**Введение**

На курсовое проектирование была поставлена задача создать познавательное приложение "Открытия ХХI века".

Цель курсового проекта заключается в создании этого познавательного приложения и документации к нему. Главной задачей при автоматизации этого проекта является облегчение понимания данной темы.

Созданная программа рассчитана на любую аудиторию, которая интересуется данной тематикой.

Приведём краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нём можно ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи. Также в этом разделе можно узнать о том, как данная задача решается в настоящее время. Описаны все входные и выходные данные. В подразделе «Инструменты обработки» рассмотрена среда, в которой был создан курсовой проект. Здесь также установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачи.

В разделе «Проектирование задачи» рассмотрены основные аспекты разработки программного продукта. Здесь можно узнать об организации данных в контексте среды разработки. В данном разделе описан пользовательский интерфейс, составление алгоритмов процесса обработки информации, описана разработка системы справочной информации.

«Реализация задачи» − это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые использованы при реализации данного приложения. В этом разделе описаны функции пользователя и их структура. Также здесь можно увидеть таблицу, в которой представлена полная аннотация файлов, используемых в данном проекте.

Четвёртый раздел – «Тестирование». В нём описано полное и функциональное тестирование данной программы, т.е. протестирован каждый пункт меню, каждая операция, которая выполняется приложением. Смоделированы всевозможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе «Применение» описано назначение, область применения, среды функционирования приложения. Также в нём описано использование справочной системы.

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

В разделе «Список используемых источников» приведён список используемых при разработке источников.

В приложении к пояснительной записке приведён листинг программы с необходимыми комментариями.

Схема работы системы представлена в графической части.

**1 Анализ задачи**

**1.1 Постановка задачи**

Темой курсового проекта является познавательного приложения "Открытия ХХI века".

Целью разработки является расширение знаний пользователя об открытиях XXI века.

Данный программный продукт будет разрабатывается для людей любого возраста, желающим получить информацию об открытиях XXI века. Пользователи смогут в любое время воспользоваться данным программным продуктом. Информация, приведённая в программном продукте будет взята из Википедии и специализированных сайтов. Это будет сделано с целью упростить доступ к информации, так как разрабатываемый программный продукт, в отличии от сайтов с которых будет браться информация, не будет зависеть от наличия интернета.

Кроме Web-сайта Wikipedia аналогичных программных продуктов найдено не было, однако рассматривая данный сайт можно выделить его удобство использования, данный сайт имеет крайне удобный интерфейс. Проблема рассматриваемого сайта в зависимости от интернета, в разрабатываемом программном обеспечении данная проблема будет отсутствовать, а также будет присутствовать удобный интерфейс.

В разрабатываемом программном продукте пользователь будет иметь возможность:

1. выбирать нужную ему статью для изучения;
2. перемещаться по формам;

Приведем описание процессов с входной, выходной и условно-постоянной информацией:

Выбор статьи. В данном процессе в качестве входной информации будет использоваться выбор статьи пользователем, в качестве выходной – статья выбранная пользователем, а в качестве условно-постоянной – количество статей.

Переход между формами. В данном процессе в качестве входной информации будет использоваться действия пользователя на форме, в качестве выходной – содержимое формы на которую перешёл пользователь, а в качестве условно-постоянной – количество форм и элементы на форме.

**1.2 Инструменты разработки**

Проект будет разрабатываться в среде разработки Delphi 7.Среда разработки Delphi 7 была выбрана в связи с требованием к реализации курсового проекта и в связи с субъективным удобством.

Delphi 7 [императивный,](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [структурированный](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [высокоуровневый](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) со [строгой](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B8_%D1%81%D0%BB%D0%B0%D0%B1%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) [статической типизацией](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) переменных. Основная область использования — написание прикладного программного обеспечения.

Параметры компьютера, на котором проводится разработка проекта:

* ОЗУ 4 ГБ;
* процессор Intel(R) Pentium(R) CPU N3710;
* ОС Windows 10 Pro .

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

* процессор не ниже Pentium 500 Mhz;
* объем ОЗУ не меньше 128 Mb;
* объем места на HDD не менее 200 Mb;
* ОС – Windows 7,8,10.

Таким образом требования к приложению оптимальные и само приложение непривередливо.

**1.3 Требования к приложению**

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

При моделировании форм следует учесть такие моменты:

* интуитивно понятный интерфейс;
* среднее количество информации на каждой форме.

Так же следует обратить внимание на такие моменты:

* одинаковый, средний размер окна формы;
* небольшие кнопки;
* приятный глазу интерфейс.

Минимальные системные требования:

* 1.6 GHz процессор (или более быстрый);
* 1 GB RAM;
* от 15 МB доступного места на жестком диске.

Рекомендуемые:

* 2GHz процессор (или более быстрый);
* 2 GB RAM;
* от 15 МB доступного места на жестком диске;
* двухъядерный процессор Intel или AMD (или более быстрый).

Требования к интерфейсу: при разработке приложения должен быть использован преимущественно серый 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 легко и понятно принимать информацию.

Пользователь не должен испытывать какого-либо дискомфорта в плане восприятия информации, изображенной на экране. Объекты (рисунки и символы) не должны быть слишком мелкие.

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

**2 Проектирование задачи**

**2.1 Организация данных**

Основными средствами хранения информации, будут являться текстовые файлы.

Система справочной информации будет представлено файлом справки, который будет содержать информацию о приложении и правила его использования.

Основными функциями приложения будут являться:

* Вывод информации по теме;
* Просмотр справочной информации.

Воспользовавшись пунктом «О программе», можно:

* Узнать информацию о разработчике
* Узнать версию программы
* Узнать название 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 задачей разрабатываемой программы будет являться помощь в изучении информации об открытиях XXI века.

Для реализации задач будут использоваться процедуры. С помощью процедуры, например, будет осуществляться считывание информации из файла, их появление на форме.

**2.3 Описание внешнего пользовательского интерфейса**

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

Особое внимание следует уделить интерфейсу. Разработчик должен так организовать внешний вид своей программы, что бы пользователь понял, что от него требуется. При решении задачи будут использованы компоненты для вывода информации, для вставки изображений, надписей и текста, и др.

Для организации эффектной 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 самостоятельно изучить информацию об открытиях XXI века.

Таким образом, для успешной работы всего проекта в целом следует выбрать оптимальный вариант теории, для хорошей возможности изучить тему, подготовить сохраняемые файлы и заставку, а также учесть всевозможные вариант ошибок при работе над проектом.

**3 Реализация**

**3.1 Структура программы**

Данный курсовой проект содержит 4 модуля. Далее рассмотрим назначение каждого модуля:

Unit1 – это модуль-заставка.

Unit2 – это главный модуль, с которого можно попасть в другие модули и который содержит меню.

Unit3 – это модуль с выбором статей.

Unit4 – это модуль, где выводится информация статьи которую выбрал пользователь.

**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 выполнения закреплена? | Назначение |
| 1 | 2 | 3 | 4 |
| 1 procedure Timer1Timer(Sender: TObject) | Unit1.pas | TTimer1 | Таймер, который определяет время визуальности формы |
| 2 procedure Button1Click(Sender: TObject); | Unit2.pas | TButton1 | Открывает Unit3.pas |
| 3 procedure Button2Click(Sender: TObject); | Unit2.pas | TButton2 | Кнопка, которая изменяет закрывает форму |
| 4 procedure N1Click(Sender: TObject); | Unit2.pas | TMenuItem | Открывает справочник «О программе» |
| 5 procedure Button1Click(Sender: TObject); | Unit3.pas | TButton1 | Открывает Unit4.pas |
| 6 procedure Button2Click(Sender: TObject); | Unit3.pas | TButton2 |
| 7 procedure Button3Click(Sender: TObject); | Unit3.pas | TButton3 |
| 8 procedure Button4Click(Sender: TObject); | Unit3.pas | TButton4 |

Продолжение Таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 9 procedure Button5Click(Sender: TObject); | Unit3.pas | TButton5 | Открывает Unit4.pas |
| 10 procedure Button6Click(Sender: TObject); | Unit3.pas | TButton6 | Кнопка, возвращает на Unit3.pas |
| 11 procedure Memo1Change(Sender: TObject); | Unit4.pas | TMemo | Поле вывода теории |
| 12 procedure Button1Click(Sender: TObject); | Unit4.pas | TButton1 | Возвращает на Unit3.pas |
| 13 procedure Button2Click(Sender: TObject); | Unit4.pas | TButton2 | Возвращает на Unit2.pas |

**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 |
| 1 TTimer1 | Form1 | Таймер |
| 2 Image1 | Form1 | Картинка заставки |
| 3 Button1 | Form2 | Кнопка «Статьи» |
| 4 Button2 | Form2 | Кнопка «Выход» |
| 5 N1 | Form2 | Кнопка «О программе» |
| 6 Lable1 | Form2 | Заголовок |
| 7 Image1 | Form2 | Фон формы |
| 8 Button1 | Form3 | Кнопки «Прочитать статью» |
| 9 Button2 | Form3 |
| 10 Button3 | Form3 |
| 11 Button4 | Form3 |
| 12 Button5 | Form3 |
| 13 Button6 | Form3 | Кнопка «Назад» |
| 14 Image1 | Form3 | Картинки для статей |
| 15 Image2 | Form3 |
| 16 Image3 | Form3 |
| 17Image4 | Form3 |
| 18 Image5 | Form3 |
| 19 Image6 | Form3 | Фон формы |
| 20 Button1 | Form4 | Кнопка «Назад» |
| 21 Button2 | Form4 | Кнопка «Главное меню» |
| 22 Memo1 | Form4 | Поле вывода теории |
| 23 Image1 | Form4 | Фон формы |

Продолжение Таблицы 2

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в Таблице 3.

Таблица 3 – Спецификация программы

|  |  |
| --- | --- |
| Имя файла | Назначение |
| 1 | 2 |
| 1 Project1.exe | Исполняемый файл проекта, используется для запуска программы на выполнение |
| 2 Printer.txt, Protez.txt, Serdce.txt, SocSeti.txt, Virtyalreal.txt | Текстовые файлы, используются для вывода информации статьи которую выбрал пользователь |
| 3 Справочник.chm | Файл справки |

**4 Тестирование**

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

Отчёт о результатах тестирования предоставлен в Таблице 4.

Таблица 4 – Отчёт о результатах тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый результат | Физический результат | Результат тестирования |
| 1 | 2 | 3 | 4 |
| Проверка открытия приложения | Окно приложения появилось | Окно приложения появилось | Выполнено |
| Вывод информации | Вывод информации в новом окне | Вывод информации в новом окне | Выполнено |
| Проверка работы кнопок | Кнопки делают свою работу | Кнопки делают свою работу | Выполнено |
| Проверка закрытия приложения | Окно приложения исчезло | Окно приложения исчезло | Выполнено |

**5 Применение**

**5.1 Инсталляция**

Для того, чтобы установить программу, необходимо запустить setup.exe. Появится окно для установки программы (Рисунок 1).

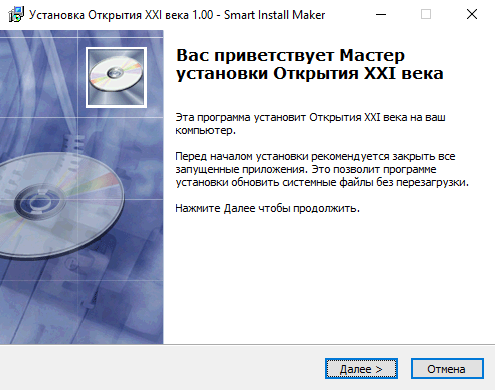


Рисунок 1 – Установка программы

Далее появится окно о запросе места установки (Рисунок 2).

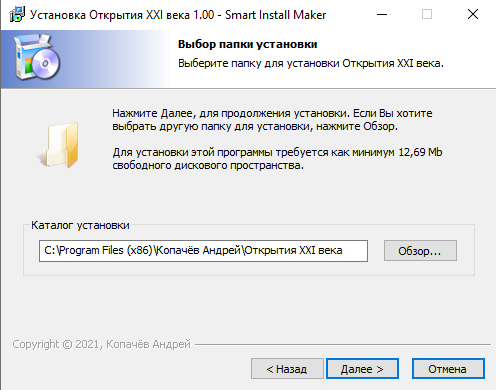


Рисунок 2 – Выбор места для установки

Далее выводится окно с выбором дополнительных ярлыков (Рисунок 3).

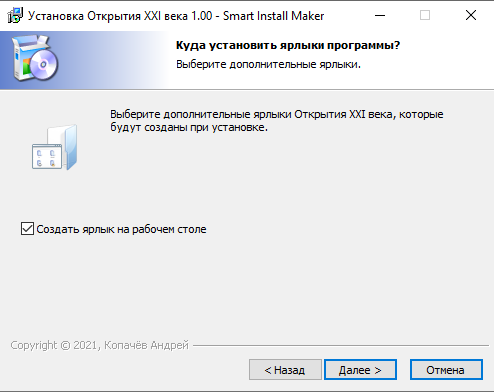


Рисунок 3 – Окно с выбором дополнительных ярлыков

Далее идет окно о том что все готово для начала установки установки (Рисунок 4).

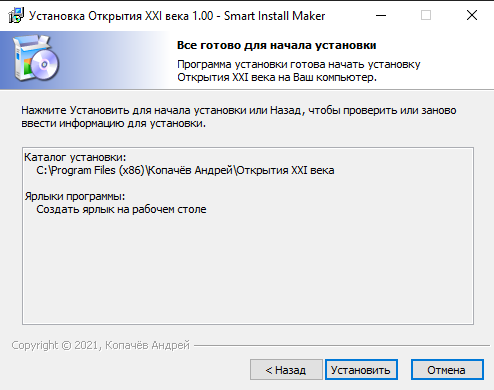


Рисунок 4 – окно о том что все готово для начала установки

Следующим шагом идёт установка данного приложения и выведение сообщения об успешности её выполнения.

**5.2 Выполнение программы**

**5.2.1 Запуск программы**

Программу можно запустить несколькими способами.

Одним из них можно запустить программу, щёлкнув два раза левой кнопкой мыши по ярлыку на рабочем столе с названием «Открытия XXI века» (Рисунок 5).



Рисунок 5 – Ярлык для захода в программу

Вторым способом является Пуск. Нужно нажать меню Пуск и нажать на приложение «Открытия XXI века» (Рисунок 6).



Рисунок 6 – Приложение для захода в программу

**5.2.3 Инструкции по работе с программой**

После запуска на экране появляется заставка (Рисунок 7).

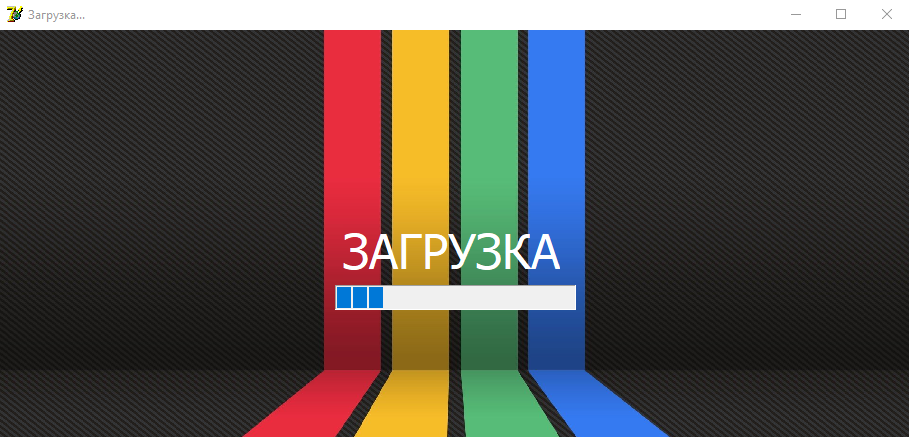
****

Рисунок 7 – Заставка

Затем появляется и главная форма (Рисунок 8), благодаря которой пользователь может увидеть кнопки с помощью которых можно перейти на форму с выбором статей.

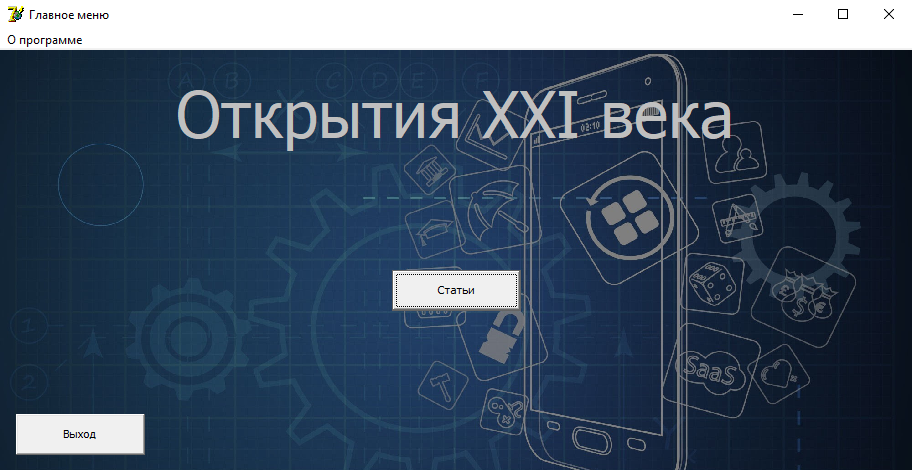
****

Рисунок 8 – Главная форма

После того, как пользователь нажал на кнопку «Статьи», то увидит окно с выбором статей(Рисунок 9).

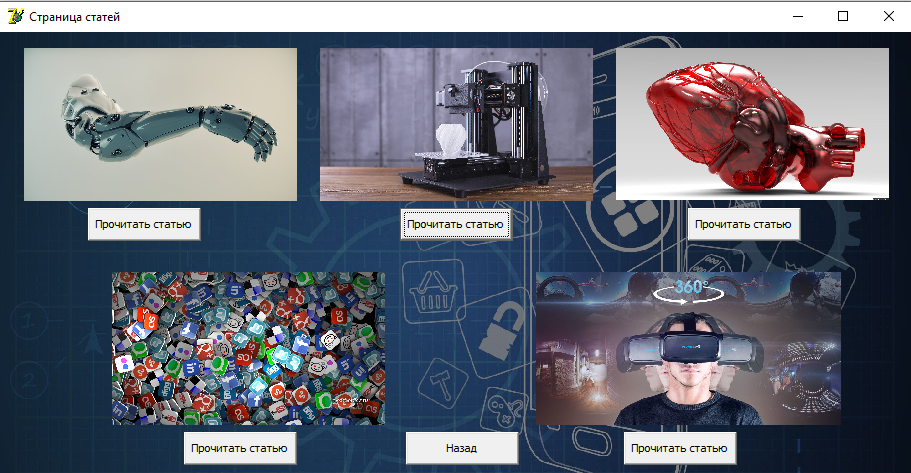


Рисунок 9 – Выбор статей

После нажатия на кнопку «Прочитать стаью» пользователь увидит окно с теорией(Рисунок 10).

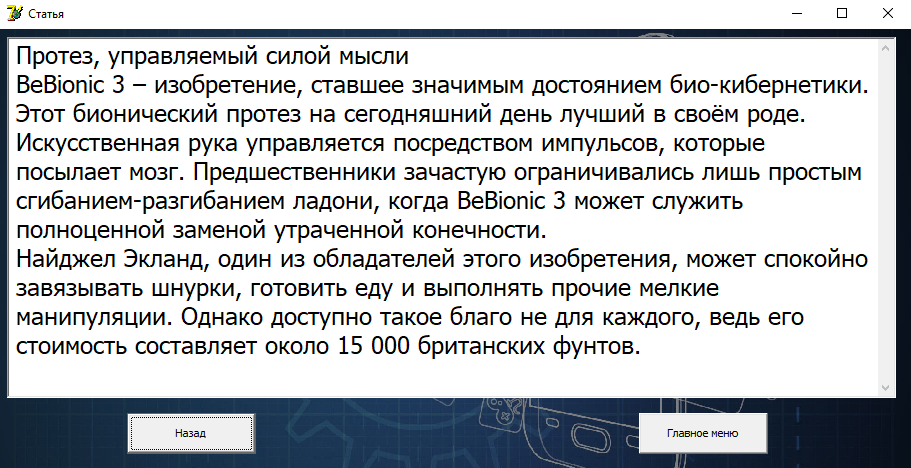
****

Рисунок 10 – Окно с теорией

На главной форме пользователь так же может вызвать справку и подробней узнать о программе, разработчике(Рисунок 11, Рисунок 12)



Рисунок 11 – Кнопка вызова справки

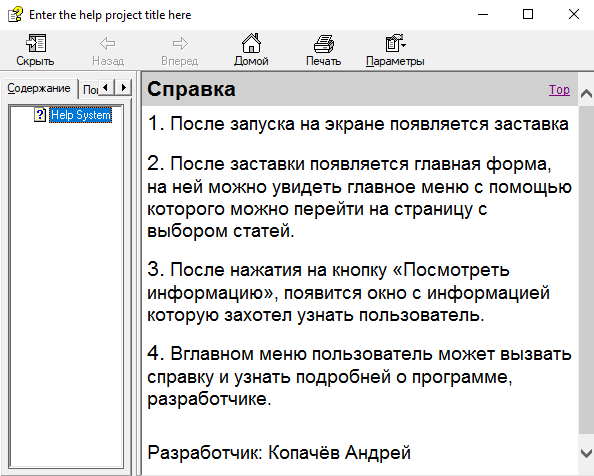


Рисунок 12 – Окно справки

**Заключение**

Целью данного курсового проекта являлась разработка познавательного приложения "Открытия ХХI века".

В ходе реализации поставленной задачи был укреплён пройденный курс программирования в среде Delphi 7, а так же получено много дополнительной информации о её возможностях.

Следует также учесть, что в поставленной задачи был реализован простой интерфейс, который позволяет использовать приложение пользователю, не обладающему дополнительными знаниями ЭВМ.

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 программа была реализована вполне успешно.

**Список использованных источников**

**1** <https://www.syl.ru/article/392955/samyie-vajnyie-izobreteniya-veka---spisok-osobennosti-i-interesnyie-faktyi>

**2** https://www.vokrugsveta.ru/article/239630/

**3** <https://www.cyberforum.ru/delphi-multimedia/thread92899.html>

**Приложение А**

**Листинг программы**

unit Unit1;

interface

uses

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

Dialogs, StdCtrls, ComCtrls, ExtCtrls, jpeg;

type

TForm1 = class(TForm)

Timer1: TTimer;

ProgressBar1: TProgressBar;

Image1: TImage;

Label1: TLabel;

procedure Timer1Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses Unit2;

{$R \*.dfm}

procedure TForm1.Timer1Timer(Sender: TObject);

begin

ProgressBar1.Position:=ProgressBar1.Position+20;

if (ProgressBar1.Position=ProgressBar1.Max) then

begin

Form1.Hide;

Timer1.Enabled := false;

Form2.Show;

end;

end;

end.

unit Unit2;

interface

uses

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

Dialogs, StdCtrls, jpeg, ExtCtrls, Menus,shellapi;

type

TForm2 = class(TForm)

Button1: TButton;

Button2: TButton;

Image1: TImage;

Label1: TLabel;

MainMenu1: TMainMenu;

N1: TMenuItem;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure N1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit3, Unit1;

{$R \*.dfm}

procedure TForm2.Button1Click(Sender: TObject);

begin

Form3.show;

form2.Hide;

end;

procedure TForm2.Button2Click(Sender: TObject);

begin

close;

end;

procedure TForm2.N1Click(Sender: TObject);

begin

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

end;

end.

unit Unit3;

interface

uses

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

Dialogs, StdCtrls, jpeg, ExtCtrls;

type

TForm3 = class(TForm)

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Image6: TImage;

procedure Button6Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

uses Unit2, Unit4;

{$R \*.dfm}

procedure TForm3.Button6Click(Sender: TObject);

begin

Form2.show;

form3.Hide;

end;

procedure TForm3.Button1Click(Sender: TObject);

begin

Form4.show;

Form4.Memo1.Lines.LoadFromFile('Protez.txt');

Form3.Hide;

end;

procedure TForm3.Button2Click(Sender: TObject);

begin

Form4.show;

Form4.Memo1.Lines.LoadFromFile('Printer.txt');

Form3.Hide;

end;

procedure TForm3.Button3Click(Sender: TObject);

begin

Form4.show;

Form4.Memo1.Lines.LoadFromFile('Serdce.txt');

Form3.Hide;

end;

procedure TForm3.Button4Click(Sender: TObject);

begin

Form4.show;

Form4.Memo1.Lines.LoadFromFile('SocSeti.txt');

Form3.Hide;

end;

procedure TForm3.Button5Click(Sender: TObject);

begin

Form4.show;

Form4.Memo1.Lines.LoadFromFile('Virtyalreal.txt');

Form3.Hide;

end;

end.

unit Unit4;

interface

uses

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

Dialogs, StdCtrls, jpeg, ExtCtrls;

type

TForm4 = class(TForm)

Memo1: TMemo;

Button1: TButton;

Button2: TButton;

Image1: TImage;

procedure Memo1Change(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

implementation

uses Unit3, Unit2;

{$R \*.dfm}

procedure TForm4.Memo1Change(Sender: TObject);

begin

Memo1.ScrollBars:=ssVertical;

end;

procedure TForm4.Button1Click(Sender: TObject);

begin

Form3.show;

Form4.Hide;

end;

procedure TForm4.Button2Click(Sender: TObject);

begin

Form2.show;

Form4.Hide;

end;

end.