Содержание

Введение…………………………………………………………………………2

1 Анализ задачи ..............................................................................................................2

1.1 Постановка задачи....................................................................................................6

1.2 Диаграмма вариантов использования ....................................................................8

1.3 Модель данных ........................................................................................................9

1.4 Техническое задание .............................................................................................10

2 Проектирование задачи ............................................................................................25

2.1 Выбор стратегии разработки и модели жизненного цикла................................25

2.2 Инструменты разработки.......................................................................................30

2.3 Разработка UML-диаграмм ...................................................................................31

2.4 Разработка пользовательского интерфейса .........................................................34

2.5 Разработка плана работы над проектом...............................................................39

3 Реализация..................................................................................................................39

3.1 Руководство программиста ...................................................................................39

4 Тестирование .............................................................................................................61

5 Руководство пользователя........................................................................................62

Заключение...................................................................................................................70

Список использованных источников ........................................................................71

Приложение А .............................................................................................................72

**Введение**

На учебную практику была поставлена задача – разработать электронное средство обучения по изучению системы ИСО по ОС.

Цель проекта заключается в разработке приложения, при помощи которого пользователь может изучить дисциплину «Операционные системы» и проверить свои знания, с помощью маленького теста

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. В нем описаны все входные и выходные данные. В подразделе «Инструменты разработки» рассмотрена среда, в которой был создан курсовой проект. Здесь также установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование проекта.

В разделе «Проектирование задачи» рассмотрены основные аспекты разработки программного продукта. Здесь можно узнать об организации данных в контексте среды разработки. Описан пользовательский интерфейс, описаны алгоритмы процессов обработки информации, разработка системы справочной информации.

«Реализация задачи» − это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые использованы при реализации данного проекта. 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 проектом. Смоделированы всевозможные действия пользователя при работе с проектом, начиная от запуска до выхода.

В разделе «Применение» описано назначение, область применения, среды функционирования проекта. 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. **Постановка задачи**

**1.1.1 Организационно-экономическая сущность задачи**

Наименование задачи: электронное средство обучения по дисциплине «Операционные системы»;

Цель разработки: возможность изучения дисциплины «ОС».

Назначение данного программного продукта разрабатывается для людей, желающих понять и изучить дисциплину «ОС».

Периодичность использования ПП: по мере необходимости.

Источники и способы получения данных: единая сборка курса.

**1.1.2 Функциональные требования**

Функции гостя:

1 Регистрация

2 Чтение теории.

Пользователь может выполнять все те же функции, что может выполнять гость, но также появляется:

1 авторизация;

2 просмотр результатов;

3 прохождение теста

**1.1.3 Эксплуатационные требования**

Будет использоваться язык программирования Delphi 7 версии. Требования к надежности: система может быть недоступна не более чем 24 часа в год. У администратора программы должна быть возможность выгрузить и загрузить копию программы.

Требования к интерфейсу: при разработке программы должны быть использованы преимущественного черные и белые оттенки. Грамотный пользовательский интерфейс.

* 1. **Диаграмма вариантов использования**

**Диаграмма** **вариантов** **использования** (**Use** **case** diagram, **диаграмма** **прецедентов**) — **диаграмма**, на которой отражены отношения, существующие между актёрами (actors) и **вариантами** **использования** системы (ВИС). **Диаграмма** **прецедентов** предоставляет возможность заказчику, конечному пользователю и разработчику совместно обсуждать поведение системы и функциональность (Рисунок 1).

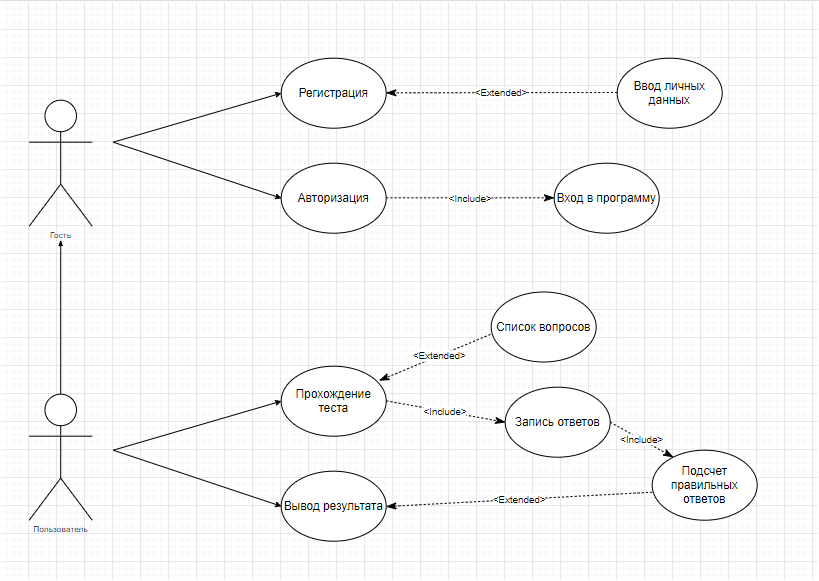


Рисунок 1 – Диаграмма вариантов использования

* 1. **Техническое задание**

**1.4.1 Наименование программы**

Наименование – «ИСО по «ОС»»

**1.4.2 Ответственные лица**

Разработчик: Кеда А.Д.

**1.4.3 Краткая характеристика области применения**

Программа предназначена для изучения дисциплины «ОС»

**1.4.4 Основание для проведения разработки**

Основание для разработки является задание на тему «ИСО по теме «ОС»». Тема согласована с преподавателем, именуемым в дальнейшем заказчиком.

* + 1. **Наименование и условное обозначение темы разработки**

Наименование темы разработки - «Разработка программного обеспечения на тему «ИСО по теме «ОС»». Условное обозначение темы разработки (шифр темы) – «ИСО».

* + 1. **Функциональное назначение**

Программа должна обеспечивать возможность выполнения следующих функций пользователя:

1 переход по формам

2 просмотр теории

3 прохождение теста;

**1.4.7 Требования к функциональным характеристикам**

Чтобы обеспечить возможность полного взаимодействия всех частей разрабатываемого сайта, необходимы следующие задачи:

1 просмотр задач;

2 добавление задачи;

3 редактирование задачи;

4 пометка завершенной задачи;

5 удаление задачи;

6 просмотр выполненной задачи.

**1.4.8** **Требования к надёжности**

Программа должна обеспечивать возможность выполнения функций гостя. Сайт должен производить самовосстановление после сбоя работы.

**1.4.9 Условия эксплуатации**

Приложение требует проведение каких-либо видов обслуживания. Администратор будет обновлять, редактировать и удалять какую-либо информацию на сайте.

**1.4.10 Требования к надежности**

Система может быть недоступна не более чем 24 часа в год.

**1.4.10 Требования к обеспечению надежного (устойчивого) функционирования программы**

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно- технических мероприятий, перечень которых приведен ниже:

1 Организацией бесперебойного питания технических средств;

2 Использование лицензионного программного обеспечения;

3 Регулярным выполнением рекомендаций Министерства труда и

социального развития;

4 Регулярным выполнением требований ГОСТ 51188-98. Защита

информации. Испытания программных средств на наличие

компьютерных вирусов.

**1.4.11 Время восстановления после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30 минут при условии соблюдения условий технических и программных средств. Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

**1.4.12 Отказы из-за некорректных действий оператора**

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

**1.4.13 Условия эксплуатации**

Приложение требует проведение каких-либо видов обслуживания. Администратор будет обновлять, редактировать и удалять какую-либо информацию на сайте.

**1.4.14 Требования к численности и квалификации персонала**

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц – системный администратор и конечный пользователь программы – оператор. Системный администратор должен иметь высшее профильное образование и сертификаты компании.

**2 Проектирования задачи**

**2.1 Выбор стратегии разработки и модели жизненного цикла**

Для выбора подходящей к условиям конкретного проекта модели ЖЦ ПС и систем Институтом качества программного обеспечения SQI рекомендуется использовать специальную процедуру. Данная процедура базируется на применении четырех таблиц вопросов.

Каждая из табл. 1 – 4 представляет одну из категорий классификации проектов. Каждый из вопросов (строка в таблице) предназначен для классификации анализируемого проекта по определенному критерию категории. Столбцы данных таблиц соответствуют обобщенным моделям ЖЦ, фактически представляющим стратегии разработки ПС. При этом под RAD-моделью подразумевается независимая RAD-модель, не встроенная в другие модели жизненного цикла.

Рассматриваемая процедура состоит из следующей последовательности шагов:

1 проанализировать отличительные черты проекта по критериям категорий, представленным в виде вопросов;

2ответить на вопросы по анализируемому проекту, отметив слова «да» или «нет» в соответствующих строках табл. 1 – 4. Если слов «да» или «нет» в строке несколько, необходимо отметить все из них (все «да» или все «нет»).

В качестве примера в табл. 1 выделены варианты ответов для проекта разработки сложного и критичного программного средства, требования к которому заранее не известны и будут уточняться по ходу разработки;

3 расположить по степени важности категории (таблицы) и/или критерии, относящиеся к каждой категории (вопросы внутри таблиц), относительно проекта, для которого выбирается модель ЖЦ;

4выбрать из моделей (см. табл. 1– 4) ту модель, которая соответствует столбцу с наибольшим количеством отмеченных ответов с учетом их степени важности (с наибольшим количеством отмеченных ответов в верхней части приоритетных таблиц). Выбранная модель ЖЦ является наиболее приемлемой для анализируемого проекта.

Таблица 1 – Выбор модели жизненного цикла на основе характеристик требований

На основе результатов заполнения табл. 1 наиболее подходящей является RAD модель, прототипированая, эволюционная.

Таблица 2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков  проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2 | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| 3 | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4 | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| 5 | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6 | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |
| 7 | Итого (Да) | 4 | 4 | 1 | 4 | 3 | 4 |
| 8 | Итого (Нет) | 2 | 2 | 5 | 2 | 3 | 2 |

На основе результатов заполнения табл. 2 наиболее подходящими являются Каскадная и V-образная, модели, RAD и эволюционная модели.

Таблица 3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 5 | 6 | 7 | 8 | 9 |
| 1 | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| 2 | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| 3 | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| 4 | Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |
| 5 | Итого (Да) | 1 | 1 | 1 | 1 | 3 | 3 |
| 6 | Итого (Нет) | 3 | 3 | 3 | 3 | 1 | 1 |

На основе результатов заполнения табл. 3 наиболее подходящими являются прототипирования и эволюционная модели.

Таблица 4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 8 |
| 1 | Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| 2 | Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| 3 | Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| 4 | Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| 5 | Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| 6 | Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7 | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| 8 | Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| 9 | Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| 10 | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |
| 11 | Итого (Да) | 2 | 3 | 3 | 9 | 7 | 9 |
| 12 | Итого (Нет) | 8 | 7 | 7 | 1 | 3 | 1 |

На основе результатов заполнения табл. 4 наиболее подходящей является инкрементная и эволюционная модели.

Продолжение таблицы 4

Исходя из результатов заполнения табл. 1 – 4 (каскадная – 9, V-образная – 10, RAD – 10, инкрементная – 10, быстрого прототипирования – 18, эволюционная – 21) наиболее подходящей моделью, для разработки данного ПП, является эволюционная модель.

**2.2 Инструменты разработки**

Для разработки данного проекта будет выбрана среда разработки WordPress, которая является наиболее актуальной средой для создания приложений данного типа.

Программа обладает удобным и понятным интерфейсом, а также полным набором необходимых функций для создания инсталляторов с минимальным размером, высокой степенью сжатия файлом и приятным интерфейсом.

Для создания пользовательского интерфейса будет использоваться Figma. Данное средство очень удобно бля проектирования ПИ и имеет все необходимые инструменты.

Также в создании проекта использовались такие приложения как:

1. **HTMLHelp -** для создния справочной системы
2. **Inno SetUp -** Для создания установщика
3. **Xmaind, draw.io -** для создания чертежей

Параметры компьютера, который будет использован при разработке:

− процессор Intel Pentium 233 МГц и выше;

− объем оперативной памяти 64 Мбайт (рекомендуется 128 Мбайт);

− объем места на жестком диске 124-520 Мбайт;

− видеокарта NVIDIA ® MX 130 with 2 GB VRAM;

− ОС Windows 95, 98, Vista, XP, 7, 8, 8.1, 10 Home, 10 Pro.

**2.3 Разработка UML-диаграмм**

В ходе создания проекта требовалось создать две UML- диаграммы, что помогли бы понять определенные процессы самого интернет-ресурса. Этими диаграммами являются:

1 Диаграмма деятельности;

2 Диаграмма последовательности;

Диаграмма последовательности – UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта и взаимодействие актеров информационной системы в рамках прецедента (Приложение А).

Диаграмма деятельности – UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов - вложенных видов деятельности и отдельных действий action, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого (Приложение Б).

**3 Реализация**

**3.1 Руководство программиста**

Что бы создать приложение нужно найти хорошую площадку. C помощью Delphi 7 можно на начальном уровне написать подобную программу.

Для начала с официального сайта скачивается Delphi.

После установки создаем формы, на которые переносим нужные компоненты, т.к. таймер, загрузочный экран, кнопки и поля для вводы и вывода текста.

**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 5.

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый результат | Физический результат | Результат тестирования |
| 1 | 2 | 3 | 4 |
| Авторизация | Успешная авторизация | Неуспешная авторизация | Не выполнено |
| Прохождение теста | Возможность загрузить свой ответ на вопрос | Возможность загрузить свой ответ на вопрос | Выполнено |
| Нажатие выхода | При нажатии на выход, производится выход из учетной записи | При нажатии на выход, производится выход из учетной записи | Выполнено |
| Переход между формами | Переход осуществляется между всеми возможными станицами | Переход осуществляется между всеми возможными станицами | Выполнено |

**5 Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Наименованием программного продукта является средство обучения Назначение - для людей любого возраста, которые желают изучить тему ОС. Периодичность использования ПП - по мере необходимости. Программа имеет возможность дать своим пользователем изучение материала и прохождение контроля знаний.

**5.2 Установка программы**

Для того чтобы установить программу, нужно запустить установщик.



Рисунок 5 – Установщик

**5.2.1 Окончание загрузки**



Рисунок 6 – Папка с программой

**5.3 Запуск программы**

Для того, чтобы запустить программу нужно нажать на ярлык на рабочем столе



Рисунок 5 – Файл запускающий программу

**5.4 Интерфейс**

**5.4.1 Загрузочный экран**

При запуске программы, мы увидим загрузочный экран.

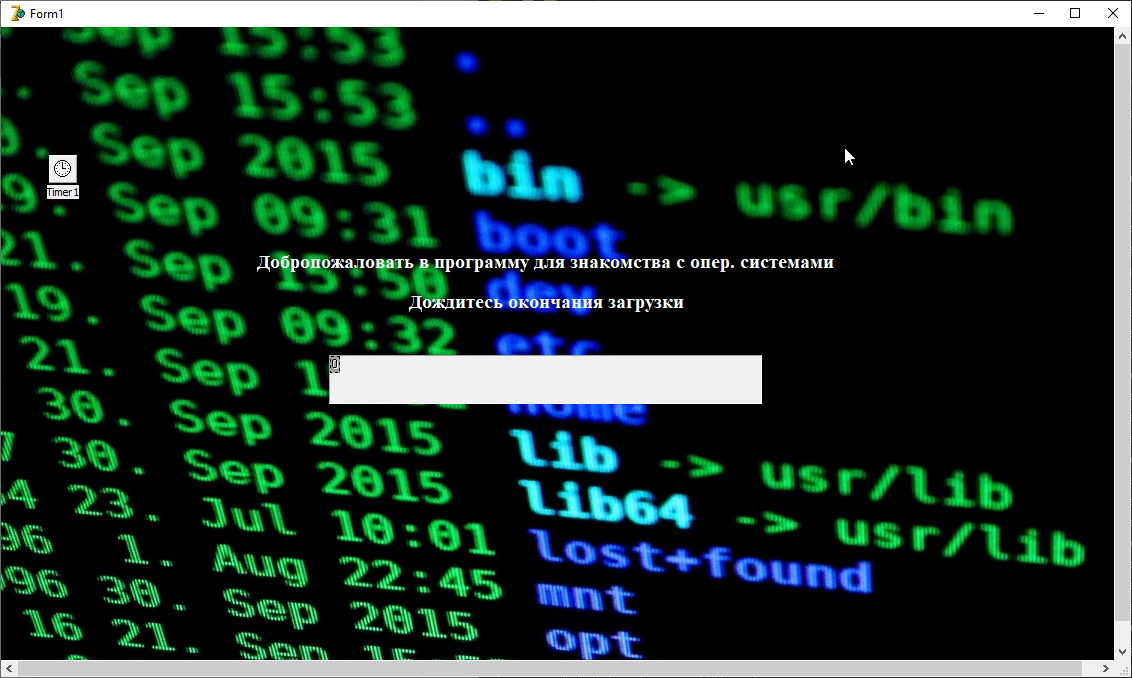


Рисунок 6 – Загрузочный экран

**5.4.2 Экран авторизации**

После окончания загрузки автоматически переходит на экран авторизации

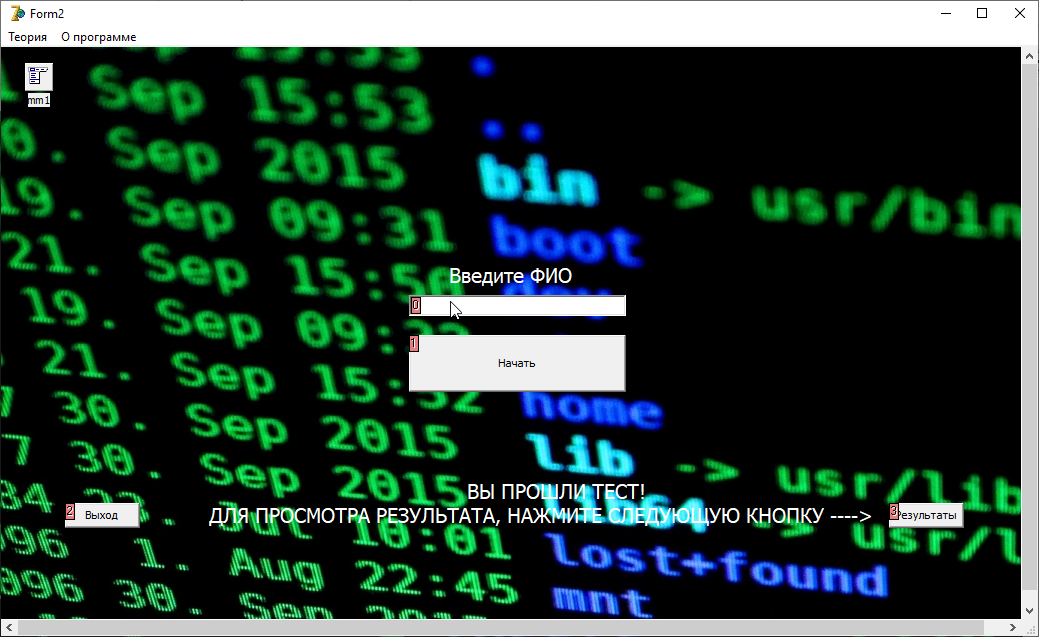


Рисунок 7 – Экран авторизации

**5.4.3 Экран с тестом**

После авторизации и нажатия кнопки, мы переходим на форму с тестом и теорией.

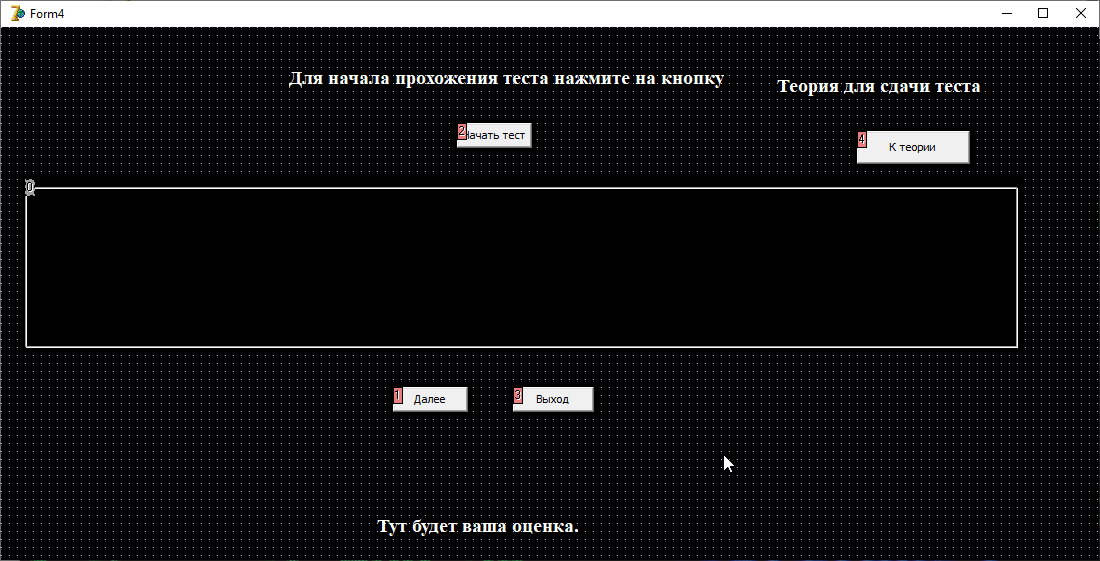


Рисунок 8 – Экран с тестом

**5.4.4 Экран с теорией**

На экране с теорией можно ознакомиться с лекционным материалом.

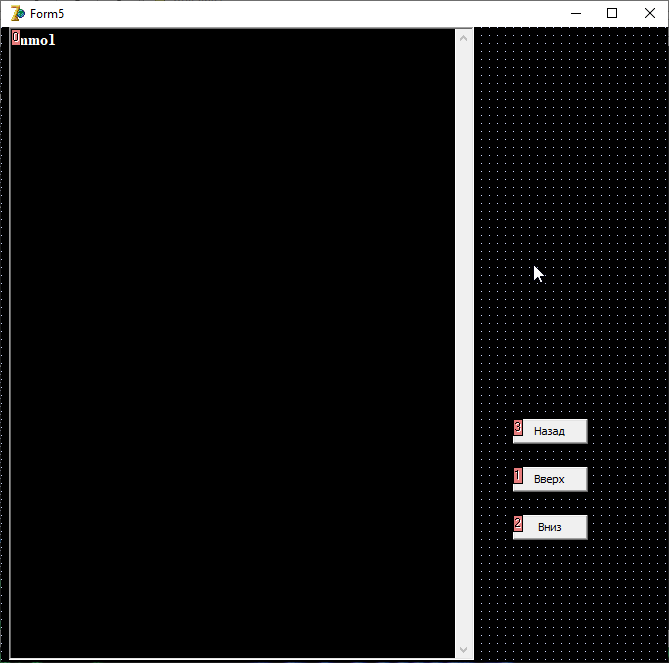


Рисунок 9 – Экран с теорией

**5.4.5 Экран с оценками**

На этом экране можно узнать результаты прохождения теста.

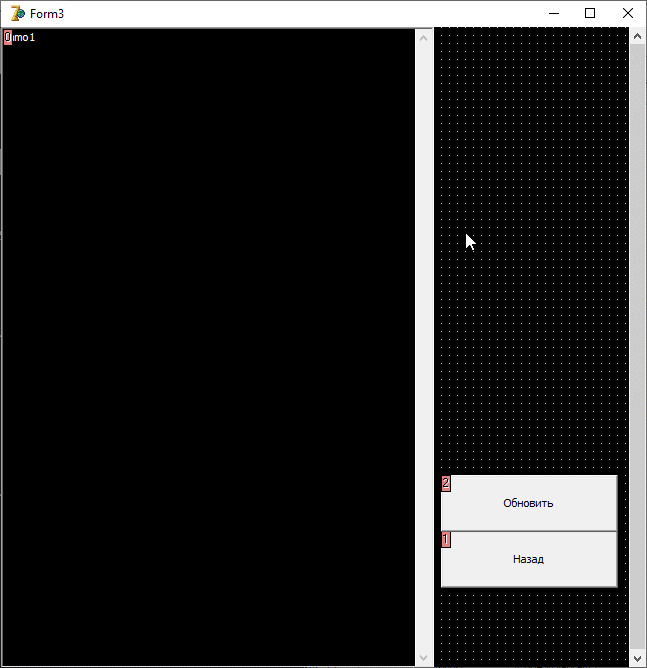


Рисунок 10 – Экран с оценками.

**5.3 Инструкции по работе с ИСО**

1. Запуск

2. Авторизоваться

3 Прохождение теста

**5.4 Завершение работы с ИСО**

1. Перейти на главную форму

2. Нажать кнопку «Выход»

**Заключение**

Целью данного проекта являлась разработка электронного средства обучения по тебе «ОС»

В ходе реализации поставленной задачи был укреплён пройденный изученный материал по дисциплине «Технология разработки программного обеспечения», а также получено много дополнительной информации о её возможностях.

Созданная программа предоставит материал, благодаря которому вы сможете изучить дисциплину «ОС». В нем собрана самая важная информация и предоставлены примеры. Также после изучения теоретического материала можно укрепить и проверить знания пройдя тест.

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

После тщательного тестирования программы 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**

**Руководсво по использованию языка Delphi**

**https://www.helloworld.ru/texts/comp/lang/delphi/delphi1/les00.htm**

#### Приложение А

Дополнение к п. 1. **Анализ задачи**

**1.2.2. Диаграмма классов**

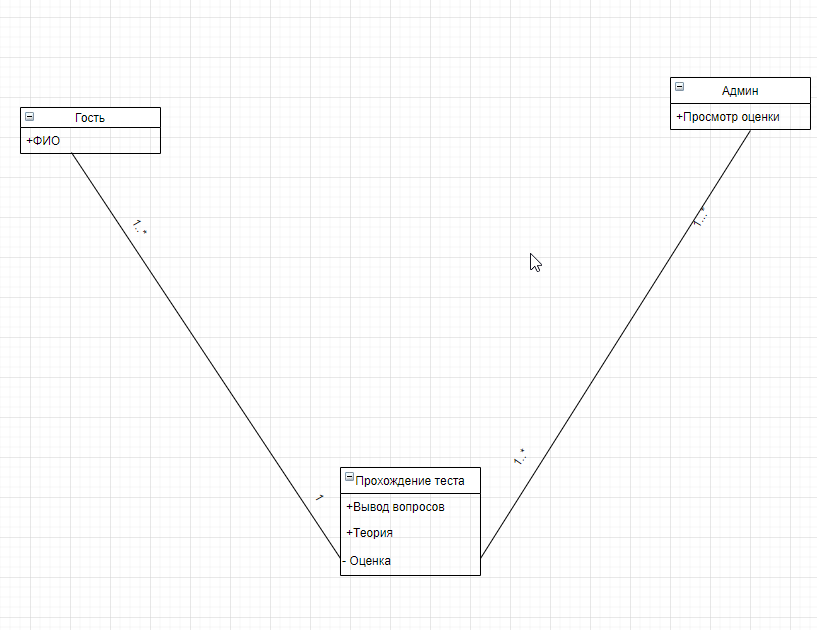


Рисунок 2 – Диаграмма классов использования

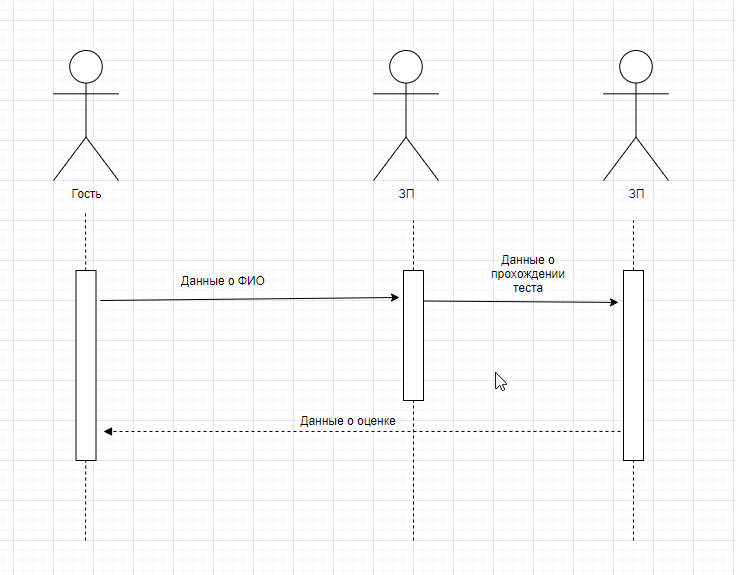
**1.2.3. Диаграмма последовательности**

Рисунок 3 – Диаграмма последованности

**1.2.4. Диаграмма последовательности**

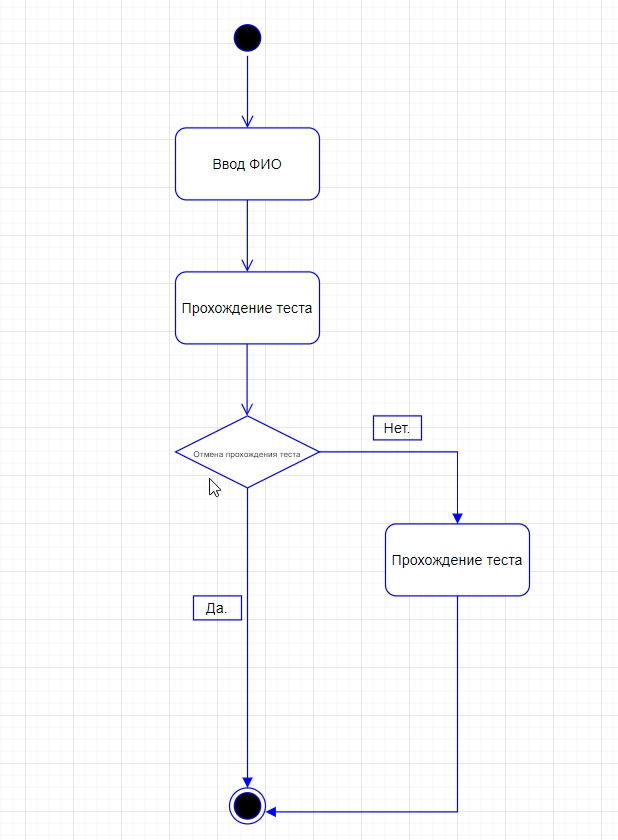


Рисунок 4 – Диаграмма последованности