|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ИВАНОВСКОЙ ОБЛАСТИ**  Областное государственное бюджетное профессиональное образовательное учреждение  «Ивановский промышленно-экономический колледж» | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | |  | |
|  | | | | |  | |
|  | | | | |  | |
|  | | | | |  | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| **КУРСОВОЙ ПРОЕКТ** | | | | | | |
|  | | | | | | |
| **Разработка системы контроля версий** | | | | | | |
|  | | | | | | |
| **ИВПЭК. 09.02.07.17** | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | **Специальность:** | | | **09.02.07 Информационные системы и программирование,**  **базовая подготовка** | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| Руководитель курсового проекта | | | |  | | В.И. Кукушкин |
| Выполнил обучающийся группы 307а | | | |  | | Д.Д. Михайлов |
|  | | | |  | |  |
|  | | | | | | |
|  | | | | | | |
| Курсовой проект выполнен и защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | | | |
|  | | | | | | |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | | | | |
|  |  | | | | | |
|  | | |  | | |  |
|  | | |  | | |  |
|  | Иваново 2024 | | | | |  |

Содержание

[Введение 3](#_Toc193196093)

[1 Концептуальное проектирование 5](#_Toc193196094)

[2 Техно–рабочий проект 7](#_Toc193196095)

# Введение

В современном мире существует такая проблема: неорганизованная работа над проектами, особенно когда над ними трудятся несколько человек одновременно. Представьте себе ситуацию: программисты вносят изменения в один и тот же файл, дизайнеры перерабатывают макеты, а копирайтеры редактируют текст. Как отследить, кто что изменил, вернуться к предыдущей версии, если что-то пошло не так, и не запутаться в бесчисленных копиях документов? Для решения этой проблемы и существуют системы контроля версий, и, как их воплощение – приложения для контроля версий. Без них современная разработка программного обеспечения и многих других видов совместной работы была бы просто немыслима. Эти приложения стали неотъемлемой частью рабочего процесса, обеспечивая прозрачность, эффективность и надежность при создании любого проекта.

Актуальность темы разработки системы контроля версий в современном мире трудно переоценить. В условиях постоянно растущей сложности программных проектов и увеличения числа разработчиков, работающих над ними, эффективное управление изменениями становится критически важным. Без надежной системы контроля версий, совместная работа над проектами быстро превращается в хаос, приводя к потере времени, ошибкам и сложностям в отслеживании изменений.

Системы контроля версий позволяют не только отслеживать историю изменений, но и эффективно управлять ветвлением кода, объединять результаты работы разных разработчиков и откатываться к предыдущим версиям в случае возникновения проблем. Однако, несмотря на распространенность существующих решений, потребность в новых, более гибких, удобных и производительных системах контроля версий сохраняется.

Существуют различные сцена того, развитие облачных технологий и тенденция к совместной работе над проектами из разных точек мира, делают вопрос эффективного контроля версий еще более важным. Поэтому разработка современной, надежной и удобной системы контроля версий не только не теряет актуальности, но и становится все более значимой для успешной реализации любых проектов в сфере IT и за ее пределами.

Цель проекта — разработка надежной и удобной системы контроля версий, которая позволит эффективно управлять изменениями в проектах любого масштаба и сложности, отслеживание истории изменений, простое разрешение конфликтов и возможность возврата к предыдущим версиям. Система должна обладать интуитивно понятным интерфейсом для пользователей с различным уровнем технической подготовки.

В рамках проекта по разработке системы контроля версий, нам предстоит реализовать базовые функции отслеживания изменений в файлах и каталогах, что позволит фиксировать (коммитить) эти изменения, сохраняя полную историю проекта. Также необходима функциональность отката к предыдущим версиям, что обеспечит безопасность и позволит вернуться к рабочему состоянию в случае необходимости. Для удобства пользователей, к каждому коммиту можно будет добавлять комментарии, объясняющие суть внесенных изменений.

Кроме того, мы разработаем удобный и интуитивно понятный пользовательский интерфейс для управления репозиторием. Важным аспектом будет наглядное отображение истории изменений, чтобы пользователям было легко ориентироваться в процессе разработки. Система будет доступна через командную строку (CLI) для опытных пользователей и через графический интерфейс (GUI) для более комфортной работы. Мы также позаботимся о кроссплатформенности приложения, чтобы система могла работать на различных операционных системах.

# 1 Концептуальное проектирование

1.1 Теоретический вопрос

Разработка систем контроля версий (СКВ) крайне актуальна в современном мире из-за роста сложности проектов и распределённой работы команд. СКВ обеспечивают координацию изменений, сохранность данных, отслеживание истории разработки и упрощают совместную работу, что критически важно для успешного создания программного обеспечения и цифровых продуктов.

Задача разработки системы контроля версий (СКВ) с максимально упрощенным интерфейсом, ориентированным на удобство пользователя, особенно начинающих программистов, заключается в том, чтобы сделать процесс структурированной разработки программного обеспечения доступным для каждого.

Это означает, что система должна быть спроектирована таким образом, чтобы минимизировать порог вхождения для новичков, не требуя глубоких технических знаний или опыта работы с традиционными СКВ. Интерфейс должен быть интуитивно понятным, наглядным и предлагать простые, логичные рабочие процессы.

Основная цель – дать возможность даже тем, кто только начинает свой путь в программировании, эффективно управлять своими проектами, отслеживать изменения, экспериментировать с кодом и сотрудничать с другими разработчиками, не тратя время и усилия на освоение сложных инструментов.

В результате, такая СКВ должна стать мощным инструментом для обучения, развития и повышения продуктивности как начинающих, так и опытных программистов, способствуя созданию качественного и хорошо структурированного программного обеспечения.

1.2 Техническое задание

Цель: Создать простую СКВ для начинающих программистов.

Функции:

* Создание локального репозитория.
* Добавление/удаление/статус файлов.
* Коммит с комментарием.
* История изменений (автор, дата, комментарий).
* Восстановление версий.

3. Интерфейс:

* Простой и понятный.
* CLI (обязательно), GUI (опционально).

4. Требования:

* Быстрая работа.
* Надежная сохранность данных.

5. Этапы:

* Проектирование.
* Разработка ядра (функции СКВ).
* Разработка UI.
* Тестирование.
* Документация.

6. Критерии приемки:

* Соответствие функциям и требованиям.
* Успешные тесты.
* Понятная документация.
* Простота использования.

# 

# 2 Техно–рабочий проект

2.1 Моделирование с использованием UML-диаграмм