**Отчет по самостоятельной работе №**3

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

Выполнил: студент

Группы 319

Суворова Анастасия Викторовна

Дата\_\_.\_\_.\_\_\_\_

**Цель работы**

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

**Основная структура задания**

1. Типы импортируемых файлов
2. .docx, .pdf, .py, .txt - типы файлов, которые будут допустимы для моего проекта. Эти форматы обеспечивают поддержку самых распространённых типов информации, которые могут потребоваться в проектах, исключены потенциально вредоносные форматы, такие как исполняемые файлы которые могут вызвать уязвимости.
3. Для настройки системы контроля - Git для работы с указанными типами файлов, можно использовать .gitignore.

Исключаю временные файлы

* .tmp
* .bak
* .log

Исключаю скомпилированные файлы

* .class
* .pyc
* .o
* .exe

Игнорирую системные и резервные файлы

.DS\_Store

Thumbs.db

Исключаем все файлы, кроме поддерживаемых

1. Настройка путей
2. [repository]

* Абсолютные пути:

path=/home/user/project

backup\_path=/home/user/project\_backup

* Относительные пути

src\_path=src

docs\_path=docs

assets\_path=assets

tests\_path=tests

configs\_path=configs

2. Корректность настройки

319 -p /home/user/project/src

319 -p /home/user/project/docs

319 -p /home/user/project/assets

319 -p /home/user/project/tests

319 -p /home/user/project/configs

cd /home/user/project

git init

echo "print('Hello, World!')" > src/test\_script.py

git add src/test\_script.py

git commit -m "Добавлен тестовый файл test\_script.py"

git status

3) Фильтрация импортируемых файлов

1. cd /home/user/project

touch .gitignore

nano .gitignore

touch example.log

touch example.tmp

319 node\_modules

319 dist

git status

git add .

git status

4) Другие параметры импорта

git config --global merge.conflictstyle diff3

319 test-repo

cd test-repo

git init

echo "Строка 1" > test.txt

git add test.txt

git commit -m "Добавлен файл test.txt"

git checkout -b branch-a

echo "Изменение из branch-а" >> test.txt

git commit -am "Изменение в branch-а"

git checkout master

echo "Изменение из master" >> test.txt

git commit -am "Изменение в master"

git merge branch-a

**Выводы работы**

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