Департамент образования Вологодской области

Бюджетное профессиональное образовательное учреждение

Вологодской области «Череповецкий химико-технологический колледж»

**КУРСОВАЯ РАБОТА**

по междисциплинарному курсу МДК.02.01 Технология сборки и апробация моделей элементов систем автоматизации с учетом специфики технологических процессов (по отраслям)

специальность: 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Тема: | Разработка онлайн-платформы обучения |

(утверждена приказом колледжа № 17-уч от 19.01.2024 г.)

|  |  |
| --- | --- |
| Выполнил студент | Громов Евгений Владимирович |
| группа 81/2021 | фамилия, имя, отчество подпись |

|  |  |  |
| --- | --- | --- |
| Руководитель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Тагизаде С.Б./ | «\_\_\_»\_\_\_\_\_\_\_\_\_2024 г. |

Дата защиты «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. Оценка защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Череповец, 2024

Изм.

Лист

№ докум.

Подпись

Дата

Лист

1

ППП.09.02.07 ПЗ

Разраб.

Громов Е.В.

Провер.

Тагизаде С.Б.

Н. Контр.

Утверд.

Курсовая работа по теме «Разработка онлайн-платформы обучения»

Лит.

Листов

54

|  |
| --- |
| БПОУ ВО «Череповецкий химико-технологический колледж»  группа 81/2021 |
|

**Содержание**

[Введение 4](#_Toc137416877)

[1 Общая часть 5](#_Toc137416878)

[1.1 Описание предметной области 5](#_Toc137416879)

[1.2 Анализ существующей ситуации 6](#_Toc137416880)

[1.3 Постановка задачи 7](#_Toc137416881)

[1.4 Анализ существующих разработок и обоснование необходимости разработки……… 8](#_Toc137416882)

[2 Специальная часть 10](#_Toc137416883)

[2.1 Выбор технологий и инструментальных средств 10](#_Toc137416884)

[2.1.1 Выбор подхода к разработке 10](#_Toc137416885)

[2.1.2 Выбор среды разработки и языка программирования 11](#_Toc137416886)

[2.2 Разработка спецификаций 13](#_Toc137416887)

[2.2.1 Разработка диаграмм вариантов использования 12](#_Toc137416888)

[2.2.2 Разработка диаграмм последовательностей системы 14](#_Toc137416889)

[2.2.3 Разработка диаграммы пакетов 14](#_Toc137416890)

[2.2.4 Разработка диаграмм классов 1](#_Toc137416891)4

[2.2.5 Разработка диаграммы состояний 16](#_Toc137416892)

[2.2.6 Разработка диаграммы деятельности 17](#_Toc137416893)  
2.2.7 Разработка диаграммы размещения………………………………….17

[2.2.8 Разработка инфологической модели базы данных 18](#_Toc137416894)

[2.2.9 Разработка физической модели базы данных 18](#_Toc137416895)

[2.3 Проектирование программного модуля 18](#_Toc137416896)

[2.3.1 Разработка алгоритмов реализации основных функций программного обеспечения…………. 18](#_Toc137416897)

[2.3.2 Проектирование пользовательского интерфейса 19](#_Toc137416898)

[2.4 Реализация программного обеспечения на выбранном языке программирования и в выбранной среде разработки 21](#_Toc137416899)

[2.5 Выбор стратегии тестирования, разработка тестов, тестирование и отладка программного обеспечения 24](#_Toc137416900)

[3 Безопасность жизнедеятельности 30](#_Toc137416907)

[3.1 Требования к рабочим помещениям и рабочим местам оператора ПК 30](#_Toc137416908)

[3.2 Расчет искусственной освещенности рабочего места оператора ПК 31](#_Toc137416909)

[3.3 Техника безопасности при работе на ПК 33](#_Toc137416910)

[Заключение 36](#_Toc137416911)

[Список используемых источников 37](#_Toc137416912)

# Введение

Цель данной пояснительной записки состоит в том, чтобы представить и обосновать необходимость разработки онлайн-платформы обучения. В современном информационном обществе, где технологии развиваются со стремительной скоростью и доступ к интернету становится всеобщим, возникает все большая потребность в образовании, доступном из любой точки мира и в любое удобное время.

Онлайн-платформы обучения представляют собой современный подход к образованию, который позволяет обучающимся получать знания и навыки через интернет с помощью различных образовательных ресурсов. Они обеспечивают гибкость и удобство для студентов, позволяя им самостоятельно выбирать время и темы обучения, а также адаптировать процесс обучения под свои потребности и возможности.

Однако, существующие платформы часто не соответствуют требованиями современного образования. Возникают проблемы с качеством и доступностью контента, неудобствами в использовании платформы, отсутствием персонализации обучения и отслеживания прогресса студентов.

Таким образом, целью разработки онлайн-платформы обучения является создание инновационного и эффективного образовательного инструмента, который будет отвечать высоким стандартам качества, предлагать широкий выбор образовательных материалов, поддерживать интерактивность и персонализацию обучения, а также предоставлять возможности для отслеживания прогресса и оценки достижений студентов.

# Общая часть

## Описание предметной области

Предметная область данной разработки связана с сферой образования и использованием современных технологий для создания онлайн-платформы обучения. Образование играет ключевую роль в развитии индивидуальных навыков и знаний студентов, а с появлением интернета обучение стало доступным для всех, независимо от места и времени.

Однако, традиционные методы обучения, такие как учебники и лекции, часто ограничиваются вариативностью и не обеспечивают индивидуальный подход к каждому студенту. Вместе с тем, современные технологии и сетевые ресурсы открывают новые возможности для эффективного образования.

Онлайн-платформы обучения предоставляют обучающимся более широкий доступ к качественным образовательным материалам, включая видеолекции, интерактивные упражнения, тесты, а также возможность общения и сотрудничества с преподавателями и другими студентами. Они объединяют обучающихся из разных стран и с разными уровнями подготовки, создавая международное образовательное сообщество.  
Разработка качественной онлайн-платформы обучения требует внимания к различным аспектам, таким как удобство использования интерфейса, обеспечение безопасности и защиты данных, адаптивность под различные типы устройств и операционные системы, а также поддержка различных форматов и стилей образования.

Целью данной разработки является предоставление студентам и преподавателям инструмента, который будет сочетать в себе передовые методы обучения и технологии с учетом конкретных потребностей пользователей. Кроме того, разработка этой платформы может способствовать более эффективному использованию времени, ресурсов и возможностей обучения для всех заинтересованных сторон.

## Анализ существующей ситуации

Анализ существующей ситуации в пояснительной записке по теме разработки онлайн-платформы обучения включает в себя оценку текущего состояния образовательной сферы, использование онлайн-обучения и доступных образовательных ресурсов, анализ потребностей и требований пользователей (студентов, преподавателей, администрации), а также изучение конкурентного окружения и существующих платформ обучения.

Анализ существующей ситуации включает следующие пункты:  
1. Обзор современных тенденций в сфере образования: изучение новых методик обучения, использование информационных технологий в образовании, развитие дистанционного обучения и т.д.  
2. Оценка использования онлайн-платформ обучения в образовательных учреждениях: анализ того, какие онлайн-платформы уже используются, их преимущества и недостатки, уровень удовлетворенности пользователей.  
3. Исследование потребностей и ожиданий пользователей: проведение опросов или интервью с преподавателями, студентами и администрацией образовательных учреждений для выявления их предпочтений и требований к онлайн-платформе обучения.  
4. Анализ конкурентной среды: изучение других платформ обучения, оценка их особенностей, уровня конкуренции и возможных областей для улучшения и отличия от конкурентов.  
5. Инвентаризация доступных ресурсов и экспертиза: определение наличия квалифицированных специалистов, технической поддержки, содержания обучающих курсов и прочих ресурсов, которые могут быть использованы при разработке платформы обучения.

Анализ существующей ситуации позволит лучше понять потребности пользователей, определить ключевые требования к разрабатываемой платформе обучения, а также выявить преимущества и недостатки существующих решений на рынке, что поможет разработать более успешный и эффективный продукт.

## Постановка задачи

Цель разработки онлайн-платформы обучения состоит в создании современного и эффективного инструмента образования, который будет сочетать в себе передовые технологии, качественный контент и персонализированный подход к каждому студенту. Для достижения данной цели необходимо решить следующие задачи:  
1. Разработка удобного и интуитивно понятного интерфейса;  
2. Предоставление качественного образовательного контента;  
3. Поддержка интерактивности и персонализации;  
4. Отслеживание прогресса и оценка достижений;  
5. Улучшение технических аспектов.

Поставленные задачи позволят разработать онлайн-платформу обучения, которая будет удовлетворять потребностям студентов, преподавателей и других заинтересованных сторон.

1.4 Анализ существующих разработок и обоснование необходимости разработки

На рынке уже существуют различные онлайн-платформы обучения, такие как Coursera, Udemy, edX и другие. Они предлагают широкий спектр курсов по самым разным предметам и предлагают доступ к образовательным материалам для тысячи студентов по всему миру. Эти платформы имеют значительное влияние на изменение образовательной парадигмы и демократизацию образования.  
Однако, существующие разработки также имеют некоторые ограничения и недостатки:  
1. Ограниченный выбор курсов: Несмотря на широкий спектр предлагаемых курсов, все еще существуют предметные области, которые не охвачены существующими платформами.  
2. Недостаток интерактивности: Некоторые платформы не обеспечивают достаточную интерактивность и сотрудничество между студентами и преподавателями.  
3. Ограниченная персонализация: Существующие платформы не всегда учитывают индивидуальные потребности и уровень подготовки студентов. Обоснование необходимости разработки:  
На основе вышеупомянутого анализа, формулируется необходимость разработки новой онлайн-платформы обучения. Эта разработка будет иметь ряд преимуществ:  
1. Расширение предметных областей: Разработка новой платформы позволит расширить предлагаемый выбор курсов и материалов, включая менее распространенные предметные области и специализации. Это позволит студентам получить доступ к более широкому спектру образовательных возможностей.  
2. Повышение интерактивности: Новая платформа будет обеспечивать возможность взаимодействия и обмена опытом между студентами и преподавателями.   
3. Персонализация обучения: Новая платформа будет учитывать индивидуальные потребности студентов и предлагать персонализированные курсы и материалы.  
4. Улучшение технических характеристик: Новая платформа будет уделять внимание техническим аспектам, таким как скорость работы, надежность, безопасность данных и совместимость с различными устройствами.

На основе вышеизложенных преимуществ и обоснования необходимости разработки, становится ясным, что новая онлайн-платформа обучения будет иметь ряд значимых преимуществ по сравнению с существующими решениями.

# Специальная часть

## Выбор технологий и инструментальных средств

### 2.1.1 Выбор подхода к разработке

Более подходящим вариантом для разработки онлайн-платформы обучения является объектно-ориентированный подход.

При разработке такой системы важно обеспечить четкое разделение ответственности между различными компонентами системы, такими как базы данных и интерфейсы пользователя. Также необходимо иметь возможность расширять систему для добавления новых функций и взаимодействия с другими системами.

В объектно-ориентированном подходе можно использовать классы и объекты для определения компонентов системы и их свойств и методов, а также для управления взаимодействием между ними. Это позволяет создать более структурированную и удобную для поддержки систему, которая будет легко масштабироваться и приспосабливаться к изменениям в будущем.

Напротив, структурный подход в данном случае может стать слишком ограничивающим и неэффективным в управлении более сложными взаимодействиями между компонентами системы, а также в обеспечении возможности переиспользования компонентов в других системах или проектах.

Кроме того, при использовании структурного подхода в разработке онлайн-платформы обучения может возникнуть проблема повторения кода и возможности конфликта между разными блоками кода, что может затруднить поддержку системы в будущем.

Объектно-ориентированный подход позволяет создать более гибкую и

удобную для использования систему, которая может легко адаптироваться к изменениям бизнес-требований и новым технологиям. Он также упрощает процесс тестирования и отладки, что позволяет более эффективно обслуживать систему в целом.

Таким образом, объектно-ориентированный подход является более предпочтительным для разработки сложных систем, таких как разработка онлайн платформы обучения, где важны гибкость, масштабируемость и работоспособность системы в целом.

### 2.1.2 Выбор среды разработки и языка программирования

Язык C# и Visual Studio Community 2022 являются, как по мне, отличным выбором для разработки онлайн-платформы обучения.

С# — это объектно-ориентированный язык программирования, который имеет сильную типизацию и поддержку сборки мусора. Он разработан Microsoft, и обладает большим набором возможностей по разработке приложений для Windows. С# также поддерживает парадигму функционального программирования, что может быть полезным в разработке более сложных частей системы.

Visual Studio Community 2022 — это интегрированная среда разработки (IDE) от Microsoft, которая обладает множеством функций и инструментов, которые значительно упрощают процесс разработки ПО. Visual Studio Community 2022 предоставляет различные инструменты разработки, включая мощный отладчик, поддержку Git, автоматические рефакторинги и другие.

Среди прочих преимуществ использования C# и Visual Studio Community 2022 для разработки онлайн-платформы обучения:

1. Высокая скорость разработки. Большое количество доступных функций и удобство Visual Studio Community 2022 позволяет быстро создавать и отлаживать код.
2. Широкий спектр библиотек и фреймворков. C# имеет большое количество полезных библиотек и фреймворков для работы с различными частями системы, включая работу с базами данных, сетевыми протоколами и т.д.
3. Большое количество сообщества. C# имеет широкое сообщество разработчиков, которые готовы помочь локализовать проблемы и поделиться знаниями и опытом.

В целом, использование C# и Visual Studio Community 2022 позволит создать быструю, функциональную и надежную автоматизированную систему, которая будет работать на высоком уровне.

## Разработка спецификаций

### 2.2.1 Разработка диаграмм вариантов использования

Диаграмма представляет собой процесс работы программного обеспечения, представлена ниже на рисунке 1.

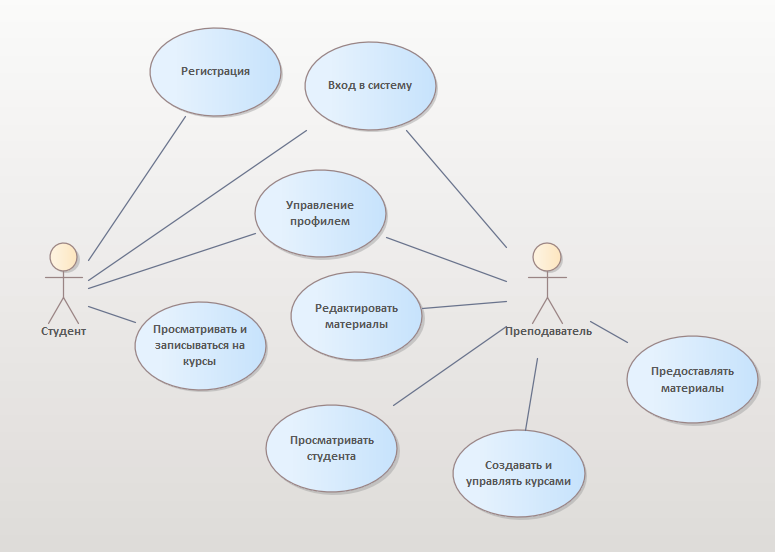


Рисунок 1 – Диаграмма вариантов использования

На этой диаграмме вариантов использования представлены основные акторы: студенты и преподаватели. Они могут регистрироваться и входить в систему, управлять своим профилем, просматривать и редактировать материалы, просматривать и записываться на курсы, и предоставлять свои материалы.

Диаграмма вариантов использования помогает наглядно представить, как различные акторы будут взаимодействовать с онлайн-платформой обучения и какие ключевые функции и возможности будут предоставляться. Это полезный инструмент при планировании и детализации требований к разрабатываемой платформе.

### 2.2.2 Разработка диаграмм последовательностей системы

### 2.2.3 Разработка диаграммы пакетов

Для данной разработки построена диаграмма пакетов, которая показана ниже на рисунке 3. Она показывает зависимости между частями.

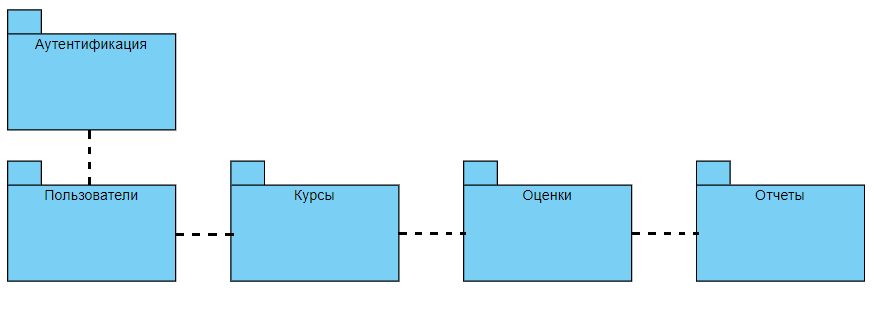


Рисунок 2 – Диаграмма пакетов

### 2.2.4 Разработка диаграмм классов

Для разработки пакета для взаимодействия создана диаграмма классов, показанная ниже на рисунке 3.

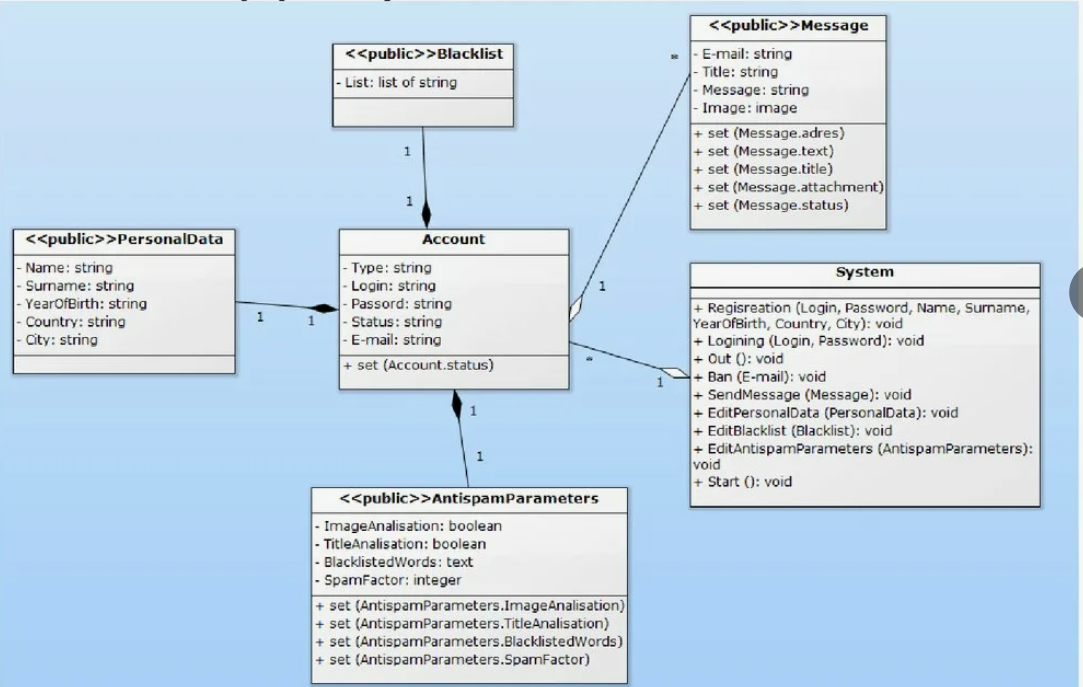


Рисунок 3 – Диаграмма классов

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними.

### 2.2.5 Разработка диаграммы состояний

Рисунок 4 – Диаграмма состояний

### 

### 2.2.6 Разработка диаграммы деятельности

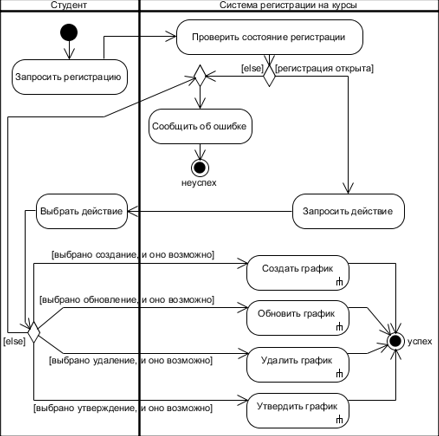


Рисунок 5 – Диаграмма деятельности.

### 2.2.7 Разработка диаграммы размещения

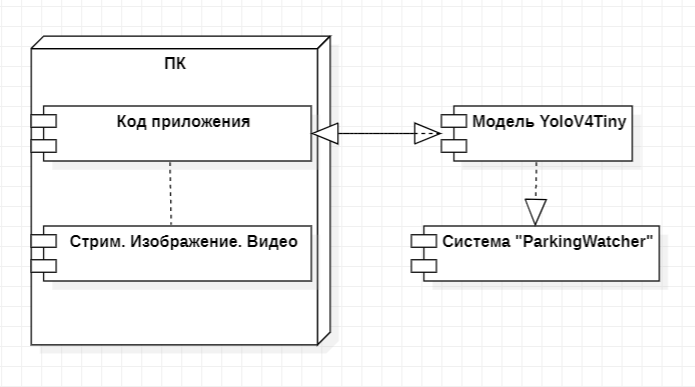
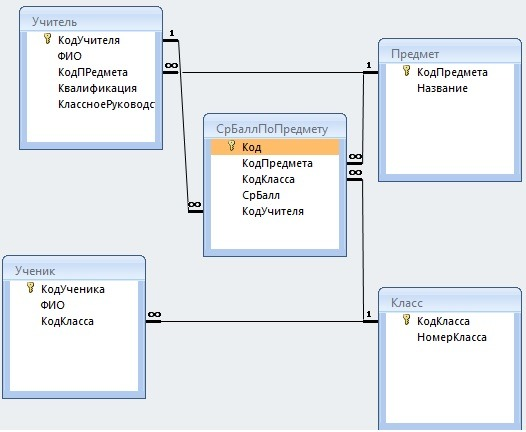


Рисунок 6 – Диаграмма размещения.

### 2.2.8 Разработка инфологической модели базы данных

Для данного все результаты сохраняются в следующем формате   
{"total\_places":50,"occupied\_places":25} (Json)

### 2.2.9 Разработка физической модели базы данных

Пункт 2.2.8

## Проектирование программного модуля

### 2.3.1 Разработка алгоритмов реализации основных функций программного обеспечения

Алгоритм начинается с запуска приложения.

1.Запуск веб-приложения:

Веб-приложение запускается, инициализируя все необходимые компоненты и библиотеки для дальнейшей работы.

2. Преобразование данных:

При запуске веб-приложения так же запускается подпроцесс преобразования данных, например, для анализа поступающих транзакций и расходов. Данные преобразуются в формат, подходящий для дальнейшего анализа.

4. Проверка данных:

Дополнительно проводится проверка данных на наличие пересечений или конфликтов, например, если одни и те же расходы или доходы были учтены несколько раз.

Повторение шагов 1-4 при необходимости позволяет постоянно обновлять и анализировать данные, обеспечивая актуальность и точность финансовой информации.

### 2.3.2 Проектирование пользовательского интерфейса

Процедурно-ориентированные интерфейсы используют традиционную модель взаимодействия с пользователем, основанную на понятиях «процедура» и «операция». В рамках этой модели программное обеспечение предоставляет пользователю возможность выполнения некоторых действий, для которых пользователь определяет соответствующие данные и следствием выполнения которых является получение желаемых результатов.

Объектно-ориентированные интерфейсы используют несколько иную модель взаимодействия с пользователем, ориентированную на манипулирование объектами предметной области. В рамках этой модели пользователю предоставляется возможность напрямую взаимодействовать с каждым объектом и инициировать выполнение операций, в процессе которых взаимодействуют несколько объектов.

Для того, чтобы понять какой интерфейс использовать для разработки проведена небольшая сравнительная характеристика ниже в таблице 1.

Таблица 1 – Сравнительная характеристика интерфейсов

|  |  |  |
| --- | --- | --- |
| № | **Процедурно-ориентированные пользовательские интерфейсы** | **Объектно-ориентированные пользовательские интерфейсы** |
|  | | |
| 1 | Обеспечивают пользователей функциями, необходимыми для выполнения задач | Обеспечивают пользователям возможность взаимодействия с объектами |
| 2 | Акцент делается на задачи | Акцент делается на входные данные и результаты |
| 3 | Пиктограммы представляют приложения, окна или операции | Пиктограммы представляют объекты |
| 4 | Содержание папок и справочников отображается с помощью таблиц и списков | Папки и справочники являются визуальными контейнерами объектов |

У данных интерфейсов также есть типы.

Процедурно-ориентированные и объектно-ориентированные интерфейсы делятся на 4 типа: «примитивные», меню и со свободной навигацией, а также прямого манипулирования. Их сравнение приведено ниже.

Примитивным называют интерфейс, который организует взаимодействие с пользователем в консольном режиме. Обычно такой интерфейс реализует конкретный сценарий работы программного обеспечения.

Интерфейс-меню в отличие от примитивного интерфейса позволяет пользователю выбирать необходимые операции из специального списка, выводимого ему программой. Эти интерфейсы предполагают реализацию множества сценариев работы, последовательность действий в которых определяется пользователем.

Интерфейсы со свободной навигацией также называют графическими пользовательскими интерфейсами - что пользователь видит на экране, то он и получит при печати. Эти названия подчеркивают, что интерфейсы данного типа ориентированы на использование экрана в графическом режиме с высокой разрешающей способностью.

Объектно-ориентированные интерфейсы пока представлены одним видом: интерфейс прямого манипулирования. Этот тип интерфейса предполагает, что взаимодействие пользователя с программным обеспечением осуществляется посредством выбора и перемещения пиктограмм, соответствующих объектам предметной области.

Для данного ПО был выбран интерфейс-меню, так как в отличие от других интерфейсов он позволяет пользователю выбирать необходимые операции из специального списка, выводимого ему программой.

## 2.4 Реализация программного обеспечения на выбранном языке программирования и в выбранной среде разработки

Для реализации программного обеспечения на языке C# для автоматизированной системы мониторинга занятости стояночных мест при помощи программной видеоаналитики можно использовать различные среды разработки, такие как Microsoft Visual Studio, JetBrains Rider, SharpDevelop.

Разработка программы в среде Visual Studio будет представлять собой следующие шаги:

1. Создание проекта.

В меню выбирается пункт "Создать проект", а затем - тип проекта, например, "Windows Forms App".

1. Добавление нужных файлов.

В проект необходимо добавить файлы, отвечающие за обработку видео, выделение областей и прочие нужные для реализации функции.

1. Написание кода.

Код программы написанный на языке C# реализует основные функции, описанные выше. В коде используются методы обработки данных о видео или фотографиях, методы для выделения объектов на изображении, алгоритмы классификации, обучения нейронных сетей и многое другое.

Пример некоторых функций, которые могут быть реализованы:

// Загрузка фото/видео/стрима для дальнейшей обработки и анализа.

private void LoadMedia(string filePath)

{

if (File.Exists(filePath))

{

// загрузка фотографии/видео/стрима

}

else

{

MessageBox.Show("Файла не существует.");

}

}

// Подсчет количества занятых мест

private int CountOccupiedParkingSpaces()

{

// подсчет количества занятых мест

return occupiedParkingSpacesCount;

}

// Подсчет количества свободных мест

private int CountFreeParkingSpaces()

{

// подсчет количества свободных мест

return freeParkingSpacesCount;

}

1. Тестирование и отладка.

После написания кода программа должна быть протестирована и отлажена. Можно использовать готовые тестовые данные или создать свои, чтобы проверить работу программы.

1. Компиляция и сборка.

Если все тесты пройдены успешно, можно скомпилировать и собрать программу. В результате получится файл с расширением ".exe", который пользователь может запустить на своем компьютере.

1. Развертывание.

Если программа должна быть доступна для пользователей, нужно развернуть ее на соответствующих платформах. Это может быть установка на компьютер пользователя, загрузка на сервер, доступ через Интернет и т.д.

Таким образом, реализация программного обеспечения на языке C# для автоматизированной системы мониторинга занятости стояночных мест при помощи программной видеоаналитики требует ряда шагов и использует средства разработки, доступные на рынке. Важно понимать, что конечное решение может сильно отличаться в зависимости от конкретных потребностей и условий задачи.

## 2.5 Выбор стратегии тестирования, разработка тестов, тестирование и отладка программного обеспечения

Для выбора наиболее подходящего метода и инструмента тестирования необходимо учитывать ряд критериев, таких как освоение, эффективность, распространенность и принцип использования.

В табличной форме приведена сравнительная характеристика методов тестирования и инструментов для автоматизации тестирования, соответственно. Так, в таблице 1 представлено сравнение черного, белого и серого ящиков, включая преимущества и недостатки каждого метода. В таблице 2 отражены основные характеристики инструментов Selenium, Selenium IDE и PHPUnit, таких как функциональность, сложность использования и распространенность среди разработчиков.

Однако стоит учитывать, что выбор метода и инструмента тестирования зависит от конкретного проекта и его специфики. Поэтому рекомендуется проводить тщательный анализ требований проекта и оценивать эффективность тестирования на основе выбранного метода и инструмента.

| **Характеристика** | **Черный ящик** | **Белый ящик** | **Серый ящик** |
| --- | --- | --- | --- |
| Описание | В этом методе тестирующий работает со стороны пользователя, не зная деталей внутренней работы программы. | В этом методе тестирующий имеет доступ к исходному коду программы и может проводить тестирование на уровне алгоритмов и структур данных. | Это смешанный метод, который сочетает некоторые элементы "черного ящика" и "белого ящика". Тестирующий имеет ограниченный доступ к исходному коду программы, что позволяет ему проводить более детальное тестирование, однако он все еще не имеет полного доступа ко всем деталям реализации. |
| Объект тестирования | Объектом тестирования является готовое приложение или система в целом. | Объектом тестирования является исходный код и отдельные компоненты программы. | Объектом тестирования является как готовое приложение, так и частично доступный для анализа исходный код. |
| Сложность тестирования | Метод "черного ящика" обычно проще в использовании, так как не требует знания языка программирования или особой экспертизы в данной области. | Метод "белого ящика" может быть более сложным, так как требуется знание языка программирования и понимание внутренней работы программы. | Метод "серого ящика" находится между "черным" и "белым" ящиками по сложности. Тестирующий должен иметь некоторые знания в области программирования, чтобы анализировать доступный ему код и проводить дополнительные тесты. |
| Уровень детализации тестирования | Метод "черного ящика" позволяет проводить тестирование на уровне функциональности приложения, однако не обеспечивает глубокого понимания внутренней работы программы. | Метод "белого ящика" позволяет проводить тестирование на уровне отдельных модулей программы и алгоритмов. | Метод "серого ящика" обеспечивает более детальное тестирование, чем "черный ящик", но не такое глубокое и полное, как "белый ящик". |
| Возможность автоматизации тестирования | Метод "черного ящика" легко поддается автоматизации с помощью специализированных инструментов для автоматического тестирования. | Метод "белого ящика" также может быть автоматизирован, однако требует более тщательной настройки и ручной работы. | Метод "серого ящика" может быть автоматизирован в определенной степени. |

Таблица 2 – Инструменты для автоматизации тестирования

| **Инструмент** | **Описание** | **Функциональность** | **Сложность использования** | **Распространенность** |
| --- | --- | --- | --- | --- |
| Selenium | Фреймворк для автоматизированного тестирования веб-приложений. | Позволяет записывать и воспроизводить действия пользователя на веб-страницах, проверять содержимое элементов интерфейса, реализовать тестовые сценарии. | Высокая сложность использования, так как требует знания языка программирования и разработки автоматических скриптов. | Один из самых популярных инструментов для автоматизации тестирования веб-приложений. |
| Appium | Фреймворк для автоматизированного тестирования мобильных приложений. | Позволяет записывать и воспроизводить действия пользователя на мобильных устройствах, проверять содержимое элементов интерфейса, реализовать тестовые сценарии. | Высокая сложность использования, так как требует знания языка программирования и разработки автоматических скриптов. | Широко используется для автоматизации тестирования мобильных приложений на платформах Android и iOS. |
| JMeter | Инструмент для функционального и нагрузочного тестирования веб-приложений. | Позволяет создавать и запускать тестовые сценарии, генерировать нагрузку на серверы, анализировать результаты. | Умеренная сложность использования, так как требует знания основ работы с веб-приложениями и настройки нагрузочных тестов. | Широко используется для функционального и нагрузочного тестирования веб-приложений. |
| TestComplete | Коммерческий инструмент для автоматизированного тестирования различных типов приложений. | Позволяет записывать и воспроизводить действия пользователя на экране, проверять содержимое элементов интерфейса, реализовать тестовые сценарии. | Относительно низкая сложность использования, так как имеет графический интерфейс и не требует знания языка программирования. | Широко используется для автоматизации тестирования различных типов приложений. |
| Postman | Инструмент для тестирования API. | Позволяет отправлять запросы к API и анализировать ответы, создавать тестовые сценарии для проверки работоспособности API. | Относительно низкая сложность использования, так как имеет графический интерфейс и не требует знания языка программирования. | Широко используется для тестирования API различных веб-сервисов. |

Исходя из таблицы выше и разрабатываемого ПО, необходимо выбрать инструмент для тестирования приложения, которое использует нейросеть для определения количества свободных парковочных мест. Учитывая, что приложение написано на языке C#, можно выделить два наиболее подходящих инструмента:

1 Selenium - этот инструмент позволяет проводить автоматизированное тестирование веб-приложений, в том числе приложений, написанных на языке C#. Он поддерживает множество языков программирования, включая C#, и может использоваться для тестирования в разных браузерах. Однако, Selenium не подходит для тестирования нейросетей.

2 NUnit - это популярный фреймворк для автоматизированного тестирования на языке C#. Он имеет обширную документацию и сообщество разработчиков, что делает его удобным инструментом для работы. Также он поддерживает множество функций и интеграцию с другими инструментами. Кроме того, NUnit позволяет проводить модульное тестирование, что может быть полезным для отдельных компонентов. Он может быть использован для тестирования нейросетей, написанных на языке C#.

# 3 Безопасность жизнедеятельности

## 3.1 Требования к рабочим помещениям и рабочим местам оператора ПК

Рабочие помещения и рабочие места оператора ПК должны соответствовать определенным требованиям, которые обеспечат комфортную и безопасную работу. К ним относятся:

1. Размеры помещения не менее 6 квадратных метров на одно рабочее место.
2. Наличие естественного и искусственного освещения, соответствующего СанПиН 2.2.2./2.4.1340-03 «Гигиенические требования к условиям труда при работе на персональных компьютерах».
3. Температура воздуха в помещении должна находиться в пределах 18-24 градусов Цельсия, влажность - 40-60%.
4. Отсутствие шума и других нежелательных звуковых эффектов.
5. Наличие пожарных средств и путей эвакуации.
6. Наличие мебели, соответствующей правилам эргономики (регулируемая высота, регулируемый наклон стола, спинка и сиденье стула, наличие подставки для ног и т.д.).
7. Наличие специального оборудования для защиты здоровья (экраны, фильтры, подставки для документов и т.д.).
8. Соответствие параметров компьютера установленным нормам.
9. Работа оператора ПК должна проходить в условиях микроклимата, обеспечивающего наиболее комфортные условия для зрения (уровень освещенности должен быть не менее 300 лк).
10. Обеспечение необходимой пространственной ориентации с помощью различных методов размещения монитора и клавиатуры.

Общие требования при использовании компьютерной техники:

1. Защита зрения.
2. Защита органов слуха.
3. Защита органов пищеварения.
4. Защита кожных покровов.
5. Установка расписания работы и время отдыха в течение дня.

## 3.2 Расчет искусственной освещенности рабочего места оператора ПК

Расчет искусственной освещенности рабочего места оператора ПК является необходимым для обеспечения комфортных условий работы с учетом световых параметров, необходимых для нормального функционирования зрительного аппарата человека.

Согласно требованиям СанПиН 2.2.2./2.4.1340-03, искусственное освещение должно обеспечивать не менее 300 лк на поверхности стола и не менее 500 лк на поверхности клавиатуры и экрана монитора.

Для расчета искусственной освещенности рабочего места оператора ПК необходимо знать:

1. Площадь рабочего помещения (S).
2. Коэффициент использования светового потока (K) - показатель, показывающий, какая часть из общей мощности светильника будет использоваться для освещения рабочей поверхности.
3. Световой поток, выделяемый одним светильником (F).
4. Коэффициент освещенности помещения, отношение светового потока к площади помещения (E=F/S).

Искусственная освещенность оператора ПК рассчитывается по формуле:

*E x K x η x T = L, (3)*

гдеE – коэффициент освещенности помещения, лк/м²;

K – коэффициент использования светового потока, принимается примерно равным 0,5;

η - коэффициент светопропускания светопрозрачных элементов в помещении (окна и т.п.), принимается в пределах 0,7-0,9;

T- коэффициент светового рассеивания стен белого цвета, имеющих светлую поверхность или покрытие, принимается равным 0,7 - 0,9;

L – необходимая освещенность поверхности рабочего места оператора ПК, лк.

Согласно стандартам, коэффициент освещенности для помещения размером 15 квадратных метров должен составлять не менее 300 лк/м². Для достижения этого значения и энергоэффективной работы светильника, мы можем рассчитать оптимальную световую мощность и эффективность светильника.

Подставляем значения в формулу:

*300 лк/м² × 15 м² = 4500 лк*

Для достижения необходимой световой мощности в 4500 лм при коэффициенте использования светового потока K=0,5, мы можем пересчитать эффективность светильника по формуле:

*P = (F × K) / η*

*η = (F × K) / P*

*η = (4500 лм × 0,5) / 50 Вт = 45*

Таким образом, чтобы обеспечить достаточный уровень освещенности помещения 15 кв. м с общим световым потоком 4500 лм, достаточно использовать светильники с эффективностью 45 лм/Вт и мощностью 50 Вт. Кроме того, для получения энергоэффективного решения, можно использовать светодиодные светильники, которые потребляют меньше энергии при той же световой мощности.

## 3.3. Техника безопасности при работе на ПК

Современная жизнь невозможна без использования компьютеров и интернета. Тем не менее, работа на ПК может представлять опасность для здоровья человека, если не соблюдать правила безопасности. Такие заболевания, как синдром карпального канала, синдром "сухого глаза", головные боли и шум в ушах, связаны с длительным пребыванием за компьютером. В связи с этим, необходимо соблюдать следующие правила техники безопасности при работе на ПК:

1. Регулярно делать перерывы. При работе за компьютером необходимо делать перерыв каждые 45-60 минут. Во время перерывов рекомендуется выполнять упражнения для глаз и рук.
2. Правильно настроить рабочее место. Рабочее место должно быть правильно настроено, чтобы минимизировать риск различных заболеваний. Клавиатура должна быть расположена на уровне локтя, а экран - на уровне глаз.
3. Использовать эргономическую мебель. Для работы за компьютером рекомендуется использовать эргономические кресла и столы, которые помогают поддерживать правильную позу.
4. Соблюдать правильный режим работы. Необходимо соблюдать правильный режим работы и сна, чтобы избежать усталости и стресса, которые могут привести к заболеваниям.
5. Использовать программы для защиты глаз. Существует множество программ, которые помогают защитить глаза при работе за компьютером. Они уменьшают яркость экрана, фильтруют синий свет и т.д.
6. Использовать антивирусное программное обеспечение. Антивирусное программное обеспечение помогает защитить компьютер от вирусов и злонамеренных программ, которые могут повредить систему или украсть личную информацию.
7. Избегать монотонной нагрузки. Повторяющиеся действия могут привести к снижению производительности и здоровью. Для этого рекомендуется использовать различные программы и ресурсы, менять виды деятельности, чтобы предотвратить нагрузку на конкретную группу мышц.
8. Использовать безопасные пароли. Для защиты личной информации и конфиденциальных данных следует использовать сложные пароли, которые не легко угадать или взломать. Рекомендуется использовать комбинации букв, цифр и символов.
9. Создавать резервные копии данных. Для сохранения важной информации необходимо регулярно создавать резервные копии данных, чтобы предотвратить потерю или повреждение важных файлов.
10. Обновлять программное обеспечение. Регулярное обновление программного обеспечения помогает устранять уязвимости и предотвращать атаки злонамеренных программ. Рекомендуется устанавливать обновления операционной системы, браузера и антивирусного ПО.
11. Использовать безопасную сеть. При работе на публичных сетях необходимо использовать VPN для защиты информации, передаваемой между компьютером и сервером или другим устройством.
12. Избегать использования нелицензионного программного обеспечения. Использование нелицензионного программного обеспечения может привести к нарушению законодательства и повышенному риску заражения вирусами и злонамеренными программами.

В целом, соблюдение правил техники безопасности при работе на ПК имеет важное значение для сохранения здоровья и безопасности человека. Рекомендуется следовать простым правилам, чтобы избежать проблем, связанных с длительным пребыванием за компьютером.

# Заключение

В рамках курсовой работы была разработана программа для онлайн курсов на языке C#. Целью разработки было создание приложения, которое позволяло бы эффективно управлять онлайн курсами, обеспечивая удобство использования для администраторов и пользователей.

В ходе разработки были выполнены следующие задачи:

Разработка программного обеспечения для онлайн курсов на языке C#;

Реализация функциональности для администраторов и пользователей;

Тестирование и отладка приложения.

Разработанное приложение позволяет администраторам создавать и управлять онлайн курсами, а также отслеживать прогресс пользователей

# Список используемых источников

1. Чуприна, С. В. "Разработка программного обеспечения: методические основы и технологии." – Москва: Академия, 2016.

2. Панкратов, О. В. "Управление проектами: методы и средства." – Санкт-Петербург: Питер, 2018.

3. Соммервилл, И. "Инженерия программного обеспечения." – Москва: Вильямс, 2016.

4. Макконнелл, С. "Совершенный код: Мастер-класс." – Москва: Вильямс, 2017.

5. Бейер, Т. и Хелмке, С. "Data Management for Researchers: Organize, Maintain and Share Your Data for Research Success." – London: SAGE Publications Ltd, 2016.

6. Смирнов, С. Н. "Опыт разработки и внедрения автоматизированной системы учета и управления предприятием." – Вестник МГТУ, 2019.

7. Иванов, П. А. "Методология проектирования информационных систем для малых предприятий." – Научный журнал СГТУ, 2020.

8. Кудрявцев, Е. В. "Анализ и моделирование бизнес-процессов для разработки программного обеспечения." – Журнал ИТ и Управление, 2018.

9. Petrova, T. V., & Smirnov, A. V. "Optimization of IT Infrastructure for Small Businesses: Case Study of a Hotels." – International Journal of Information Technology, 2019.

10. Stack Overflow – https://stackoverflow.com/ – Сообщество разработчиков программного обеспечения для обсуждения технических вопросов.

11. GitHub – https://github.com/ – Платформа для хостинга и совместной разработки проектов.

12. Habrahabr – https://habr.com/ru/ – Портал для публикаций и обсуждений в области IT и программирования.

**Приложение А**

**(Обязательное)**

# Техническое задание

**Приложение Б**

**(обязательное)**

# Фрагмент листинга программы