемые системные требования:

- двухъядерный AMD Athlon или аналогичный процессор Intel (или более производительный);

- 2 GB RAM;

- от 500 MB доступного места на жестком диске.

Требования к интерфейсу**:** при разработке приложения должны быть использованы мягкие голубые и фиолетовые оттенки. Основные разделы приложения должны быть доступны с первой страницы. Грамотный пользовательский интерфейс.

Следовательно, каждое окно должно иметь ясную визуальную иерархию своих элементов. Фрагменты текста должны располагаться на экране так, чтобы пользователю 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.

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 один вид данных.

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

Вторым видом данных является результат программы –отсортированные пользовательские данные. Его также относят к отдельномувиду, так как ни пользователь, ни разработчик его не вводят, а программа сама получает его в результате выполнения определенных действий.

Описание данных – теоретическая текстовая информация

Представление изображений графических работ, видеоизображения.

Система справочной информации будет представлена файлом справки, которая будет содержать информацию об функционале приложения, правилах ее использования.

Таким образом, организация данных является важной задачей при разработке данной и любой программы.

**2.2 Процессы**

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно должна заниматься разрабатываемая программа. Главной ее задачей будет являться: просмотру фотографий и информации для вспоминания того или иного родственника.

Основными функциями приложения являются:

- Открытие страницы “Дерева”.

- Возможность открыть и посмотреть страницу с биографией родственника

- Изучение справочной информации про функционал программы

- Открытие формы с видеороликом о семье

Для решения задачи будет использоваться процедуры.

**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, как работать с данной программой.

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 1):

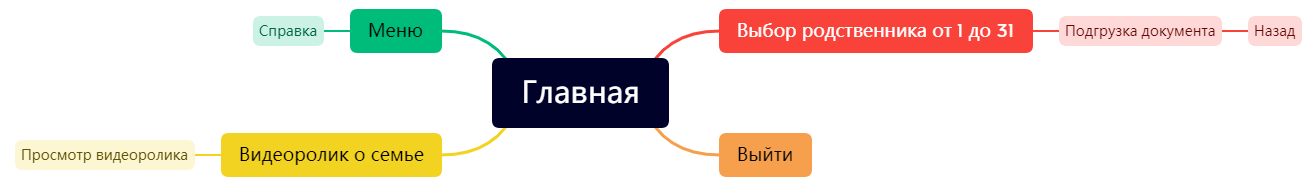


Рисунок – Навигация между окнами программы

**3. Реализация**

**3.1 Реализация проекта**

**3.1.1 Структура программы**

Данный курсовой проект содержит 5 модулей. Далее будет описано назначение каждого из них.

Модуль Unit 1 является модулем, за которым закреплена форма загрузки программы.

Модуль Unit 2 – является модулем, за которым закреплена главная форма, с котором можно перейти на другие формы.

Модуль Unit 3 - модуль, содержащий в себе ссылки на документы с биографией родственников.

Модуль Unit 4 - модуль, содержащий в себе форму c видеороликом о семье.

Модуль Unit 5 - модуль, содержащий в себе форму для загрузки документов при нажатии на кнопку.

**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 элементом управления закреплена | Назначение |
| 2 | 3 | 4 | 5 |
| procedure FormClose(Sender: TObject; var Action: TCloseAction); | Во всех | TForm | Закрытие программы |
| procedure FormShow(Sender: TObject); | Во всех | TForm | Подгрузка всех нужных файлов/картинок для формы |
| procedure TForm1.Image1ClickTimer(Sender: TObject); | Start | Timer1 | Отвечает за работу таймера, который в свою очередь загружает программу |
| procedure TForm1.Label1ClickTimer(Sender: TObject); | Start | Label1 | Отвечает за работу таймера, который обеспечивает моргание надписи |
| procedure TMain.N3Click(Sender: TObject); | Main,Author,Three | TMainMenu | Открытие формы с генеологическим деревом |
| procedure TMain.N4Click(Sender: TObject); | Main,Author,Three | TMainMenu | Открытие формы с видео-роликом |
| procedure TMain.N2Click(Sender: TObject); | Main,Author,Three | TMainMenu | Открытие справки |
| procedure TThree.ListBox31Click(Sender: TObject); | Three | ListBox1 | Переход на другую форму и загрузка html документа |
| procedure TAuthor.Image2Click(Sender: TObject); | Three,InfoForm,Author | Image1 | Вернуться назад |
| procedure TInfoForm.Image2MouseEnter(Sender: TObject); | Main,Three,InfoForm,Author | Image1 | Увеличение картинки при наведении на нее курсора |
| procedure TInfoForm.Image2MouseLeave(Sender: TObject); | Main,Three,InfoForm,Author | Image1 | Уменьшение картинки |
| procedure TStart.mainbackgroundClick(Sender: TObject); | Start | Image1 | Запуск таймера загрузочного экрана |
| procedure TAuthor.Panel1Click(Sender: TObject); | Author | Panel1 | Пауза/продолжение |

**3.1.3 Описание использованных компонентов**

Описание использованных при разработке приложения компонентов приводиться в таблице 2

Таблица 2 – Использованные компоненты

|  |  |  |
| --- | --- | --- |
| Компонент | На какой форме расположено | Назначение |
| TMaimMenu | Main,Three,Author | Используется для создания справки |
| TImage | На всех формах | Картинки на формах |
| TWebBrowser1 | InfoForm | Браузер для чтения |
| TLabel1 | Start | Вывод информации |
| Timer1 | Start | Движение картинки,моргание надписей, для Gauge1 |
| Gauge1 | Start | Для загрузочного экрана, для просмотра прогресса изучения материала |
| Panel1 | Author | Вывод видео-ролика |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

| Имя файла | Назначение |
| --- | --- |
| Family Pedegrii.exe | Исполняемый файл проекта, используется для запуска программы на выполнение. |
| Project1.dproj | Файл проекта, связывает все файлы из которых состоит приложение. |
| Unit1.pas | Файл программного модуля загрузочной формы |
| Unit2.pas | Файл программного модуля главной формы |
| Unit3.pas | Файл программного модуля формы с генеологическим деревом |
| Unit4.pas | Файл программного модуля формы с видео-роликом |
| Unit5.pas | Файл программного модуля формы с информацией о родственнике |
| Unit1.dfm | Форма загрузочного экрана |
| Unit2.dfm | Главная форма |
| Unit3.dfm | Форма с генеологическим деревом |
| Unit4.dfm | Форма с видео-роликом |
| Unit5.dfm | Форма с информацией о родственнике |
| Video.wmv | Видео-ролик |
| Delphihelp.chm | Справочная информация |
| Setup.exe | Установочный пакет программы |
| Img | Папка со всеми картинками, задействованными программой |
| Files | Папка со всеми файлами задействованными программой |

**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 представлены в таблице 4

Таблица 4 - Отчет о результатах тестирования.

|  |  |  |  |
| --- | --- | --- | --- |
| Форма с древом | Ожидаемый результат | Фактический результат | Результат тестирования |
| Проверка пункта меню “Дерево” | Переход на форму выбора теста | Переход на форму | Выполнено |
| Проверка пункта меню «Выход» | Выход из проекта | Выход из проекта | Выполнено |
| Проверка пункта меню “Видео-ролик” | Открытие главной формы | Открылась главной форме | Выполнено |
| Проверка кнопки меню «О программе» | Открытие справочной инф. | Открылась справочная инф. | Выполнено |
| Проверка кнопок на форме с Деревом | Открытие формы с документом | Открылась форма с документом | Выполнено |
| Проверка всех имеющихся кнопок «Назад» | Переход на предыдущую форму | Переход на предыдущую форму | Выполнено |
| Проверка нажатия на видео-ролик | Пауза\Пуск видео | Видео останавливается и запускается при 1 нажатии | Выполнено |

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

При использовании данного электронного средства обучения пользователь не столкнуться с ошибками системы.

**5 Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Разработанная программа является Генеологическое древо

Данный средство обучения предназначено для просмотра биографии родственников

Основными функциями приложения являются:

- Открытие страницы “Дерева”.

- Возможность открыть и посмотреть страницу с биографией родственника

- Изучение справочной информации про функционал программы

- Открытие формы с видеороликом о семье

- Просмотр видеоролика о семье

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

Тестирование проводилось на различных классах ЭВМ и работать с данной программой было комфортно. Но даже на системе следующего типа работать было комфортно:

Разработка происходила на ПК со следующими характеристиками:

- Процессор: i5-10300h 2.50 GHz;

- ОЗУ: 8Gb;

- Память: SSG 512GB;

- ОС – Windows 10.

**5.2 Инсталляция**

Для того что бы установить программу необходимо запустить файл Setup.exe. Появится окно установки обучающего приложения Семейная родословная

Рисунок – Установка обучающего приложения Семейная родословная

После нажатия на кнопку далее появится окно выбора каталога установки

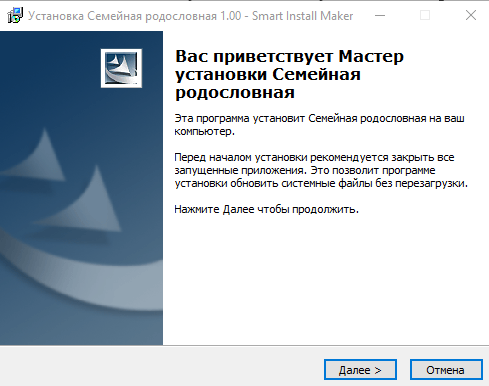
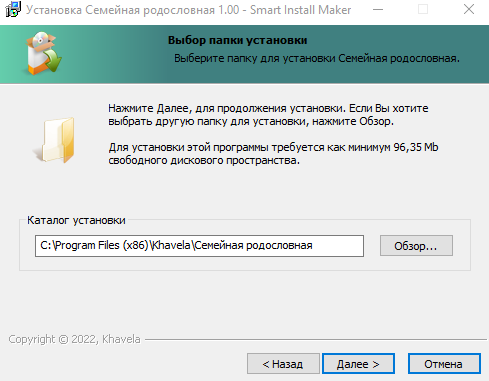


Рисунок – Выбор каталога установки

 После выбора каталога, нажимаем «Далее», после нам показывают каталог установки. Если же все устраивает, нажимаем «Установить»

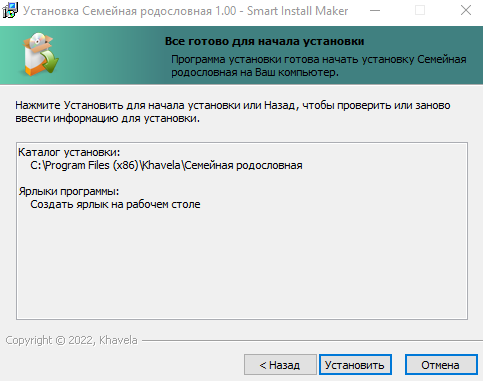


Рисунок – выбранный каталог установки

Затем происходит установки Программы, следует только ждать

После установки, можно найти каталог с скачанными файлами

**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 клавишей мыши на ярлык под названием Family pedigree



Вторым способом является запуск из каталога, в который установилась программа.

**5.3.2 Инструкции по работе с программой**

После запуска программы на экране появляется загрузочный экран (Рисунок 14).



Рисунок – Загрузочный экран

После нажатия в любое место программы происходит загрузка и нас переносит на форму с главным меню.

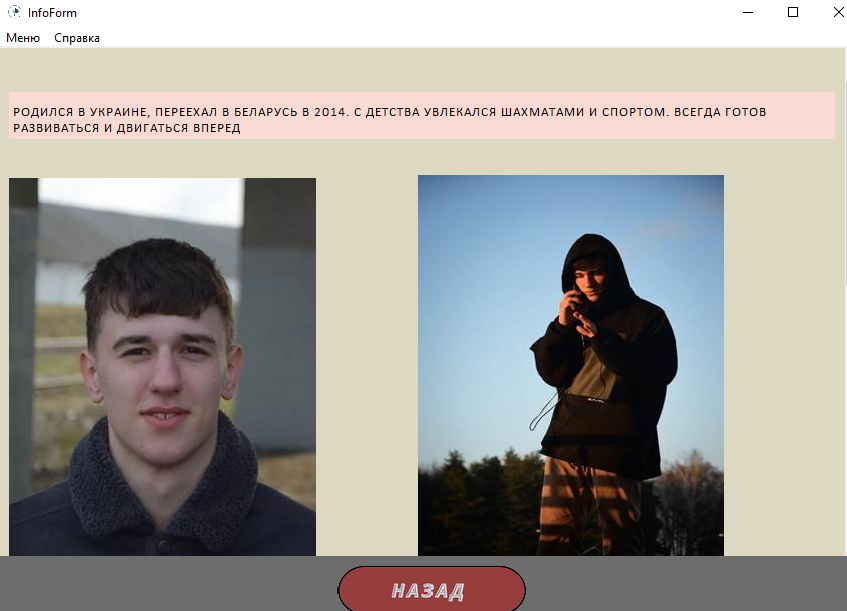


Рисунок – Окно с главным меню

Затем мыможем нажать на кнопку Дерево и нас переносит на форму со схемой родственников.



Рисунок – Окно со схемой родственников

Затем мы можем выбрать любого родственника и нажатием открыть форму с файлом (фото, текст). 

Окно с

Рисунок 17 – Окно с теоретическим материалом

Просмотрев кого либо из родственников мы можем вернуться назад нажатием на кнопку “Назад”

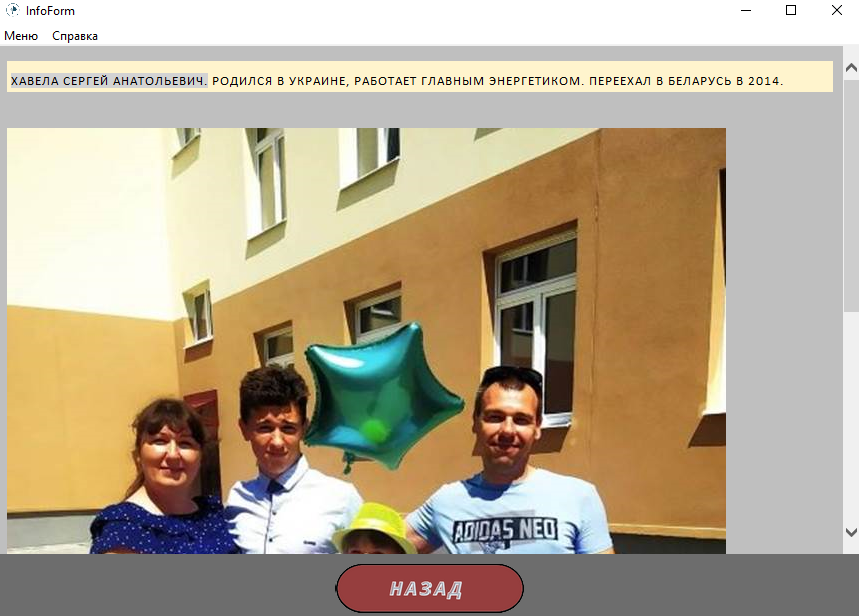
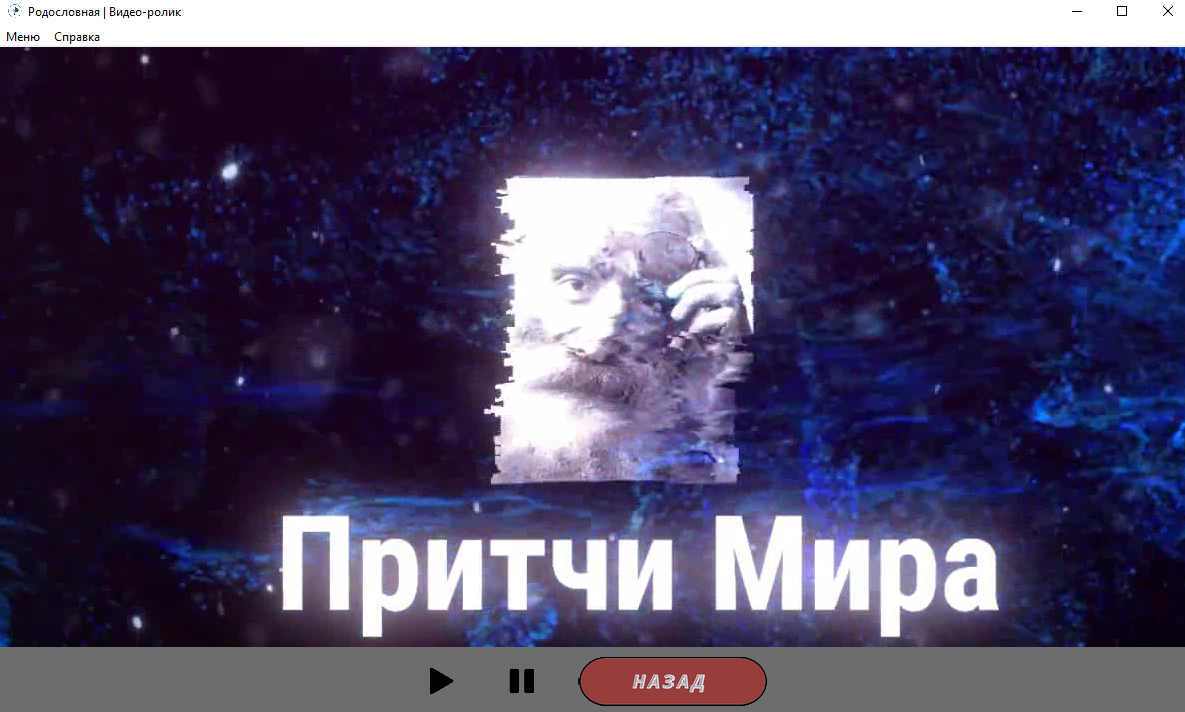


Рисунок 18 – Увеличение прогресса изучения уровня.

После просмотра родственников, можно вернуться на главное меню по кнопке Назад или через шапку приложения.

В Меню мы можем выбрать кнопку “Видео-материал” и нажатием перейти на форму с видеороликом. 



Во время просмотра видео, можно поставить его на паузу/запустить нажатием на экран, а также, с помощью кнопок на нижней панели. При необходимости, можно вернуться в меню нажав на кнопку Назад

При необходимости, можно выйти с программы в главном меню нажатием на кнопку Выход или нажатием на кнопку выход в шапке приложения.

**5.3.3 Завершение работы с программой**

Завершить работу с программой можно единственным способом: из окна Главное меню, нажав на кнопку «Выход» (Рисунок 23).



Рисунок – Выход из программы

**5.4 Использование системы справочной информации**

Справочную систему можно запустить с помощью команды меню Справка



Рисунок – справка

После нажатия появиться окно с справочной информацией о программе

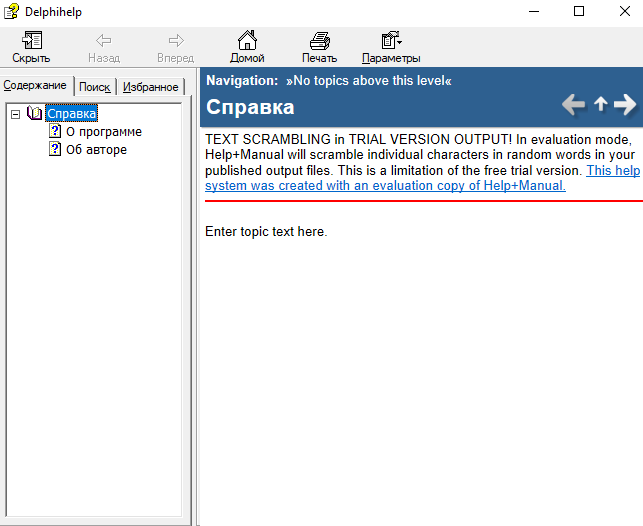


Рисунок – Справочная информация

**Заключение**

Цель данного проекта была реализована - создано Генеалогическое древо.

Данное электронное средство обучения будет интересно членам семьи, для которой было разработано это приложение.

Был реализован простой и понятный пользовательский интерфейс, который позволяет использовать приложение пользователю, не обладающему дополнительными знаниями ЭВМ.

В ходе реализации поставленной задачи был укреплён пройденный курс программирования в среде Delphi, а также получено много дополнительной информации о её возможностях.

После тщательного тестирования приложения были выявлены некоторые недоработки, которые были частично исправлены на стадии проектирования, и полностью исключены на стадии тестирования программы. В целом при реализации программы, были выполнены все условия, перечисленные в предыдущих разделах пояснительной записки.

Таким образом, можно сказать, что поставленная задача была реализована успешно.

**Список используемых источников**

1. <https://youtu.be/GB_seMNBzcQ> - видео о семье
2. Фотографии полученные от родственников.

3. <https://pixabay.com> - Иллюстрации и картинки

Приложение A

Листинг программы

unit Unit1; (Загрузочная форма)

interface

uses

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

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

Vcl.ComCtrls, Vcl.Samples.Gauges, Vcl.StdCtrls, Vcl.Imaging.pngimage,

Vcl.Menus;

type

TStart = class(TForm)

mainbackground: TImage;

Text: TLabel;

Timer1: TTimer;

Timer2: TTimer;

Gauge1: TGauge;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormShowTimer(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure mainbackgroundClick(Sender: TObject);

procedure secondtimer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Start: TStart;

implementation

{$R \*.dfm}

uses Unit2;

procedure TStart.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Application.Terminate;

end;

procedure TStart.FormCreate(Sender: TObject);

begin

mainbackground.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/mainbackground.png');

Start.Icon.LoadFromFile(ExtractFilePath(Application.ExeName)+'img\icon.ico');

end;

procedure TStart.FormShowTimer(Sender: TObject);

begin

if(Text.Visible=True) then Text.Visible:=False

else Text.Visible:=True;

end;

procedure TStart.mainbackgroundClick(Sender: TObject);

begin

Timer2.Enabled:=True;

Timer1.Enabled:=False;

Text.Visible:=False;

end;

procedure TStart.secondtimer(Sender: TObject);

begin

Gauge1.Progress:=Gauge1.Progress+1;

if(Gauge1.Progress=100) then

begin

Timer2.Enabled:=False;

Start.Hide;

Main.Show;

end;

end;

end.

unit Unit2; (Главная форма)

interface

uses

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

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

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

type

TMain = class(TForm)

menubackground: TImage;

AuthorButton: TImage;

ExitButton: TImage;

ThreeButton: TImage;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure ExitButtonClick(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure ThreeButtonMouseEnter(Sender: TObject);

procedure ThreeButtonMouseLeave(Sender: TObject);

procedure AuthorButtonMouseEnter(Sender: TObject);

procedure ExitButtonMouseEnter(Sender: TObject);

procedure ExitButtonMouseLeave(Sender: TObject);

procedure AuthorButtonMouseLeave(Sender: TObject);

procedure ThreeButtonClick(Sender: TObject);

procedure AuthorButtonClick(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Main: TMain;

implementation

{$R \*.dfm}

uses Unit3,Unit4;

procedure TMain.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Application.Terminate;

end;

procedure TMain.FormShow(Sender: TObject);

begin

menubackground.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/menubackground.png');

AuthorButton.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/video.png');

ThreeButton.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/three.png');

ExitButton.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/exit.png');

Main.Icon.LoadFromFile(ExtractFilePath(Application.ExeName)+'img\icon.ico');

end;

procedure TMain.N2Click(Sender: TObject);

begin

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

end;

procedure TMain.N3Click(Sender: TObject);

begin

Main.Hide;

Three.Show;

end;

procedure TMain.N4Click(Sender: TObject);

begin

Main.Hide;

Author.Show;

end;

procedure TMain.N5Click(Sender: TObject);

begin

Application.Terminate;

end;

procedure TMain.ThreeButtonClick(Sender: TObject);

begin

Main.Hide;

Three.Show;

end;

procedure TMain.ThreeButtonMouseEnter(Sender: TObject);

begin

if(ThreeButton.Stretch=True) then ThreeButton.Stretch:=False;

end;

procedure TMain.ThreeButtonMouseLeave(Sender: TObject);

begin

if(ThreeButton.Stretch=False) then ThreeButton.Stretch:=True;

end;

procedure TMain.AuthorButtonClick(Sender: TObject);

begin

Main.Hide;

Author.Show;

end;

procedure TMain.AuthorButtonMouseEnter(Sender: TObject);

begin

if(AuthorButton.Stretch=True) then AuthorButton.Stretch:=False;

end;

procedure TMain.AuthorButtonMouseLeave(Sender: TObject);

begin

if(AuthorButton.Stretch=False) then AuthorButton.Stretch:=True;

end;

procedure TMain.ExitButtonClick(Sender: TObject);

begin

Application.Terminate;

end;

procedure TMain.ExitButtonMouseEnter(Sender: TObject);

begin

if(ExitButton.Stretch=True) then ExitButton.Stretch:=False;

end;

procedure TMain.ExitButtonMouseLeave(Sender: TObject);

begin

if(ExitButton.Stretch=False) then ExitButton.Stretch:=True;

end;

end.

unit Unit3; (Генеологическое дерево)

interface

uses

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

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

Vcl.ExtCtrls, Vcl.Menus,ShellAPI;

type

TThree = class(TForm)

Image1: TImage;

ListBox1: TListBox;

ListBox2: TListBox;

ListBox4: TListBox;

rvxv: TListBox;

ListBox3: TListBox;

ListBox5: TListBox;

ListBox6: TListBox;

ListBox7: TListBox;

ListBox8: TListBox;

ListBox9: TListBox;

ListBox10: TListBox;

ListBox11: TListBox;

ListBox12: TListBox;

ListBox13: TListBox;

ListBox14: TListBox;

ListBox15: TListBox;

ListBox16: TListBox;

ListBox17: TListBox;

ListBox18: TListBox;

ListBox19: TListBox;

ListBox20: TListBox;

ListBox21: TListBox;

ListBox22: TListBox;

ListBox23: TListBox;

ListBox24: TListBox;

ListBox27: TListBox;

ListBox28: TListBox;

ListBox29: TListBox;

ListBox30: TListBox;

ListBox31: TListBox;

ListBox32: TListBox;

MainMenu1: TMainMenu;

N1: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N2: TMenuItem;

Image2: TImage;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure rvxvClick(Sender: TObject);

procedure ListBox31Click(Sender: TObject);

procedure ListBox18Click(Sender: TObject);

procedure ListBox2Click(Sender: TObject);

procedure ListBox4Click(Sender: TObject);

procedure ListBox13Click(Sender: TObject);

procedure ListBox19Click(Sender: TObject);

procedure ListBox15Click(Sender: TObject);

procedure ListBox30Click(Sender: TObject);

procedure ListBox16Click(Sender: TObject);

procedure ListBox24Click(Sender: TObject);

procedure ListBox21Click(Sender: TObject);

procedure ListBox10Click(Sender: TObject);

procedure ListBox20Click(Sender: TObject);

procedure ListBox8Click(Sender: TObject);

procedure ListBox27Click(Sender: TObject);

procedure ListBox23Click(Sender: TObject);

procedure ListBox14Click(Sender: TObject);

procedure ListBox1Click(Sender: TObject);

procedure ListBox11Click(Sender: TObject);

procedure ListBox6Click(Sender: TObject);

procedure ListBox17Click(Sender: TObject);

procedure ListBox29Click(Sender: TObject);

procedure ListBox22Click(Sender: TObject);

procedure ListBox12Click(Sender: TObject);

procedure ListBox9Click(Sender: TObject);

procedure ListBox7Click(Sender: TObject);

procedure ListBox28Click(Sender: TObject);

procedure ListBox32Click(Sender: TObject);

procedure ListBox5Click(Sender: TObject);

procedure ListBox3Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure Image2Click(Sender: TObject);

procedure Image2MouseEnter(Sender: TObject);

procedure Image2MouseLeave(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure FormShow(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Three: TThree;

implementation

{$R \*.dfm}

uses Unit2,Unit4,Unit5;

procedure TThree.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Application.Terminate;

end;

procedure TThree.FormShow(Sender: TObject);

begin

Three.Icon.LoadFromFile(ExtractFilePath(Application.ExeName)+'img\icon.ico');

Image1.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/1.png');

Image2.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/backbutton.png');

end;

procedure TThree.Image2Click(Sender: TObject);

begin

Three.Hide;

Main.Show;

end;

procedure TThree.Image2MouseEnter(Sender: TObject);

begin

if(Image2.Stretch=True) then Image2.Stretch:=False;

end;

procedure TThree.Image2MouseLeave(Sender: TObject);

begin

if(Image2.Stretch=False) then Image2.Stretch:=True;

end;

procedure TThree.ListBox10Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\6.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox11Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\13.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox12Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\12.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox13Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\15.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox14Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\5.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox15Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\9.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox16Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\4.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox17Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\20.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox18Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\31.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox19Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\30.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox1Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\7.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox20Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\14.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox21Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\18.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox22Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\11.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox23Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\10.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox24Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\8.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox27Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\19.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox28Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\22.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox29Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\21.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox2Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\3.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox30Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\17.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox31Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\16.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox32Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\23.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox3Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\25.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox4Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\2.html');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox5Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\24.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox6Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\28.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox7Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\27.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox8Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\29.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.ListBox9Click(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\26.htm');

InfoForm.Show;

Three.Hide;

end;

procedure TThree.N2Click(Sender: TObject);

begin

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

end;

procedure TThree.N3Click(Sender: TObject);

begin

Three.Hide;

Main.Show;

end;

procedure TThree.N4Click(Sender: TObject);

begin

Author.Show;

Three.Hide;

end;

procedure TThree.N5Click(Sender: TObject);

begin

Application.Terminate;

end;

procedure TThree.rvxvClick(Sender: TObject);

begin

InfoForm.WebBrowser1.Navigate(ExtractFilePath(Application.ExeName)+'files\(1).html');

InfoForm.Show;

Three.Hide;

end;

end.

unit Unit4; (Форма с видео-роликом)

interface

uses

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

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

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

type

TAuthor = class(TForm)

MainMenu1: TMainMenu;

N1: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N2: TMenuItem;

MediaPlayer1: TMediaPlayer;

Image1: TImage;

Image2: TImage;

Panel1: TPanel;

Image3: TImage;

Image4: TImage;

procedure N4Click(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormShow(Sender: TObject);

procedure Panel1Click(Sender: TObject);

procedure Image2Click(Sender: TObject);

procedure Image2MouseEnter(Sender: TObject);

procedure Image2MouseLeave(Sender: TObject);

procedure Image3Click(Sender: TObject);

procedure Image4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Author: TAuthor;

i:integer;

implementation

{$R \*.dfm}

uses Unit3,Unit2;

procedure TAuthor.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Application.Terminate;

end;

procedure TAuthor.FormShow(Sender: TObject);

begin

Author.Icon.LoadFromFile(ExtractFilePath(Application.ExeName)+'img\icon.ico');

Image2.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/backbutton.png');

MediaPlayer1.FileName:=ExtractFilePath(Application.ExeName)+'video.wmv';

Image3.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/play.png');

Image4.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/stop.png');

MediaPlayer1.Open;

MediaPlayer1.Play;

i:=0;

end;

procedure TAuthor.Image2Click(Sender: TObject);

begin

MediaPlayer1.Close;

Author.Hide;

Main.Show;

end;

procedure TAuthor.Image2MouseEnter(Sender: TObject);

begin

if(Image2.Stretch=True) then Image2.Stretch:=False;

end;

procedure TAuthor.Image2MouseLeave(Sender: TObject);

begin

if(Image2.Stretch=False) then Image2.Stretch:=True;

end;

procedure TAuthor.Image3Click(Sender: TObject);

begin

MediaPlayer1.Play;

end;

procedure TAuthor.Image4Click(Sender: TObject);

begin

MediaPlayer1.Stop;

end;

procedure TAuthor.N2Click(Sender: TObject);

begin

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

end;

procedure TAuthor.N3Click(Sender: TObject);

begin

Author.Hide;

Main.Show;

end;

procedure TAuthor.N4Click(Sender: TObject);

begin

Three.Show;

Author.Hide;

end;

procedure TAuthor.N5Click(Sender: TObject);

begin

Application.Terminate;

end;

procedure TAuthor.Panel1Click(Sender: TObject);

begin

if(i=0) then

begin

MediaPlayer1.Stop;

i:=1;

end

else

begin

MediaPlayer1.Play;

i:=0;

end;

end;

end.

unit Unit5; (Форма с биографией родственника)

interface

uses

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

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

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

type

TInfoForm = class(TForm)

WebBrowser1: TWebBrowser;

Image1: TImage;

MainMenu1: TMainMenu;

N1: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N2: TMenuItem;

Image2: TImage;

procedure N3Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure Image2Click(Sender: TObject);

procedure Image2MouseEnter(Sender: TObject);

procedure Image2MouseLeave(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure FormShow(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

InfoForm: TInfoForm;

implementation

{$R \*.dfm}

uses Unit2,Unit3,Unit4;

procedure TInfoForm.FormShow(Sender: TObject);

begin

InfoForm.Icon.LoadFromFile(ExtractFilePath(Application.ExeName)+'img\icon.ico');

Image2.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'img/backbutton.png');

end;

procedure TInfoForm.Image2Click(Sender: TObject);

begin

InfoForm.Hide;

Three.Show;

end;

procedure TInfoForm.Image2MouseEnter(Sender: TObject);

begin

if(Image2.Stretch=True) then Image2.Stretch:=False;

end;

procedure TInfoForm.Image2MouseLeave(Sender: TObject);

begin

if(Image2.Stretch=False) then Image2.Stretch:=True;

end;

procedure TInfoForm.N2Click(Sender: TObject);

begin

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

end;

procedure TInfoForm.N3Click(Sender: TObject);

begin

InfoForm.Hide;

Three.Show;

end;

procedure TInfoForm.N4Click(Sender: TObject);

begin

InfoForm.Hide;

Author.Show;

end;

procedure TInfoForm.N5Click(Sender: TObject);

begin

Application.Terminate

end;

end;