

Отчёт по лабораторной работе

Тема: Организация рабочего окружения и работа с Git. Стратегии ветвления и автоматизация контроля качества.

Сведения о студенте

Дата: 2025-10-18

Семестр: 3

Группа: ПИН-б-о-24-1

Дисциплина: Технологии программирования

Студент: Макаров Роман Дмитриевич

Цель работы

Приобрести практические навыки настройки среды разработки в Ubuntu, освоить базовые операции системы контроля версий Git, изучить стратегии ветвления Git Flow/GitHub Flow и настроить автоматизированный контроль качества с помощью pre-commit хуков.

Ход работы

1. Предварительная настройка (SSH-аутентификация)

- Сгенерирована пара SSH-ключей с использованием алгоритма Ed25519.
- Публичный ключ добавлен в аккаунт GitHub.
- Проверено подключение к GitHub через SSH.

Результат: Успешная аутентификация.

```
ssh -T git@github.com
# Hi drrrm471! You've successfully authenticated, but GitHub does not provide shell access.
```

2. Настройка Git и создание локального репозитория

- Установлен Git и настроены глобальные параметры (имя, email, редактор).
- Создана директория проекта и инициализирован локальный репозиторий.

```
git config --global user.name "Roman"
git config --global user.email "makarovr0471@gmail.com"
git init
```

3. Настройка pre-commit хуков

- Установлен инструмент `pre-commit`.
- Создан файл `.pre-commit-config.yaml` с настройками для автоматического форматирования кода.
- Pre-commit хуки активированы.

Содержание `.pre-commit-config.yaml`:

```
repos:  
  - repo: https://github.com/pre-commit/pre-commit-hooks  
    rev: v4.4.0  
    hooks:  
      - id: trailing-whitespace  
      - id: end-of-file-fixer  
      - id: check-yaml  
      - id: check-added-large-files  
  - repo: https://github.com/psf/black  
    rev: 23.3.0  
    hooks:  
      - id: black  
        language_version: python3
```

4. Рабочий процесс с использованием GitHub Flow

- Создан файл `hello.py` с простейшей программой на Python.
- Выполнены коммиты с сообщениями:
 - "Add initial hello.py script"
 - "Add new print statement to hello.py"
- Создана ветка `feature/greeting` для разработки новой функциональности.
- В ветке `feature/greeting` создан файл `greeting.py` с функцией приветствия.
- Изменения из ветки `feature/greeting` слиты в основную ветку `main`.

Пример кода `greeting.py`:

```
def greet_user():  
    name = input("Введите ваше имя: ")  
    print(f"Привет, {name}!")  
  
if __name__ == "__main__":  
    greet_user()
```

5. Работа с удалённым репозиторием

- Создан удалённый репозиторий на GitHub с названием `python-git-lab1`.
- Локальный репозиторий связан с удалённым.
- Ветки `main` и `feature/greeting` отправлены на GitHub.

```
git remote add origin git@github.com:drrrm471/python-git-lab1.git
git push -u origin main
git push origin feature/greeting
```

6. Имитация Git Flow

- Создана ветка `develop` для разработки.
- От ветки `develop` создана ветка `feature/calculator` для реализации калькулятора.
- В ветке `feature/calculator` создан файл `calculator.py` с простой функцией калькулятора.
- Изменения из `feature/calculator` слиты в `develop`.

Пример кода `calculator.py`:

```
def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

if __name__ == "__main__":
    print(add(5, 3))
    print(subtract(5, 3))
```

7. Финальные штрихи

- Создан файл `README.md` с описанием проекта и инструкциями по запуску.
- Добавлены скриншоты выполнения операций (например, работа pre-commit хуков).
- Все изменения отправлены на GitHub.

Результаты

Структура репозитория:

```
python-git-lab1/
├── hello.py
├── greeting.py
├── calculator.py
├── README.md
├── .pre-commit-config.yaml
└── screenshots/
```

```
├── pre-commit-work.png  
└── test-result.png
```

Ссылка на репозиторий:

[GitHub Repository Link](#)

Выводы

- Освоены базовые операции Git: создание репозитория, коммиты, ветвление, слияние.
- Изучены стратегии ветвления GitHub Flow и Git Flow.
- Настроены pre-commit хуки для автоматического контроля качества кода.
- Приобретены навыки работы с удалённым репозиторием на GitHub.

Рекомендуемая литература

1. **Официальная книга "Pro Git"** (доступна онлайн на [git-scm.com](#)).
2. **Vincent Driessen. "A Successful Git Branching Model"** (статья о Git Flow).
3. **Документация pre-commit** (доступна на [pre-commit.com](#)).