Приложение В

|  |
| --- |
| **Автономное учреждение  профессионального образования**  **Ханты-Мансийского автономного округа – Югры**  **«СУРГУТСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**  **(АУ «Сургутский политехнический колледж»)** |
|  |
| СТРУКТУРНОЕ ПОДРАЗДЕЛЕНИЕ - 4  (Энергетическое отделение) |

**Отчет по практической работе №\_3\_**

**по дисциплине**

**МДК 01.02 “Инструментальные средства разработки программного обеспечения”.**

Выполнил:

студент группы 319

Тимергалиев Тимур Ильдарович

«\_\_18\_\_» \_\_\_\_11\_\_\_\_\_\_\_\_\_ 2024г.

Сургут – 2024 г.

# Цель работы

Цель данной работы – Изучение и применение системы контроля версий на практике, включая настройку типов импортируемых файлов, путей, фильтров и других параметров для эффективного управления репозиториями.

# Практическая часть

# 1. Типы импортируемых файлов

Система контроля версий позволяет управлять различными типами файлов, которые можно импортировать в репозиторий. Для корректной работы необходимо определить список поддерживаемых форматов файлов, например, .txt, .docx, .jpg, .png и др. Правильная настройка типов файлов предотвращает загрузку неподдерживаемых или вредоносных файлов.

Задание:

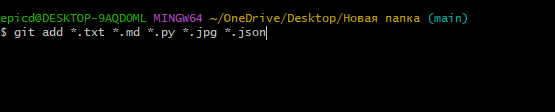
1. Определите список типов файлов, которые будут допустимы для вашего проекта. Объясните свой выбор.

В моем проекте я буду использовать следующие типы файлов:

* .txt — для текстовых документов.
* .md — для файлов Markdown.
* .py — для исходного кода на Python.
* .jpg, .png — для изображений.
* .json — для конфигурационных файлов.

Выбор этих типов файлов обусловлен тем, что они наиболее часто используются в моем проекте и не представляют угрозы для безопасности.

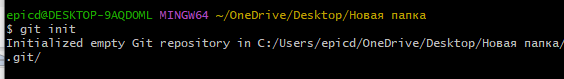
2. Настройте систему контроля версий для работы с указанными типами файлов.



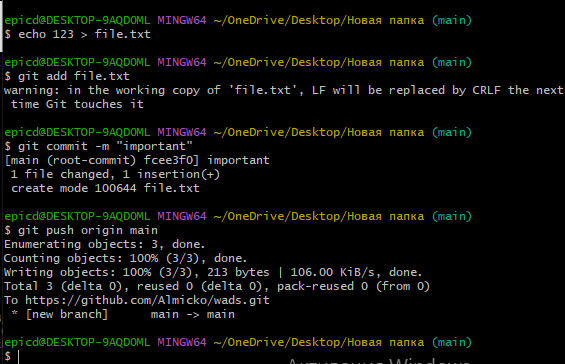
# 2. Настройка путей

Задание:

1. Настройте систему контроля версий для вашего проекта, указав абсолютные и относительные пути к основным директориям.



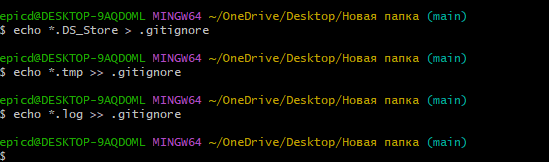
2. Проверьте корректность настройки путем создания и сохранения тестового файла.



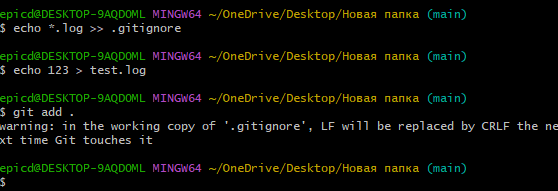
# 3. Фильтрация импортируемых файлов

Задание:

1. Создайте файл .gitignore для вашего проекта, включив в него временные файлы и директории, которые не должны быть добавлены в репозиторий.

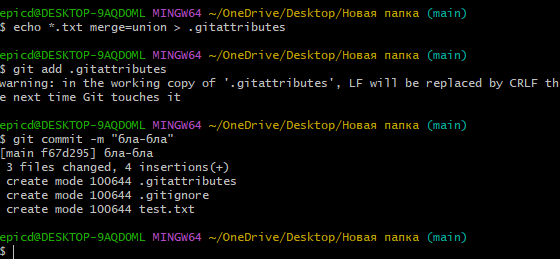


2. Проверьте работу .gitignore, добавив тестовые файлы и убедившись, что они исключены.

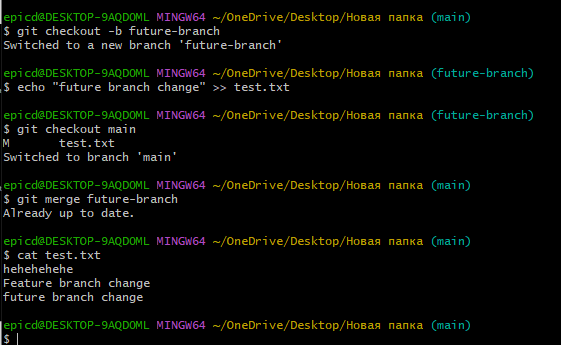


# 4. Другие параметры импорта

1. Настройте систему контроля версий так, чтобы при возникновении конфликтов использовались правила слияния.



2. Проверьте работу системы при попытке одновременного изменения одного и того же файла с разных версий.



# Заключение

Я изучил и применил системы контроля версий на практике, включая настройку типов импортируемых файлов, путей, фильтров и других параметров для эффективного управления репозиториями.