**Содержание**

[Введение 4](#_Toc137416877)

[1 Общая часть 6](#_Toc137416878)

[1.1Описание предметной области 6](#_Toc137416879)

[1.2Анализ существующей ситуации 8](#_Toc137416880)

[1.3Постановка задачи 9](#_Toc137416881)

[1.4Анализ существующих разработок и обоснование необходимости разработки……… 9](#_Toc137416882)

[2 Специальная часть 11](#_Toc137416883)

[2.1Выбор технологий и инструментальных средств 11](#_Toc137416884)

[2.1.1Выбор подхода к разработке 11](#_Toc137416885)

[2.1.2Выбор среды разработки и языка программирования 13](#_Toc137416886)

[2.2Разработка спецификаций 14](#_Toc137416887)

[2.2.1Разработка диаграмм вариантов использования 14](#_Toc137416888)

[2.2.2 Разработка диаграмм последовательностей системы 15](#_Toc137416889)

[2.2.3 Разработка диаграммы пакетов 15](#_Toc137416890)

[2.2.4 Разработка диаграмм классов 15](#_Toc137416891)

[2.2.5 Разработка диаграммы состояний 16](#_Toc137416892)

[2.2.6 Разработка диаграммы деятельности 18](#_Toc137416893)

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

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

[2.3Проектирование программного модуля 18](#_Toc137416896)

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

[2.3.2Проектирование пользовательского интерфейса 20](#_Toc137416898)

[2.4Реализация программного обеспечения на выбранном языке программирования и в выбранной среде разработки 22](#_Toc137416899)

[2.5Выбор стратегии тестирования, разработка тестов, тестирование и отладка программного обеспечения 25](#_Toc137416900)

[2.6Разработка эксплуатационной документации 30](#_Toc137416901)

[2.6.1Разработка руководства системного программиста 30](#_Toc137416902)

[2.6.2Разработка руководства пользователя 31](#_Toc137416903)

[3 Экономика производства 32](#_Toc137416904)

[3.1Расчет затрат на разработку и внедрение программного обеспечения 32](#_Toc137416905)

[3.2. Оценка эффективности внедрения программного обеспечения……………………. 32](#_Toc137416906)

[4 Безопасность жизнедеятельности 33](#_Toc137416907)

[4.1Требования к рабочим помещениям и рабочим местам оператора ПК 33](#_Toc137416908)

[4.2Расчет искусственной освещенности рабочего места оператора ПК 34](#_Toc137416909)

[4.3 Техника безопасности при работе на ПК 36](#_Toc137416910)

[Заключение 39](#_Toc137416911)

[Библиография 48](#_Toc137416912)

Приложение А (Обязательное) [Графическая часть 50](#_Toc137416913)

Приложение Б (Обязательное) [Техническое задание 51](#_Toc137416914)

Приложение В (Обязательное) [Руководство системного администратора 52](#_Toc137416915)

Приложение Г (Обязательное) [Руководство пользователя 53](#_Toc137416916)

Приложение Д (Обязательное) [Фрагмент листинга программы 54](#_Toc137416917)

# Введение

В наше время онлайн-обучение стало неотъемлемой частью образовательного процесса. С многочисленными возможностями, которые предоставляет интернет, студенты могут учиться из любой точки мира, что существенно упрощает доступ к образованию. Однако, несмотря на множество позитивных аспектов онлайн-обучения, многие студенты сталкиваются с проблемами, такими как отсутствие мотивации, недостаточная индивидуализация материалов и сложность оценки их успеваемости.

В связи с этим я решила сосредоточиться на теме своей дипломной работы: «Разработка системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости». Основная идея заключается в создании платформы, которая не только предлагает обучающие материалы, но и подстраивается под нужды и способности каждого студента. Это очень важно, ведь каждый учащийся – это уникальная личность, и стандартные подходы часто не работают для всех.

Адаптивный контент - это основа успешной системы обучения. Он позволяет оценивать уровень знаний студента и соответственно подбирать задания и материалы. Например, если один ученик быстро осваивает определенную тему, система может предоставить ему более сложные задания, в то время как другой студент, который нуждается в дополнительной помощи, получит материалы, которые помогут ему лучше усвоить материал. Такой подход делает процесс обучения более гибким и эффективным.

Аналитика успеваемости станет важным инструментом как для студентов, так и для преподавателей. С помощью аналитических данных преподаватели смогут видеть, где студенты испытывают трудности, и давать им дополнительную поддержку. Для студентов же будет полезно видеть свой прогресс и понимать, какие области они могут улучшить. В результате, такая система позволит не только повысить качество обучения, но и сделать его более регулярным и мотивирующим.

К тому же, в условиях большого потока информации, важно не только давать знания, но и уметь их структурировать. Адаптивная система будет давать возможность учащимся находить именно те виды материала, которые им необходимы, в зависимости от их интересов и предпочтений. Это создаст более комфортную и продуктивную образовательную среду.

Цель моей дипломной работы состоит в создании такой системы онлайн-обучения, которая будет максимально удобна и эффективна как для обучающихся, так и для преподавателей. Я планирую изучить существующие платформы, выявить их сильные и слабые стороны, а также определить требования, которые помогут мне реализовать свою систему. В процессе работы я разработаю алгоритм адаптации контента и внедрю инструменты для анализа успеваемости с последующим тестированием системы.

Данная система обучения даст возможность не только углубить знания студентов, но и поможет им быть более мотивированными в их обучении. Надеюсь, что она станет полезным инструментом в сфере образования и внесет свой вклад в развитие онлайн-обучения, улучшая качество образования для всех его участников.

# Общая часть

## Описание предметной области

Предметной областью данной дипломной работы является онлайн-обучение, а именно создание системы с адаптивным контентом и аналитикой успеваемости. Онлайн-обучение представляет собой образовательный процесс, который осуществляется с использованием цифровых технологий, позволяя студентам получать знания и навыки через интернет.

Современные образовательные платформы позволяют студентам учиться в любом месте и в любое время, что значительно расширяет доступ к образованию. Однако наличие большого объема информации и разнообразных форматов обучения иногда создает трудности в восприятии материала. Студенты могут теряться среди множества курсов и структур, что часто приводит к снижению их мотивации и эффективности обучения.

Адаптивное обучение становится решением этой проблемы. Системы с адаптивным контентом автоматически подстраивают учебные материалы под нужды каждого учащегося. Это значит, что в зависимости от уровня знаний, стиля обучения и предпочтений студента система сможет предлагать наиболее подходящие задания и ресурсы. Такой персонализированный подход помогает повысить вовлеченность студентов и улучшить усвоение знаний.

Ключевым элементом любой образовательной системы является возможность отслеживания успеваемости. Это особенно важно для преподавателей, которые могут видеть, как их студенты справляются с материалом, и на каких этапах им требуется дополнительная поддержка. Инструменты аналитики в системе помогут как учителям, так и учащимся лучше оценить прогресс и определить области, требующие улучшения.

Система онлайн-обучения, разработанная в рамках данной работы, будет полезна как для студентов, так и для учебных заведений. Она обеспечит удобный и доступный формат обучения, который будет отвечать современным требованиям. Кроме того, такая платформа может стать основой для более глубокой интеграции образования в цифровую среду, где образовательные процессы будут адаптироваться к меняющимся условиям и потребностям пользователей.

Таким образом, предметная область данной работы охватывает все аспекты онлайн-обучения - от формирования контента до анализа успеваемости, создавая условия для более эффективного и комфортного образовательного процесса. Данная система будет востребована в учебных заведениях, корпоративном обучении и других сферах, где важно предоставление качественного образовательного опыта.

## Анализ существующей ситуации

На сегодняшний день ситуация в сфере онлайн-обучения также характеризуется рядом проблем и вызовов, которые можно решить с помощью системы, основанной на адаптивном контенте и аналитике успеваемости. Рассмотрим некоторые из них:

1. Низкий уровень вовлеченности студентов. Во многих учебных учреждениях студенты могут испытывать затруднения с поддержанием интереса к материалу. Это может быть вызвано однообразием учебных материалов или недостаточной индивидуализацией обучения. В результате, многие ученики теряют мотивацию и не достигают поставленных учебных целей.
2. Однообразие контента. Традиционные методы обучения часто не учитывают уникальные потребности и стили обучения каждого студента. В результате того, что студенты получают одинаковые задания и материалы, это может приводить к тому, что одни учащиеся не успевают, а другие испытывают недостаток в сложности и вызовах.
3. Трудности в оценивании успеваемости. Преподавателям может быть сложно обнаружить, какие именно знания и навыки усваиваются плохо. Отсутствие систематовки и аналитики прогресса затрудняет выявление проблемных зон и формирование стратегии поддержки для студентов, что может негативно сказаться на качестве обучения.
4. Нехватка обратной связи. Многие онлайн-платформы не предоставляют студентам достаточной обратной связи о их успеваемости. Студенты могут не понимать, какие именно аспекты они должны улучшать, и, как следствие, не могут эффективно планировать свое время на обучение.
5. Сложности с доступом к образовательным ресурсам. Не все обучающиеся имеют равный доступ к качественным образовательным ресурсам. Многие студенты сталкиваются с проблемами в поиске подходящих материалов, особенно если они учатся самостоятельно вне традиционных учебных учреждений.

Разработка системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости может помочь решить эти проблемы. Она повысит уровень вовлеченности студентов, обеспечит индивидуализацию учебного материала, улучшит обратную связь и позволит преподавателям легче отслеживать прогресс учащихся. В конечном итоге, такая система сможет значительно улучшить качество образовательного процесса и создать более комфортные условия для обучения как студентов, так и преподавателей.

## Постановка задачи

Целью данной дипломной работы является разработка системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости студентов. Эта система должна предложить пользователям индивидуализированный подход к обучению, адаптируя материалы под особенности каждого учащегося и его прогресс.

Для достижения данной цели необходимо исследовать существующие методологии и подходы к адаптивному обучению, а также изучить, как анализ успеваемости может повлиять на образовательный процесс. Это включает изучение того, как данные о успеваемости студентов могут использоваться для улучшения содержания курсов и поддержки учащихся.

На основе полученного анализа требуется разработать архитектуру системы, обеспечивающую взаимодействие между компонентами обучающего процесса, такими как интерфейс пользователя и инструменты аналитики. Важно, чтобы система могла обрабатывать данные о студентах и их успеваемости, а также предлагать рекомендации для улучшения результатов.

Система должна хранить и обрабатывать данные в удобном формате для интеграции с другими образовательными инструментами и упрощения обмена данными. Процесс разработки должен основываться на современных методах проектирования программного обеспечения, включая применение принципов модульности и удобства использования, с акцентом на комфорт взаимодействия студентов и преподавателей.

Задача заключается в создании многофункциональной платформы онлайн-обучения, которая предоставит ученикам адаптивные образовательные материалы и инструменты для отслеживания успеваемости, повышая качество обучения и удовлетворенность пользователей.

1.4 Анализ существующих разработок и обоснование необходимости разработки

На сегодняшний день на рынке образовательных технологий представлено множество решений, предоставляющих возможности онлайн-обучения. Некоторое из них предлагает стандартные курсы без учета индивидуальных потребностей учащихся, в то время как другие системы предоставляют более адаптивные подходы, учитывающие стиль и скорость обучения каждого студента.

Например, платформа Coursera предлагает широкий ассортимент курсов, однако она не всегда адаптируется к индивидуальным требованиям учащихся, что может снизить эффективность обучения. В отличие от этого, система Adaptive Learning, такая как Knewton, использует алгоритмы для анализа успеваемости учащихся и на основе полученных данных предлагает персонализированные учебные материалы и рекомендации.

Существуют также платформы, такие как Edmodo и Google Classroom, которые обеспечивают взаимодействие между учениками и преподавателями, однако их функциональность в плане адаптации контента к индивидуальным потребностям остаётся ограниченной. Тем не менее, они предоставляют полезные инструменты для управления учебным процессом и общения.

Необходимо отметить, что многие используемые решения акцентируют внимание на одной из сторон обучения, например, на управлении курсами или взаимодействии, но не обладают интегрированным подходом, который бы объединял адаптивные возможности с аналитикой успеваемости студентов. Это создаёт пробел в области онлайн-обучения, где есть необходимость в разработке многофункциональной системы, способной учитывать особенности каждого учащегося и предоставлять им адаптивный контент, а также аналитические инструменты для отслеживания успехов в обучении.

Таким образом, существует очевидная необходимость в разработке системы онлайн-обучения, которая объединит адаптивный контент и аналитику успеваемости, что позволит повысить качество образовательного процесса и удовлетворённость пользователей. Это решение сможет улучшить эффективность обучения, предоставляя ученикам необходимые ресурсы и поддержку для достижения их целей.

1. Специальная часть

2.1 Выбор технологий и инструментальных средств

* + 1. Выбор подхода к разработке

Для дипломной работы по теме "Разработка системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости" более подходящим вариантом является применение объектно-ориентированного подхода. Это связано с тем, что такая система требует четкой структуры и разделения ответственности между различными компонентами, такими как модули адаптивного контента, системы аналитики и интерфейсы пользователей.

Объектно-ориентированный подход позволяет создать классы и объекты для моделирования компонентов системы, таких как пользователи, курсы и результаты их обучения. Каждому из этих компонентов можно присвоить уникальные свойства и методы, обеспечивающие возможность взаимодействия между ними. Такой подход способствует лучшей структурированности кода и упрощает его поддержку, что особенно важно при необходимости внесения изменений или добавления новых функций в систему.

В отличие от этого, применение структурного подхода может привести к усложнению управления взаимодействиями между компонентами, а также затруднить возможность использования кода в других системах или проектах. Структурный подход может также стать причиной дублирования кода и конфликтов, что усложнит дальнейшую поддержку и разработку системы.

Объектно-ориентированная модель позволяет быстро адаптироваться к изменяющимся требованиям пользователей, что особенно актуально в сфере онлайн-образования. Гибкость этой модели также упрощает процесс тестирования и отладки, что в свою очередь оптимизирует обслуживание системы.

Таким образом, для разработки системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости наиболее оправданным является объектно-ориентированный подход, который обеспечивает необходимую гибкость, масштабируемость и надежность системы в целом.

* + 1. Выбор среды разработки и языка программирования

Для разработки системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости я выбрала язык программирования C# и интегрированную среду разработки Visual Studio 2022.

C# является объектно-ориентированным языком программирования, разработанным Microsoft, который предлагает мощные возможности для создания как настольных, так и веб-приложений. Благодаря сильной типизации, поддержке асинхронного программирования и обширной стандартной библиотеке, C# идеально подходит для разработки сложных систем, требующих высокой производительности и надежности. Язык также хорошо поддерживает концепции модульности, что облегчит масштабирование моего проекта в будущем.

Visual Studio 2022 - это современная интегрированная среда разработки (IDE), предлагающая множество инструментов, которые значительно упрощают процесс создания программного обеспечения. К числу её ключевых преимуществ можно отнести:

* 1. Мощный отладчик: Visual Studio предоставляет удобные средства для пошаговой отладки, что позволяет быстро выявлять и исправлять ошибки в коде, а также анализировать поведение приложения на разных этапах.
  2. Отличные инструменты интеграции: IDE поддерживает работу с Git и другими системами контроля версий, что позволяет удобно управлять проектом совместно с командой и отслеживать изменения в коде.
  3. Широкий выбор библиотек и фреймворков: C# имеет доступ к мощным фреймворкам, таким как ASP.NET, которые облегчают разработку веб-приложений и обеспечивают поддержку адаптивного контента и аналитики.
  4. Удобные инструменты для разработки интерфейсов: Visual Studio включает в себя дизайнеры для создания пользовательских интерфейсов, что позволяет быстро и эффективно разрабатывать привлекательные и функциональные интерфейсы для системы онлайн-обучения.

Таким образом, использование C# и Visual Studio 2022 для разработки системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости обеспечит надежную и высокопроизводительную платформу для достижения поставленных целей.

2.2 Разработка спецификаций

2.2.1 Разработка диаграмм вариантов использования

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

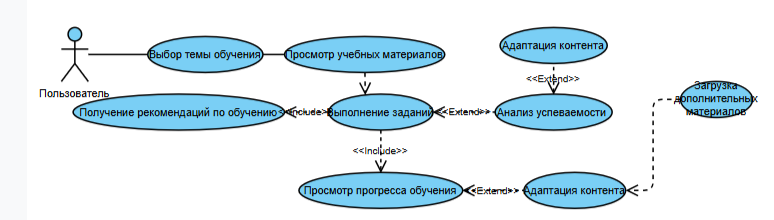


Рисунок 1 – Диаграмма вариантов использования

На диаграмме представлены следующие варианты использования:

"Выбор темы обучения": Пользователь выбирает тему, которую хочет изучать. Это начальная точка взаимодействия с системой.

"Просмотр учебных материалов": Пользователь просматривает учебные материалы, предоставленные системой по выбранной теме.

"Выполнение заданий": Пользователь выполняет задания, связанные с изучаемой темой, для проверки усвоения материала.

"Просмотр прогресса обучения": Пользователь отслеживает свой прогресс в изучении выбранной темы. Этот вариант использования всегда включается в "Выполнение заданий".

"Получение рекомендаций по обучению": Система предоставляет пользователю персонализированные рекомендации по дальнейшему обучению, основанные на его прогрессе и результатах выполнения заданий. Этот вариант использования всегда включается в "Выполнение заданий".

"Анализ успеваемости (Преподаватель)": Преподаватель анализирует успеваемость учащихся по собранным системой данным. Этот вариант использования может расширять "Выполнение заданий", предоставляя дополнительную информацию для анализа.

"Адаптация контента": Система адаптирует контент в зависимости от результатов анализа успеваемости и прогресса обучения. Этот вариант использования может расширять "Анализ успеваемости (Преподаватель)" и "Просмотр прогресса обучения", изменяя сложность и тип предоставляемых материалов.

"Загрузка дополнительных материалов (Преподаватель)": Преподаватель загружает в систему дополнительные учебные материалы. Этот вариант использования может расширять "Адаптация контента", предоставляя новые материалы для обучения.

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.

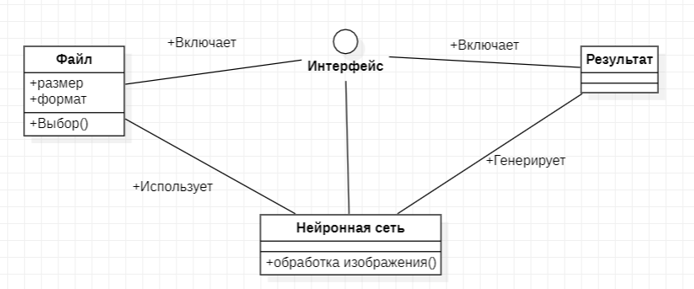


Рисунок 3 – Диаграмма классов

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

Диаграмма состоит из трех состояний:

1. Ожидание взаимодействия: начальное состояние, в котором система ожидает действия пользователя – выбора темы обучения или продолжения работы с текущей темой.
2. Обучение: состояние, в котором пользователь взаимодействует с учебным контентом – просматривает материалы, выполняет задания, получает обратную связь. В этом состоянии система также непрерывно анализирует прогресс пользователя.
3. Адаптация: состояние, в котором система, основываясь на собранных данных об обучении пользователя, адаптирует контент, подбирая оптимальную сложность и тип материалов для следующего этапа обучения.

Переход между состояниями происходит следующим образом:

* Пользователь совершает действие (выбирает тему, начинает/продолжает обучение), и система переходит из состояния "Ожидание взаимодействия" в состояние "Обучение".
* По мере обучения пользователя и накопления данных о его прогрессе, система периодически переходит в состояние "Адаптация".
* После адаптации контента, система возвращается в состояние "Обучение" с обновленными материалами. Если пользователь завершает обучение по выбранной теме, система переходит в состояние "Ожидание взаимодействия".

Данная диаграмма позволяет лучше понимать работу системы онлайн-обучения и последовательность её действий в зависимости от взаимодействия с пользователем и анализа его прогресса. Она показывает циклический характер процесса обучения с постоянной адаптацией контента.

2.2.6 Разработка диаграммы деятельности

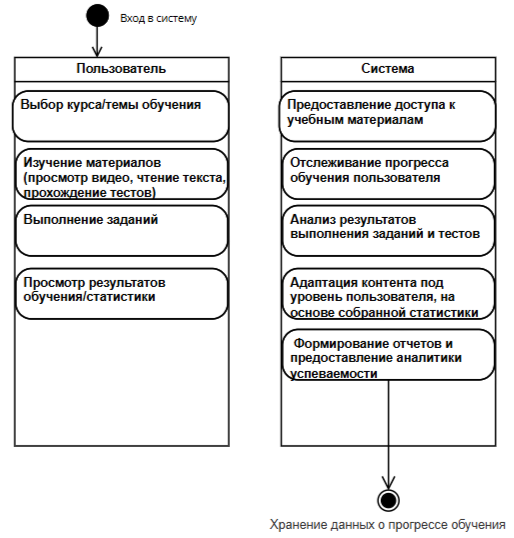
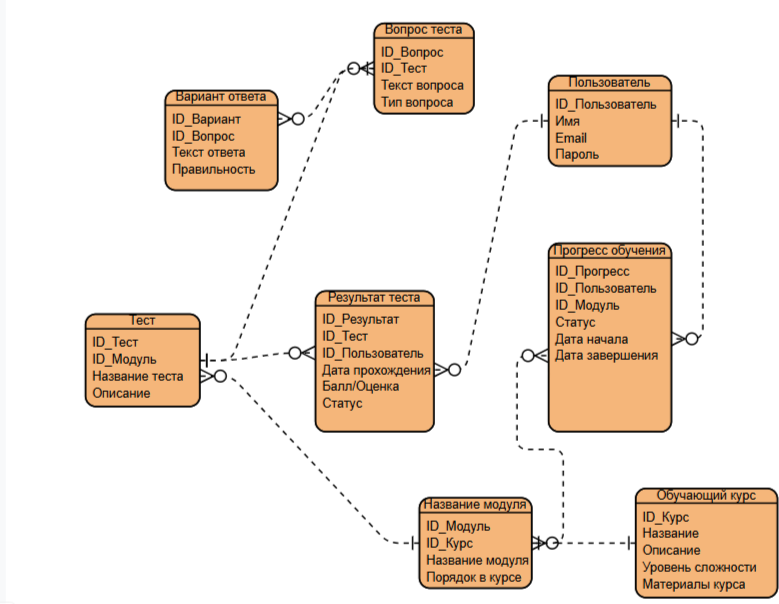


Рисунок 5- Диаграмма деятельности

Данная диаграмма деятельности описывает последовательность действий в системе онлайн-обучения с адаптивным контентом и аналитикой успеваемости. Пользователь входит в систему, выбирает интересующий курс или тему, изучает предоставленные материалы, выполняет задания и просматривает результаты обучения. Система, в свою очередь, предоставляет доступ к учебным материалам, отслеживает прогресс пользователя, анализирует результаты его деятельности, адаптирует контент под уровень пользователя на основе собранной статистики и предоставляет аналитику успеваемости. Этот цикл взаимодействия позволяет системе персонализировать обучение и повысить его эффективность.

2.2.8 Разработка инфологической модели базы данных

 Рисунок 6- Инфологическая модель базы данных.

2.2.9 Разработка физической модели базы данных

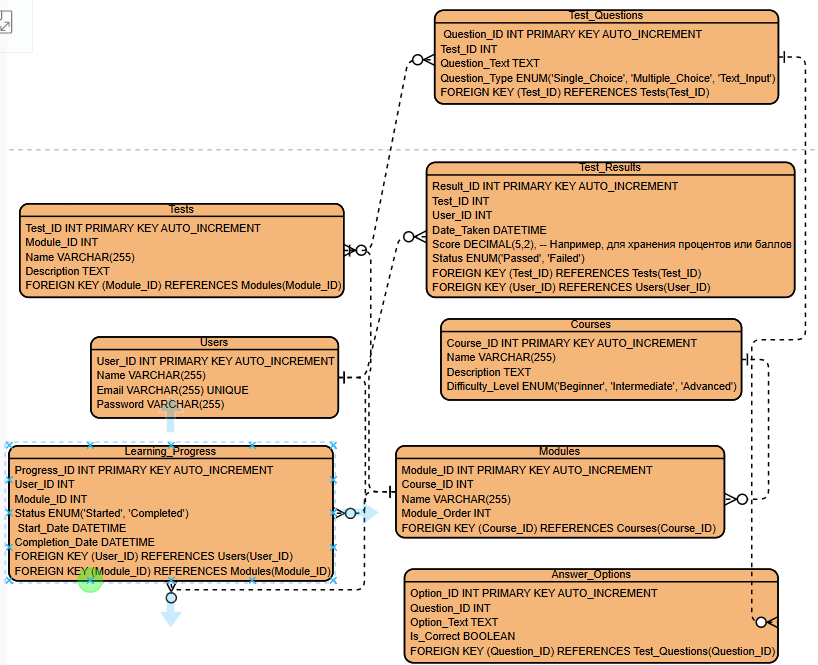


Рисунок 6- Инфологическая модель базы данных.

2.3 Проектирование программного модуля

2.3.1 Разработка алгоритмов реализации основных функций программного обеспечения

⦁ Авторизация: Запуск ПО. Ввод данных пользователем. Проверка данных в БД. Открытие Dashboard с учетом профиля пользователя и рекомендациями.

⦁ Регистрация: Запуск ПО. Ввод данных пользователем. Валидация данных. Сохранение данных в БД. Создание профиля с начальными настройками и предложением пройти первичное тестирование.

⦁ Список курсов: Запуск Dashboard. Запрос к БД с учетом профиля пользователя. Отображение отсортированного списка курсов с информацией о сложности и прогрессе.

⦁ Создание курса: Запуск формы. Ввод данных курса. Настройка адаптивности контента. Сохранение в БД. Обновление списка курсов.

⦁ Добавление материалов: Выбор курса. Добавление материалов. Настройка адаптивности материалов. Сохранение материалов. Обновление списка материалов.

⦁ Прохождение курса: Выбор курса студентом. Отображение адаптированного контента. Отслеживание прогресса и результатов. Адаптация сложности в зависимости от успеваемости.

⦁ Аналитика успеваемости: Сбор данных о прогрессе. Анализ данных и представление в виде отчетов. Отображение прогресса и рекомендаций для студентов. Предоставление статистики преподавателям для корректировки курса.

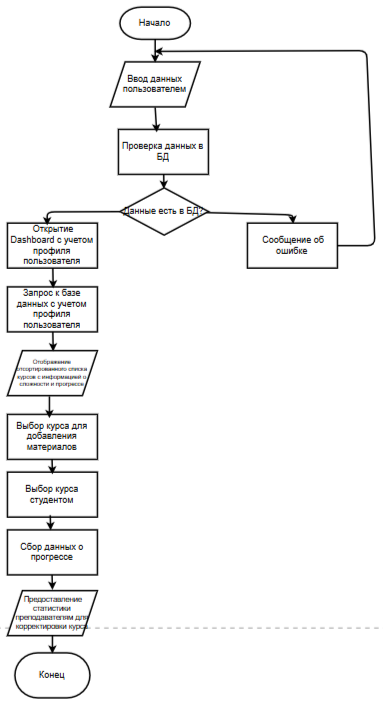


Рисунок 7- Алгоритм получения результатов

### 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".

2. Добавление нужных файлов.

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

3. Написание кода.

Код программы, написанный на языке C#, реализует основные функции системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости. В коде используются методы для обработки данных о курсах и материалах, методы для управления пользователями, а также алгоритмы для адаптации контента в зависимости от уровня знаний учащегося и анализа успеваемости.

Некоторые из реализованных функций могут включать:

// Загрузка курса пользователем для дальнейшего обучения.

private void LoadCourse(string courseId)

{

Course course = CourseManager.GetCourseById(courseId);

if (course != null)

{

// отображение материалов курса

}

else

{

MessageBox.Show("Курс не найден.");

}

}

// Подсчет прогресса пользователя в курсе

private double CalculateUserProgress(User user, Course course)

{

// логика подсчета прогресса на основе результатов тестов и завершенных материалов

return userProgressPercentage;

}

// Адаптация уровня материалов в зависимости от знаний пользователя

private Material GetAdaptedMaterial(User user)

{

// получение материалов, соответствующих уровню знаний пользователя

return MaterialManager.GetMaterialsForKnowledgeLevel(user.KnowledgeLevel);

}

4. Тестирование и отладка

После написания кода программа должна пройти полное тестирование и отладку. Для этого можно использовать заранее подготовленные тестовые данные или создать собственные наборы, чтобы проверить правильность работы системы. Каждый модуль, отвечающий за определенные функции, должен быть протестирован отдельно, а затем проведены интеграционные тесты для проверки взаимодействия между модулями.

5. Компиляция и сборка.

Если все тесты проходят успешно и система работает без ошибок, можно скомпилировать и собрать программу. В результате на выходе получится исполняемый файл с расширением ".exe", который сможет запустить пользователь на своем компьютере. Процесс компиляции включает создание всех необходимых сборок и библиотек, а также проверку выявленных ошибок.

6. 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 нейросети. Он может быть использован для тестирования нейросетей, написанных на языке C#.

Таким образом, для тестирования приложения, использующего нейросеть, наиболее подходящим инструментом будет NUnit. Он предоставляет полный набор функций для тестирования приложений на языке C# и позволяет проводить проверку отдельных компонентов системы, включая использование нейросети.

**2.6 Разработка эксплуатационной документации**

**2.6.1 Разработка руководства системного программиста**

По ГОСТ 19.503-79 руководство системного программиста должно содержать всю информацию, необходимую для установки программного обеспечения, его настройки и проверки работоспособности. Кроме того, как указывалось выше, в него часто включают и описание необходимого обслуживания, которое раньше приводилось в руководстве оператора (ГОСТ 19.505-79) и/или руководстве по техническому обслуживанию (ГОСТ 19.508-79). В настоящее время данную схему используют для составления руководства системному администратору.

Руководство системного программиста должно содержать следующие разделы:

* 1. общие сведения о программном продукте;
  2. структура;
  3. настройка;
  4. проверка;
  5. дополнительные возможности;
  6. сообщения системному программисту.

Руководство системного программиста представлено в приложении В.

**2.6.2 Разработка руководства пользователя**

Руководство пользователя согласно ГОСТ РД 50-34.698-90- документ, в котором описывается, как пользоваться продуктом, включая установку на компьютер и работу с ним. В него также может быть включена информация об ограничениях на использование продукта и обратная связь с поддержкой.

Руководство пользователя представлено в приложении Г.

1. **Экономика производства**

**3.1 Расчет затрат на разработку и внедрение программного обеспечения**

* Апгрейд серверов: 300 000 рублей (основано на данных поставщиков оборудования или анализе рыночных цен)
* Обновление сетевого оборудования: 150 000 рублей (основано на <https://rdtex.ru/services/>)

Общая сумма инфраструктурных затрат составит 450 000 рублей (31% от общей суммы).

Если обучение 20 сотрудников обойдется в 50 000 рублей, то эта статья затрат составит 50 000 рублей (усредненные данные рынка, 3% от общей суммы).

* Услуги консультантов: 200 000 рублей (основано на https://nv-ot.com/).

К прочим затратам можно отнести расходы на маркетинг внедрения системы внутри компании, а также потенциальные расходы на техническую поддержку.

* Прочие расходы: 50 000 рублей (основано на https://direct.yandex.ru/), (14% от общей суммы).

Таблица 2. Общие начальные затраты на внедрение системы онлайн обучения

|  |  |  |
| --- | --- | --- |
| Наименование затрат | Сумма (руб.) | % от общей суммы |
| Приобретение ПО | 700000 | 48% |
| Инфраструктурные затраты | 450000 | 31% |
| Обучение персонала | 50000 | 3% |
| Консультационные услуги | 200000 | 14% |
| Прочие затраты | 50000 | 3% |
| Итого | 1450000 | 100% |

Таким образом, мы видим, что общие начальные затраты на внедрение системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости в АО «Апатит» составляют 1 450 000 рублей.

* 1. **Оценка эффективности внедрения программного обеспечения**

Прогнозируемые доходы от реализации системы составляют 19 100 000 рублей в год. Эти доходы включают:

* Увеличение производительности труда: 18 000 000 рублей.
* Снижение затрат на обучение: 600 000 рублей.
* Снижение затрат на устранение ошибок: 500 000 рублей.

**4 Безопасность жизнедеятельности**

**4.1 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.

**4.2 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.

**4.3. 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 заключение дипломной работы по разработке системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости можно отметить, что данная система представляет собой эффективное и современное решение в области образования. Разработка системы обеспечивает персонализированный подход к обучению, что позволяет значимо повысить заинтересованность студентов и их уровень усвоения материала.

Интеграция адаптивного контента, который изменяется в зависимости от индивидуальных успехов учащегося, дает возможность улучшить образовательный процесс и добиться более высоких результатов. Внедрение аналитики успеваемости позволяет преподавателям своевременно корректировать учебные программы и подходы, что способствует не только повышению качества обучения, но и поддержанию мотивации студентов.

Для достижения поставленной цели выполнены следующие задачи:

- проведено описание предметной области;

- проведен анализ существующих онлайн- обучающих платформ;

- выполнено обоснование необходимости разработки программного обеспечения;

- выполнено обоснование выбора операционной системы;

- выполнено обоснование выбора языка программирования и инструментального программного обеспечения;

- разработана спецификация проектируемой системы;

- выполнено проектирование программного обеспечения;

- реализовано программное обеспечение;

- протестировано программное обеспечение;

- составлено руководство пользователя;

- составлено руководство для администраторов системы.

Таким образом, разработка системы онлайн-обучения с адаптивным контентом и аналитикой успеваемости представляет собой актуальную и перспективную задачу, которая может значительно улучшить процесс образования, сделать его более доступным и эффективным. Данная система имеет потенциал для применения в различных учебных заведениях, корпоративных университетах, а также в контексте самостоятельного обучения, что подчеркивает её универсальность и значимость в нынешних условиях.

**Библиография**

1. Джозеф Албахари, Бен Албахари. C# 7.0 в деталях: справочник (O'Reilly Media, 2017).  
   Этот справочник предоставляет глубокое понимание языка C# и его возможностей.
2. Джон Скиит. C# в деталях, 4-е издание (Manning Publications, 2020).  
   Книга предлагает подробное описание языка C# и его применения.
3. Иан Гудфеллоу, Джошуа Бенджио, Аарон Курвилль. Глубокое обучение (MIT Press, 2016).  
   Для создания адаптивного контента может быть полезно применение методов глубокого обучения.
4. Кристофер Бишоп. Распознавание образов и машинное обучение (Springer, 2006).  
   Эта книга поможет понять принципы машинного обучения, которые можно использовать для анализа успеваемости.
5. Майкл Нильсен. Нейронные сети и глубокое обучение: учебник (diveintodeeplearning.org, 2018).  
   Учебник по нейронным сетям, который может быть полезен для разработки адаптивного контента.
6. Руководство по C# (Microsoft Learn) <https://learn.microsoft.com/ru-ru/dotnet/csharp/> ( дата обращения 21.03.25)
7. .NET для студентов (Microsoft Learn) <https://dotnet.microsoft.com/ru-ru/learntocode> (дата обращения 11.03.25)
8. Применение шаблонов проектирования для создания новых классов на примере шаблона «Стратегия» (Быцан А.И., Ивакин А.С., Зиновьева В.В.) <https://vsuet.ru/images/struct-dept/npp/sno/confer/stud/2024/mat_stud_2024.pdf> (дата обращения 16.03.25)
9. Искусственный интеллект в науке и образовании: плюсы, минусы, прогнозы (Васечкин В.М., Саввина Е.А.) <https://vsuet.ru/images/struct-dept/npp/sno/confer/stud/2024/mat_stud_2024.pdf> (дата обращения 01.04.25)
10. Цифровая трансформация агропромышленного комплекса (Холин Д.А., Филатова М.В.) <https://vsuet.ru/images/struct-dept/npp/sno/confer/stud/2024/mat_stud_2024.pdf> (дата обращения 12.03.25)
11. Джеффри Рихтер. CLR на C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#
12. TutorialsPoint: Базовый учебник по C # и Расширенный учебник по C # <https://www.tutorialspoint.com/csharp/index.htm> (дата обращения 01.03.25)

Приложение А

(обязательное)

Графическая часть

Приложение Б

(Обязательное)

Техническое задание

Приложение В

(Обязательное)

Руководство системного администратора

Приложение Г

(обязательное)

Руководство пользователя

Приложение Д

(обязательное)

Фрагмент листинга программы