**Введение**

На курсовое проектирование была поставлена задача создать информационно-справочное приложение «Паттерны в Python».

Цель курсового проекта заключается в создании этого информационно-справочного 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 информационно-справочное приложение «Паттерны в Python».

Целью разработки является расширение знаний пользователя о разработке в Python.

Данный программный продукт будет разрабатывается для людей любого возраста, желающим получить информацию о паттернах программирования на языке программирования Python. Пользователи смогут в любое время воспользоваться данным программным продуктом. Информация, приведённая в программном продукте будет взята из Вики 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, Wiki и refactoring.guru аналогичных программных продуктов найдено не было, однако рассматривая данные сайты можно выделить их удобство использования, данные сайты имеют крайне удобные интерфейсы. Проблема рассматриваемых сайтов 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 виды паттернов;
2. перемещаться по формам;
3. сохранить выбранную информацию.

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

Выбор вида паттернов. В данном процессе в качестве входной информации будет использоваться выбор вида паттернов пользователем, в качестве выходной – паттерны соответствующие выбранному пользователем виду, а в качестве условно-постоянной – количество паттернов и их видов.

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.

Сохранение информации. В данном процессе в качестве входной информации будет использоваться выбранный пользователем паттерн, в качестве выходной – текстовые файлы, а в качестве условно-постоянной – путь сохранения и имя сохраняемых текстовых файлов.

**1.2 Инструменты разработки**

Проект будет разрабатываться в среде разработки Delphi 8.Среда разработки Delphi 8 была выбрана в связи с требованием к реализации курсового проекта и в связи с субъективным удобством. В Delphi 8 в отличии от Delphi 7 более удобный интерфейс, упрощённый доступ к коду программы и настройкам среды разработки.

Delphi 8 [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 8 ГБ;
* процессор Intel(R) Core(TM) i5-7200U;
* ОС Windows 10.

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:

* 2 GHz процессор (или более быстрый);
* 1 GB RAM;
* от 70 МB доступного места на жестком диске.

Рекомендуемые:

* 2.5 GHz процессор (или более быстрый);
* 2 GB RAM;
* от 70 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:

* Сохранение информации;
* 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 пользователю в изучении паттернов проектирования Python.

Для реализации задач будут использоваться процедуры. С помощью процедуры, например, будет осуществляться считывание описания паттернов 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 ввода информации и её вывода, 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 паттерны проектирования Python.

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, в котором можно прочитать описание паттернов и посмотреть пример их кода.

About – это модуль, в котором можно узнать информацию о программе и ее разработчике

**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 FormHide(Sender: TObject) | Unit1.pas | Form1 | Прячет форму и выключает музыку |
| 3 procedure FormShow(Sender: TObject) | Unit1.pas | Form1 | Показывает форму и включает музыку |
| 4 procedure Button1Click(Sender: TObject) | Unit2.pas | TButton1 | Кнопка, которая изменяет содержимое формы или закрывает её |

Продолжение Таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 procedure Button2Click(Sender: TObject) | Unit2.pas | TButton2 | Открывает Unit3.pas |
| 6 procedure FormClose(Sender: TObject; var Action: TCloseAction) | Unit2.pas | TForm2 |
| 7 procedure Label4Click(Sender: TObject) | Unit2.pas | TLabel4 |
| 8 procedure Label3Click(Sender: TObject) | Unit2.pas | TLabel3 |
| 9 procedure Label6Click(Sender: TObject) | Unit2.pas | TLabel6 | Изменяет содержимое формы или открывает Unit3.pas |
| 10 procedure Label9Click(Sender: TObject) | Unit2.pas | TLabel9 | Открывает Unit3.pas |
| 11 procedure Label12Click(Sender: TObject) | Unit2.pas | TLabel12 |
| 12 procedure Label15Click(Sender: TObject); | Unit2.pas | TLabel15 |
| 13 procedure Label18Click(Sender: TObject) | Unit2.pas | TLabel18 |
| 14 procedure Label21Click(Sender: TObject) | Unit2.pas | TLabel21 |
| 15 procedure Label24Click(Sender: TObject) | Unit2.pas | TLabel24 |
| 16 procedure Label27Click(Sender: TObject) | Unit2.pas | TLabel27 |
| 17 procedure Label30Click(Sender: TObject) | Unit2.pas | TLabel30 |
| 18 procedure Label7Click(Sender: TObject) | Unit2.pas | TLabel7 |
| 19 procedure Label10Click(Sender: TObject) | Unit2.pas | TLabel10 |
| 19 procedure Label13Click(Sender: TObject) | Unit2.pas | TLabel13 |
| 20 procedure Label16Click(Sender: TObject) | Unit2.pas | TLabel16 |
| 21 procedure Label19Click(Sender: TObject) | Unit2.pas | TLabel19 |
| 22 procedure Label22Click(Sender: TObject) | Unit2.pas | TLabel22 |
| 23 procedure Label25Click(Sender: TObject) | Unit2.pas | TLabel25 |
| 24 procedure Label28Click(Sender: TObject) | Unit2.pas | TLabel28 |

Продолжение Таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 25 procedure Label31Click(Sender: TObject) | Unit2.pas | TLabel31 | Надпись, которая открывает Unit3.pas |
| 26 procedure Button5Click(Sender: TObject) | Unit2.pas | TButton5 | Кнопка, которая открывает Unit3.pas |
| 27 procedure Button6Click(Sender: TObject) | Unit2.pas | TButton6 |
| 28 procedure N2Click(Sender: TObject) | Unit2.pas | TMainItem | открывает About.pas |
| 29 procedure Button3Click(Sender: TObject) | Unit2.pas | TButton3 | Кнопка, которая изменяет содержимое формы |
| 31 procedure Button4Click(Sender: TObject) | Unit2.pas | TButton6 |
| 32 procedure N3Click(Sender: TObject) | Unit2.pas | TMainItem | Открывает справку программы |
| 33 procedure N3Click(Sender: TObject) | Unit3.pas | TMainItem | Открывает About.pas |
| 34 procedure N2Click(Sender: TObject) | Unit3.pas | TMainItem | Сохраняет текст из TMemo1 в текстовый файл |
| 35 procedure N4Click(Sender: TObject) | Unit3.pas | TMainItem | Открывает справку программы |
| 36 procedure Button1Click(Sender: TObject) | Unit3.pas | TButton1 | Возвращает на Unit2.pas |
| 37 procedure FormClose(Sender: TObject; var Action: TCloseAction) | Unit2.pas  Unit3.pas | Form2  Form3 | Закрывает Unit1.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 MediaPlayer1 | Form2 | Плеер |
| 3 Button1 | Form2 | Кнопка «Назад»/«Выход» |
| 4 Button2 | Form2 | Кнопка «ПОРОЖДАЮЩИЕ ПАТТЕРНЫ» |
| 5 Button3 | Form2 | Кнопка «СТРУКТУРНЫЕ ПАТТЕРНЫ» |
| 6 Button4 | Form2 | Кнопка «ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ» |
| 7 Button5 | Form2 | Кнопка «Наивный» |
| 8 Button6 | Form2 | Кнопка «Мультипоточный» |

Продолжение Таблицы 2

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 9 N1 | Form2 | Кнопка «Помощь» |
| 10 N2 | Form2 | Кнопка «О Программе» |
| 11 N3 | Form2 | Кнопка «Справка» |
| 12 Image1 | Form1 | Картинка заставки |
| 12 Image1 | Form2 | Картинка для Lable2 |
| 15 Image2 | Form2 | Картинка для Lable5 |
| 16 Image3 | Form2 | Картинка для Lable8 |
| 17 Image4 | Form2 | Картинка для Lable11 |
| 18 Image5 | Form2 | Картинка для Lable14 |
| 19 Image6 | Form2 | Картинка для Lable17 |
| 20 Image7 | Form2 | Картинка для Lable20 |
| 21 Image8 | Form2 | Картинка для Lable23 |
| 22 Image9 | Form2 | Картинка для Lable26 |
| 23 Image10 | Form2 | Картинка для Lable29 |
| 24 Image11 | Form2 | Картинка для «Одиночки» |
| 26 Lable1 | Form2 | Заголовок |
| 27 Lable3, Lable6, Lable9, Lable12, Lable15, Lable18, Lable21, Lable24, Lable27, Lable30 | Form2 | Надпись «Описание» |
| 28 Lable4, Lable7, Lable10, Lable13, Lable16, Lable19, Lable22, Lable25, Lable28, Lable31 | Form2 | Надпись «Пример кода» |
| 29 Lable2, Lable5, Lable8, Lable11, Lable14, Lable17, Lable20, Lable23, Lable26, Lable29 | Form2 | Надпись «Lable2» |
| 30 N1 | Form3 | Кнопка «Файл» |
| 31 N2 | Form3 | Кнопка «Сохранить» |
| 32 N3 | Form3 | Кнопка «О программе» |
| 33 N4 | Form3 | Кнопка «Справка» |
| 34 Memo1 | Form3 | Теория |
| 35 Image1 | AboutBox | Изображение-иконка приложения |
| 36 Product Name | AboutBox | Надпись «Product Name: Паттерны программирования Python» |
| 37 Version | AboutBox | Надпись «Version: 4.1» |
| 38 Lable1 | AboutBox | Надпись «Devoloper: Колпаков Даниил Андреевич ПЗТ-33» |
| 39 OKButton | AboutBox | Кнопка «Ок» |

**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 (\Content\Description\Behavioral\) Chain\_of\_responsibility.txt, Command.txt, Iterator.txt, Mediator.txt, Memento.txt, Observer.txt, State.txt, Strategy.txt, Template\_method.txt, Visitor.txt  (\Content\Description\Generative\)  abstract-factory.txt, builder.txt, factory-method.txt, prototype.txt, singleton.txt  (\Content\Description\Structural\)  adapter.txt, bridge.txt, composite.txt, decorator.txt, facade.txt, flyweight.txt, proxy.txt | Текстовые файлы, используются для хранения описания паттернов |

Продолжение Таблицы 3

|  |  |
| --- | --- |
| 1 | 2 |
| 3 (\Content\CodeExample\Behavioral\) Chain\_of\_responsibility.txt, Command.txt, Iterator.txt, Mediator.txt, Memento.txt, Observer.txt, State.txt, Strategy.txt, Template\_method.txt, Visitor.txt  (\Content\ CodeExample\Generative\)  abstract-factory.txt, builder.txt, factory-method.txt, prototype.txt, singleton.txt  (\Content\ CodeExample\Structural\)  adapter.txt, bridge.txt, composite.txt, decorator.txt, facade.txt, flyweight.txt, proxy.txt | Текстовые файлы, используются для хранения кодов-примеров паттернов |
| 4 ruby.ico | Файл иконки, используется как иконка программы |
| 5 ruby.png | Файл изображения, используется как изображение в About.pas |
| 6 Заполнитель.jpg | Файл изображения, используется как изображение заставки |
| 7 (\Pictures\ Behavioral\)  chain-of-responsibility-mini.bmp, command-mini.bmp, iterator-mini.bmp, mediator-mini.bmp, memento-mini.bmp, observer-mini.bmp, state-mini.bmp, strategy-mini.bmp, template-method-mini.bmp, visitor-mini.bmp  (\Pictures\ Generative\)  abstract-factory-mini.bmp, builder-mini.bmp, factory-method-mini.bmp, prototype-mini.bmp, singleton-mini.bmp  (\Pictures\ Structural\)  adapter-mini.bmp, bridge-mini.bmp, composite-mini.bmp, decorator-mini.bmp, facade-mini.bmp, flyweight-mini.bmp, proxy-mini.bmp | Файлы изображения, которые используются как изображения паттернов |
| 8 elevator-background-music.wav | Звуковой файл, используется как музыка для заставки |
| 9 Manual.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 теории | Вывод теории, как веб-страница | Вывод теории, как веб-страница | Выполнено |
| Корректное начало воспроизведения и остановка воспроизведения музыки | Музыка должна воспроизводиться при появления заставки и останавливаться при скрытии заставки | Проигрывание и остановка музыки при появлении и скрытии заставки соответственно | Выполнено |
| Корректное сохранение текстовых файлов | Файлы сохраняются в указанном месте и под указанным именем | Файлы сохраняются в указанном месте и под указанным именем | Выполнено |
| Проверка работы кнопок | Кнопки делает свою работу | Кнопки делает 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).

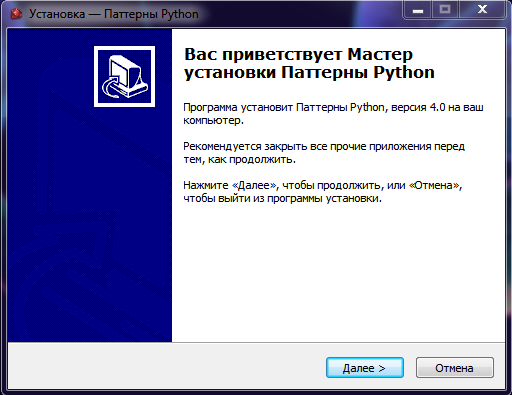


Рисунок 1 – Установка программы

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

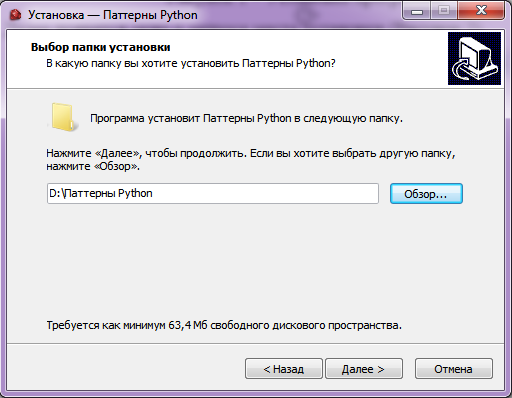


Рисунок 2 – Выбор места для установки

Далее выводится окно о том, что всё готово для начала установки (Рисунок 3).

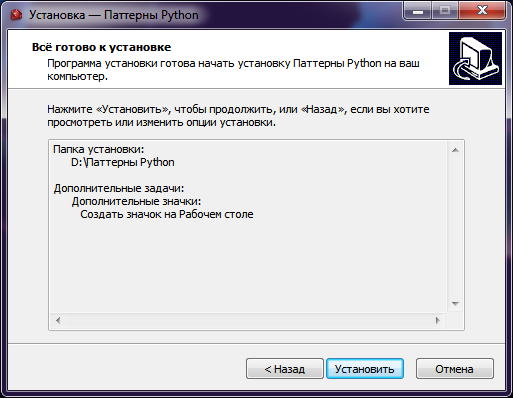


Рисунок 3 – Окно о том, что всё готово для начала установки

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 «Комбинаторик» (Рисунок 4).



Рисунок 4 – Ярлык для захода в программу

Вторым способом является Пуск. Нужно нажать меню Пуск и нажать на приложение «Комбинаторик» (Рисунок 5).



Рисунок 5 – Приложение для захода в программу

**5.2.3 Инструкции по работе с программой**

После запуска на экране появляется заставка (Рисунок 6).

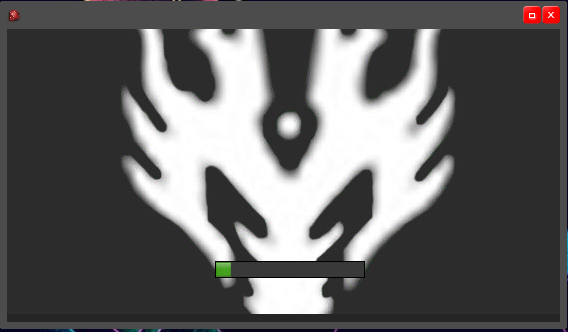
****

Рисунок 6 – Заставка

Затем появляется и главная форма (Рисунок 7), благодаря которой пользователь может увидеть кнопки с помощью которых можно перейти на форму с теорией, а также кнопку меню «Помощь» с помощью неё пользователь может больше узнать о программе.

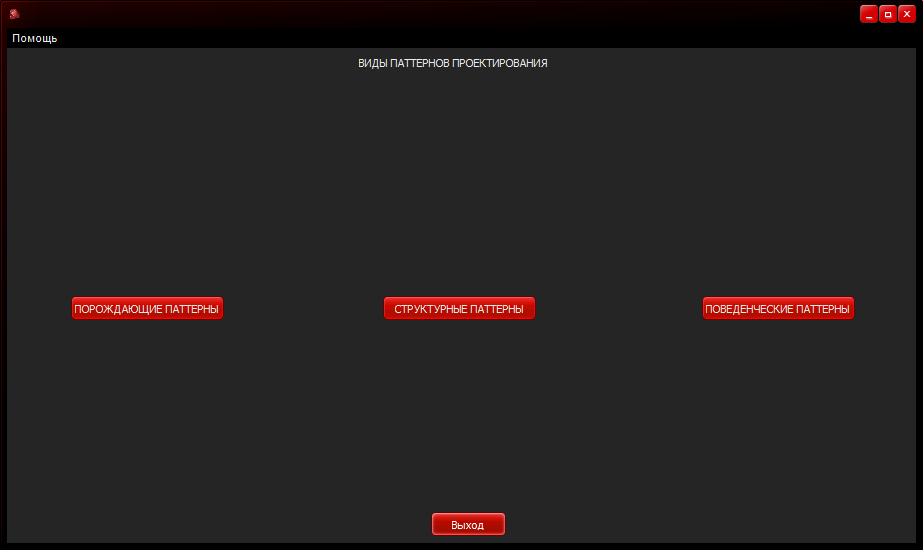
****

Рисунок 7 – Главная форма

После того, как пользователь нажал, например, на кнопку «ПОРОЖДАЮЩИЕ ПАТТЕРНЫ», то увидит окно со списком паттернов(Рисунок 8).

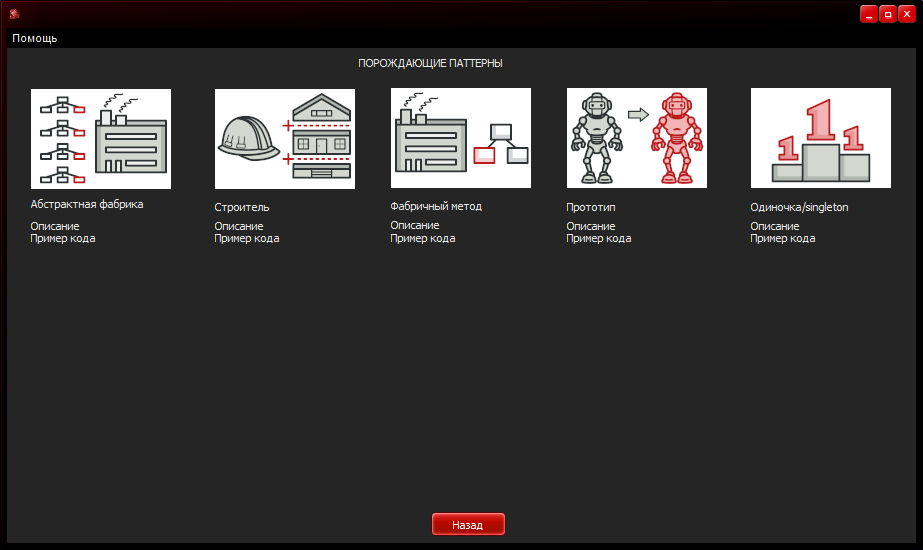


Рисунок 8 – Список паттернов

После нажатия на одну из надписей, «Описание» или «Пример кода» пользователь увидит окно с теорией(Рисунок 9).

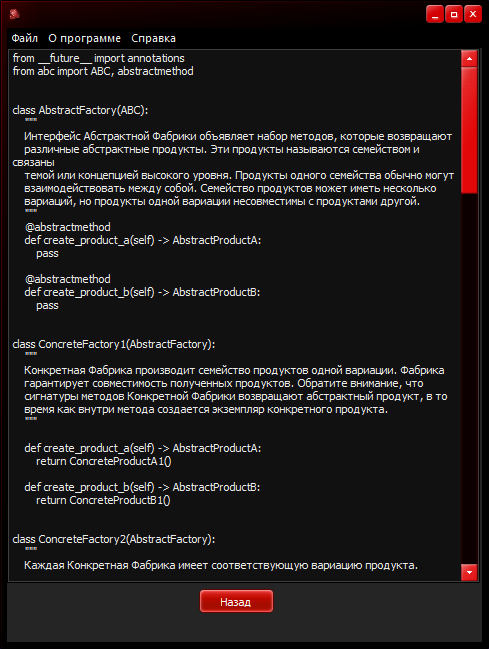
****

Рисунок 9 – Окно с теорией

После нажатия на кнопку Файл->Сохранить пользователь сохранит на рабочем столе текстовый файл(Рисунок 12), при сохранении появиться окно предупреждающее о месте сохранения и о имени сохраненного файла(Рисунок 10, Рисунок 11).

****

Рисунок 10 – Сохранение текстовых файлов с помощью кнопки «Сохранить»

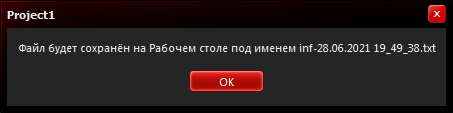


Рисунок 11 – Предупреждающее окно



Рисунок 12 – Сохраннёный текстовый файл

Нажав на кнопку «О программе»(Рисунок 13) появится окно в котором пользователь сможет подробнее узнать о программе и разработчике программы.

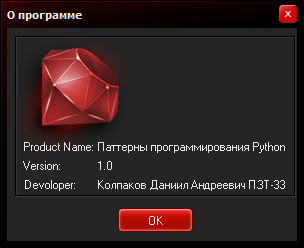


Рисунок 13 – окно «О программе»

В любой момент пользователь может вызвать справку(Рисунок 14) и узнать подробности о программе.

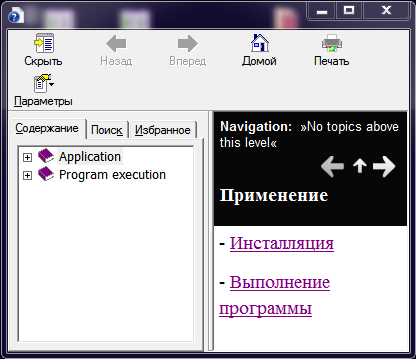


Рисунок 14 – Окно справки

**Заключение**

Целью данного курсового проекта являлась разработка информационно-справочное приложение «Паттерны в Python».

В ходе реализации поставленной задачи был укреплён пройденный курс программирования в среде Delphi 8EX, 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** Refactoring.guru [Электронный ресурс]: – Режим доступа:<https://refactoring.guru/ru/design-patterns/catalog>**.** –Дата доступа: 22.06.2021

**2** ВикипедиЯ – Шаблон проектирования [Электронный ресурс]: – Режим доступа:<https://ru.wikipedia.org/wiki/Шаблон_проектирования>**.** –Дата доступа: 22.06.2021

**3** Delphi Sources - Подключение файла справки .chm к приложению [Электронный ресурс]: – Режим доступа:<https://delphisources.ru/forum/showthread.php?t=1754>**.** –Дата доступа: 28.06.2021

**4** YouTube - Подключение справочного файла в делфи [Электронный ресурс]: – Режим доступа: <https://www.youtube.com/watch?v=dYRC2ipiT14>**.** –Дата доступа: 28.06.2021

**5** CyberForum.ru - Как вставлять музыку в программы на Delphi? [Электронный ресурс]: – Режим доступа:<https://www.cyberforum.ru/delphi-multimedia/thread92899.html>**.** –Дата доступа: 29.06.2021

**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.StdCtrls,

Vcl.Imaging.jpeg, DateUtils, Vcl.ComCtrls, Vcl.MPlayer, MMSystem;

type

TForm1 = class(TForm)

Image1: TImage;

Timer1: TTimer;

ProgressBar1: TProgressBar;

MediaPlayer1: TMediaPlayer;

procedure Timer1Timer(Sender: TObject);

procedure FormHide(Sender: TObject);

procedure FormShow(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

MyTime: TTime;

implementation

uses Unit2;

{$R \*.dfm}

procedure TForm1.FormHide(Sender: TObject); //Плавное скрытие формы и выключение музыки

begin

AnimateWindow(handle, 1000, AW\_BLEND or AW\_HIDE);

sndPlaySound(nil, SND\_NODEFAULT or SND\_ASYNC);

end;

procedure TForm1.FormShow(Sender: TObject); //Плавное появление формы и запуск музыки

begin

AnimateWindow(handle, 1000, AW\_CENTER or AW\_BLEND);

sndPlaySound('elevator-background-music.wav', SND\_NODEFAULT or SND\_ASYNC);

end;

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

begin

ProgressBar1.Position:=ProgressBar1.Position+5;

if (ProgressBar1.Position=ProgressBar1.Max) then

begin

Form1.Hide;

Timer1.Enabled := false;

Form2.Show;

end;

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.ExtCtrls, Vcl.Menus, ShellAPI;

type

TForm2 = class(TForm)

Button1: TButton;

Label1: TLabel;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

Image7: TImage;

Image8: TImage;

Image9: TImage;

Image10: TImage;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

Label24: TLabel;

Label25: TLabel;

Label26: TLabel;

Label27: TLabel;

Label28: TLabel;

Label29: TLabel;

Label30: TLabel;

Label31: TLabel;

Button5: TButton;

Button6: TButton;

Image11: TImage;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Label4Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Label3Click(Sender: TObject);

procedure FormShow(Sender: TObject);

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

procedure Label6Click(Sender: TObject);

procedure Label9Click(Sender: TObject);

procedure Label12Click(Sender: TObject);

procedure Label15Click(Sender: TObject);

procedure Label18Click(Sender: TObject);

procedure Label21Click(Sender: TObject);

procedure Label24Click(Sender: TObject);

procedure Label27Click(Sender: TObject);

procedure Label30Click(Sender: TObject);

procedure Label7Click(Sender: TObject);

procedure Label10Click(Sender: TObject);

procedure Label13Click(Sender: TObject);

procedure Label16Click(Sender: TObject);

procedure Label19Click(Sender: TObject);

procedure Label22Click(Sender: TObject);

procedure Label25Click(Sender: TObject);

procedure Label28Click(Sender: TObject);

procedure Label31Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure N3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit1, Unit3, About;

{$R \*.dfm}

procedure TForm2.Button1Click(Sender: TObject); //Кнопка "Выход" и "Назад"

begin

if Button2.Visible = False and Button3.Visible = False and Button4.Visible = False then

begin

Button1.Caption:= 'Выход';

Button2.Visible:= True;

Button3.Visible:= True;

Button4.Visible:= True;

Button5.Visible:= False;

Button6.Visible:= False;

Image1.Visible:= False;

Image2.Visible:= False;

Image3.Visible:= False;

Image4.Visible:= False;

Image5.Visible:= False;

Image6.Visible:= False;

Image7.Visible:= False;

Image8.Visible:= False;

Image9.Visible:= False;

Image10.Visible:= False;

Image11.Visible:= False;

Label2.Visible:= False;

Label3.Visible:= False;

Label4.Visible:= False;

Label5.Visible:= False;

Label6.Visible:= False;

Label7.Visible:= False;

Label8.Visible:= False;

Label9.Visible:= False;

Label10.Visible:= False;

Label11.Visible:= False;

Label12.Visible:= False;

Label13.Visible:= False;

Label14.Visible:= False;

Label15.Visible:= False;

Label16.Visible:= False;

Label17.Visible:= False;

Label18.Visible:= False;

Label19.Visible:= False;

Label20.Visible:= False;

Label21.Visible:= False;

Label22.Visible:= False;

Label23.Visible:= False;

Label24.Visible:= False;

Label25.Visible:= False;

Label26.Visible:= False;

Label27.Visible:= False;

Label28.Visible:= False;

Label29.Visible:= False;

Label30.Visible:= False;

Label31.Visible:= False;

Label1.Caption:= 'ВИДЫ ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ';

end

else

begin

Form1.Close;

end;

end;

procedure TForm2.Button2Click(Sender: TObject); //Изменение формы под список порождающих паттернов

begin

Button1.Caption:= 'Назад';

Button2.Visible:= False;

Button3.Visible:= False;

Button4.Visible:= False;

Image1.Visible:= True;

Image2.Visible:= True;

Image3.Visible:= True;

Image4.Visible:= True;

Image5.Visible:= True;

Label2.Visible:= True;

Label3.Visible:= True;

Label4.Visible:= True;

Label5.Visible:= True;

Label6.Visible:= True;

Label7.Visible:= True;

Label8.Visible:= True;

Label9.Visible:= True;

Label10.Visible:= True;

Label11.Visible:= True;

Label12.Visible:= True;

Label13.Visible:= True;

Label14.Visible:= True;

Label15.Visible:= True;

Label16.Visible:= True;

Label1.Caption := 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ';

Label2.Caption:='Абстрактная фабрика';

Label5.Caption:='Строитель';

Label8.Caption:='Фабричный метод';

Label11.Caption:='Прототип';

Label14.Caption:='Одиночка/singleton';

Image1.Picture.LoadFromFile('Pictures/Generative/abstract-factory-mini.bmp');

Image2.Picture.LoadFromFile('Pictures/Generative/builder-mini.bmp');

Image3.Picture.LoadFromFile('Pictures/Generative/factory-method-mini.bmp');

Image4.Picture.LoadFromFile('Pictures/Generative/prototype-mini.bmp');

Image5.Picture.LoadFromFile('Pictures/Generative/singleton-mini.bmp');

end;

procedure TForm2.Button3Click(Sender: TObject); //Изменение формы под список структурных паттернов

begin

Button1.Caption:= 'Назад';

Button2.Visible:= False;

Button3.Visible:= False;

Button4.Visible:= False;

Image1.Visible:= True;

Image2.Visible:= True;

Image3.Visible:= True;

Image4.Visible:= True;

Image5.Visible:= True;

Image6.Visible:= True;

Image7.Visible:= True;

Label2.Visible:= True;

Label3.Visible:= True;

Label4.Visible:= True;

Label5.Visible:= True;

Label6.Visible:= True;

Label7.Visible:= True;

Label8.Visible:= True;

Label9.Visible:= True;

Label10.Visible:= True;

Label11.Visible:= True;

Label12.Visible:= True;

Label13.Visible:= True;

Label14.Visible:= True;

Label15.Visible:= True;

Label16.Visible:= True;

Label17.Visible:= True;

Label18.Visible:= True;

Label19.Visible:= True;

Label20.Visible:= True;

Label21.Visible:= True;

Label22.Visible:= True;

Label2.Caption:='Адаптер';

Label5.Caption:='Мост';

Label8.Caption:='Компоновщик';

Label11.Caption:='Декоратор';

Label14.Caption:='Фасад';

Label17.Caption:='Легковес';

Label20.Caption:='Заместитель';

Label1.Caption := 'СТРУКТУРНЫЕ ПАТТЕРНЫ';

Image1.Picture.LoadFromFile('Pictures/Structural/adapter-mini.bmp');

Image2.Picture.LoadFromFile('Pictures/Structural/bridge-mini.bmp');

Image3.Picture.LoadFromFile('Pictures/Structural/composite-mini.bmp');

Image4.Picture.LoadFromFile('Pictures/Structural/decorator-mini.bmp');

Image5.Picture.LoadFromFile('Pictures/Structural/facade-mini.bmp');

Image6.Picture.LoadFromFile('Pictures/Structural/flyweight-mini.bmp');

Image7.Picture.LoadFromFile('Pictures/Structural/proxy-mini.bmp');

end;

procedure TForm2.Button4Click(Sender: TObject); //Изменение формы под список поведенческих паттернов

begin

Button1.Caption:= 'Назад';

Button2.Visible:= False;

Button3.Visible:= False;

Button4.Visible:= False;

Image1.Visible:= True;

Image2.Visible:= True;

Image3.Visible:= True;

Image4.Visible:= True;

Image5.Visible:= True;

Image6.Visible:= True;

Image7.Visible:= True;

Image8.Visible:= True;

Image9.Visible:= True;

Image10.Visible:= True;

Label2.Visible:= True;

Label3.Visible:= True;

Label4.Visible:= True;

Label5.Visible:= True;

Label6.Visible:= True;

Label7.Visible:= True;

Label8.Visible:= True;

Label9.Visible:= True;

Label10.Visible:= True;

Label11.Visible:= True;

Label12.Visible:= True;

Label13.Visible:= True;

Label14.Visible:= True;

Label15.Visible:= True;

Label16.Visible:= True;

Label17.Visible:= True;

Label18.Visible:= True;

Label19.Visible:= True;

Label20.Visible:= True;

Label21.Visible:= True;

Label22.Visible:= True;

Label23.Visible:= True;

Label24.Visible:= True;

Label25.Visible:= True;

Label26.Visible:= True;

Label27.Visible:= True;

Label28.Visible:= True;

Label29.Visible:= True;

Label30.Visible:= True;

Label31.Visible:= True;

Label1.Caption := 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ';

Label2.Caption:='Цепочка обязанностей';

Label5.Caption:='Команда';

Label8.Caption:='Итератор';

Label11.Caption:='Посредник';

Label14.Caption:='Снимок';

Label17.Caption:='Наблюдатель';

Label20.Caption:='Состояние';

Label23.Caption:='Стратегия';

Label26.Caption:='Шаблонный метод';

Label29.Caption:='Посетитель';

Image1.Picture.LoadFromFile('Pictures/Behavioral/chain-of-responsibility-mini.bmp');

Image2.Picture.LoadFromFile('Pictures/Behavioral/command-mini.bmp');

Image3.Picture.LoadFromFile('Pictures/Behavioral/iterator-mini.bmp');

Image4.Picture.LoadFromFile('Pictures/Behavioral/mediator-mini.bmp');

Image5.Picture.LoadFromFile('Pictures/Behavioral/memento-mini.bmp');

Image6.Picture.LoadFromFile('Pictures/Behavioral/observer-mini.bmp');

Image7.Picture.LoadFromFile('Pictures/Behavioral/state-mini.bmp');

Image8.Picture.LoadFromFile('Pictures/Behavioral/strategy-mini.bmp');

Image9.Picture.LoadFromFile('Pictures/Behavioral/template-method-mini.bmp');

Image10.Picture.LoadFromFile('Pictures/Behavioral/visitor-mini.bmp');

end;

procedure TForm2.Button5Click(Sender: TObject); //Пример кода "наивный одиночка"

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Generative/singleton\_naive.txt');

end;

procedure TForm2.Button6Click(Sender: TObject); //пример кода "мультипоточный одиночка"

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Generative/singleton\_multithreaded.txt');

end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction); //закрытие

begin //формы

Form1.Close;

end;

procedure TForm2.FormShow(Sender: TObject); //Плавное появление окна

begin

Button1.Caption:= 'Выход';

AnimateWindow(handle, 800, AW\_CENTER or AW\_BLEND);

end;

procedure TForm2.Label10Click(Sender: TObject); //примеры кодов "Итератор", "Фабричный метод" и "Компоновщик"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Generative/factory-method.txt');

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Structural/composite.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Behavioral/Iterator.txt');

end;

procedure TForm2.Label12Click(Sender: TObject); //Описание "Прототип", "Декоратор" и "Посредник"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Generative/prototype.txt');

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Structural/decorator.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/Mediator.txt');

end;

procedure TForm2.Label13Click(Sender: TObject); //примеры кодов "Прототип", "Декоратор" и "Посредник"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Generative/prototype.txt');

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Structural/decorator.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Behavioral/Mediator.txt');

end;

procedure TForm2.Label15Click(Sender: TObject); //Описание "Одиночки", "Фасад" и "Снимок"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Generative/singleton.txt');

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Structural/facade.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/Memento.txt');

end;

procedure TForm2.Label16Click(Sender: TObject); //Изменение формы под выбор вида "Одиночки"

begin //примеры кодов "Фасад" и "Снимок"

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

begin

Button5.Visible:= True;

Button6.Visible:= True;

Button2.Visible:= False;

Button3.Visible:= False;

Button4.Visible:= False;

Image1.Visible:= False;

Image2.Visible:= False;

Image3.Visible:= False;

Image4.Visible:= False;

Image5.Visible:= False;

Image6.Visible:= False;

Image7.Visible:= False;

Image8.Visible:= False;

Image9.Visible:= False;

Image10.Visible:= False;

Image11.Visible:= True;

Label2.Visible:= False;

Label3.Visible:= False;

Label4.Visible:= False;

Label5.Visible:= False;

Label6.Visible:= False;

Label7.Visible:= False;

Label8.Visible:= False;

Label9.Visible:= False;

Label10.Visible:= False;

Label11.Visible:= False;

Label12.Visible:= False;

Label13.Visible:= False;

Label14.Visible:= False;

Label15.Visible:= False;

Label16.Visible:= False;

Label17.Visible:= False;

Label18.Visible:= False;

Label19.Visible:= False;

Label20.Visible:= False;

Label21.Visible:= False;

Label22.Visible:= False;

Label23.Visible:= False;

Label24.Visible:= False;

Label25.Visible:= False;

Label26.Visible:= False;

Label27.Visible:= False;

Label28.Visible:= False;

Label29.Visible:= False;

Label30.Visible:= False;

Label31.Visible:= False;

Label1.Caption:= 'ОДИНОЧКА'

end;

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Structural/facade.txt');

end;

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Behavioral/Memento.txt');

end;

end;

procedure TForm2.Label18Click(Sender: TObject); //Описание "Легковес" и "Наблюдатель"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Structural/flyweight.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/Observer.txt');

end;

procedure TForm2.Label19Click(Sender: TObject); //примеры кодов "Легковес" и "Наблюдатель"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Structural/flyweight.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExamle/Behavioral/Observer.txt');

end;

procedure TForm2.Label21Click(Sender: TObject); //Описание "Заместитель" и "Состояние"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Structural/proxy.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/State.txt');

end;

procedure TForm2.Label22Click(Sender: TObject); //примеры кодов "Заместитель" и "Состояние"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Structural/proxy.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Behavioral/State.txt');

end;

procedure TForm2.Label24Click(Sender: TObject); //Описание "Стратегия"

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/Strategy.txt');

end;

procedure TForm2.Label25Click(Sender: TObject); //примеры кода "Стратегия"

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Behavioral/Strategy.txt');

end;

procedure TForm2.Label27Click(Sender: TObject); //Описание "Шаблонный метод"

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/Template\_method.txt');

end;

procedure TForm2.Label28Click(Sender: TObject); //примеры кода "Шаблонный метод"

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Behavioral/Template\_method.txt');

end;

procedure TForm2.Label30Click(Sender: TObject); //Описание "Посетитель"

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/Visitor.txt');

end;

procedure TForm2.Label31Click(Sender: TObject); //примеры кода "Посетитель"

begin

Form2.Hide;

Form3.Show;

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Behavioral/Visitor.txt');

end;

procedure TForm2.Label3Click(Sender: TObject); //Описание "Абстрактная фабрика", "Адаптер" и "Цепочка обязанностей"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Generative/abstract-factory.txt');

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Structural/adapter.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/Chain\_of\_responsibility.txt');

end;

procedure TForm2.Label4Click(Sender: TObject); //примеры кодов "Абстрактная фабрика", "Адаптер" и "Цепочка обязанностей"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Generative/abstract-factory.txt');

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Structural/adapter.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Behavioral/Chain\_of\_responsibility.txt');

end;

procedure TForm2.Label6Click(Sender: TObject); //Описание "Строитель", "Мост" и "Команда"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Generative/builder.txt');

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Structural/bridge.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/Command.txt');

end;

procedure TForm2.Label7Click(Sender: TObject); //примеры кодов "Строитель", "Мост" и "Команда"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Generative/builder.txt');

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Structural/bridge.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/CodeExample/Behavioral/Command.txt');

end;

procedure TForm2.Label9Click(Sender: TObject); //Описание "Фабричный метод", "Компоновщик" и "Итератор"

begin

Form2.Hide;

Form3.Show;

if Label1.Caption = 'ПОРОЖДАЮЩИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Generative/factory-method.txt');

if Label1.Caption = 'СТРУКТУРНЫЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Structural/composite.txt');

if Label1.Caption = 'ПОВЕДЕНЧЕСКИЕ ПАТТЕРНЫ' then

Form3.Memo1.Lines.LoadFromFile('Content/Description/Behavioral/Iterator.txt');

end;

procedure TForm2.N2Click(Sender: TObject); // Вызов "О программе"

begin

AboutBox.ShowModal;

end;

procedure TForm2.N3Click(Sender: TObject); //Вызов "Справка"

begin

ShellExecute(0,PChar('Open'),PChar('Manual.chm'),nil,nil,SW\_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.StdCtrls, Vcl.Buttons, Vcl.Menus, ShellAPI;

type

TForm3 = class(TForm)

Memo1: TMemo;

Button1: TButton;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

procedure N3Click(Sender: TObject);

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

procedure Button1Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure N4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

value: integer;

implementation

uses Unit2, About, Unit1;

{$R \*.dfm}

procedure TForm3.Button1Click(Sender: TObject); //Кнопка "Назад"

begin

Form3.Hide;

Form2.Show;

Form2.Button1.Caption:='Назад';

end;

procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction); //закрытие

begin //программы

Form1.Close;

end;

procedure TForm3.N2Click(Sender: TObject); //кнопка "Сохранение"

var name:string;

inf:TDateTime;

begin

Randomize;

name:=GetEnvironmentVariable('username');

inf:=Date+Time;

name:='C:\Users\'+name+'\Desktop\inf-'+StringReplace(Datetostr(Date)+ ' ' +Timetostr(Time),':','\_',[rfReplaceAll])+'.txt';

memo1.Lines.SaveToFile(name);

name:='Файл будет сохранён на Рабочем столе под именем inf-'+StringReplace(Datetostr(Date)+ ' ' +Timetostr(Time),':','\_',[rfReplaceAll])+'.txt';

ShowMessage(name);

end;

procedure TForm3.N3Click(Sender: TObject); //вызов "О программе"

begin

AboutBox.ShowModal;

end;

procedure TForm3.N4Click(Sender: TObject); //вызов "Справка"

begin

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

end;

end.

unit About;

interface

uses WinApi.Windows, System.SysUtils, System.Classes, Vcl.Graphics,

Vcl.Forms, Vcl.Controls, Vcl.StdCtrls, Vcl.Buttons, Vcl.ExtCtrls;

type

TAboutBox = class(TForm)

Panel1: TPanel;

ProductName: TLabel;

Version: TLabel;

OKButton: TButton;

Label1: TLabel;

Image1: TImage;

private

{ Private declarations }

public

{ Public declarations }

end;

var

AboutBox: TAboutBox;

value: integer;

implementation

{$R \*.dfm}

end.