**РЕСПУБЛИКА БЕЛАРУСЬ**

**КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА**

**учреждение образования**

МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИИ

##### Группа 89

## Отчет

## по лабораторной работе № 1

**"Работа с системой контроля версий"**

Учебный предмет

«Инструментальное программное обеспечение»

**Исполнитель: Кудлаш И.О.**

**Руководитель: Бровка Д.С.**

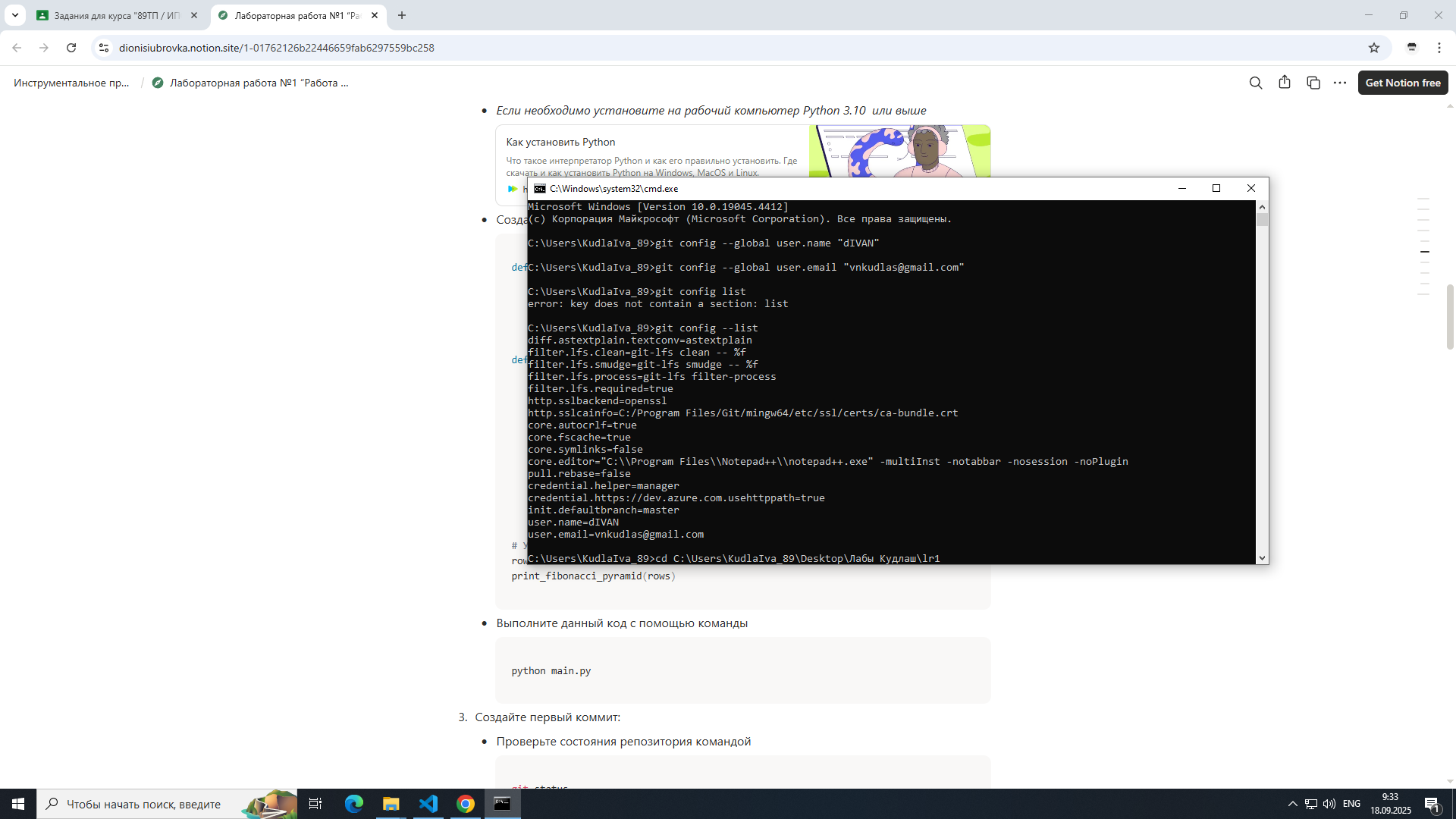
**Минск, 2025**

**Введя команду git --version мы проверяем установку git**

**Настраиваем имя пользователя и электронную почту для Git с помощью команд:**

**git config --global user.name "Ваше Имя"**

**git config --global user.email "ваш.email@example.com"**

****

**Проверяем настройки, выполнив команду**

**Создаем новую папку lr1 для нашего проекта и переходим в неё.**

**Инициализируем новый Git-репозиторий с помощью команды**

**git init**

**Создаем файл main.py, и вставляем в него следующий код:**

**def fibonacci(n):**

**fib\_sequence = [0, 1]**

**for i in range(2, n \* (n + 1) // 2): # Достаточно для создания пирамиды**

**fib\_sequence.append(fib\_sequence[-1] + fib\_sequence[-2])**

**return fib\_sequence**

**def print\_fibonacci\_pyramid(rows):**

**fib\_sequence = fibonacci(rows)**

**index = 0**

**for i in range(1, rows + 1):**

**# Печатаем пробелы для выравнивания**

**print(" " \* (rows - i), end='')**

**for j in range(i):**

**print(fib\_sequence[index], end=' ')**

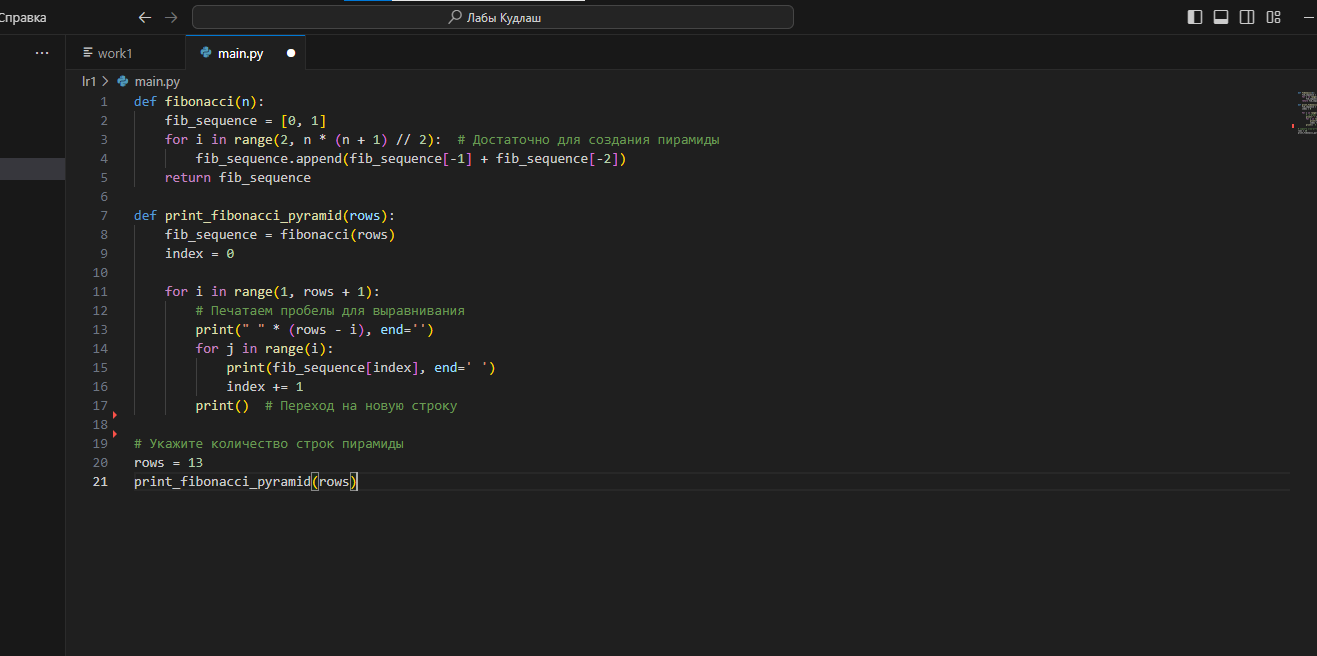
**index += 1**

**print() # Переход на новую строку**

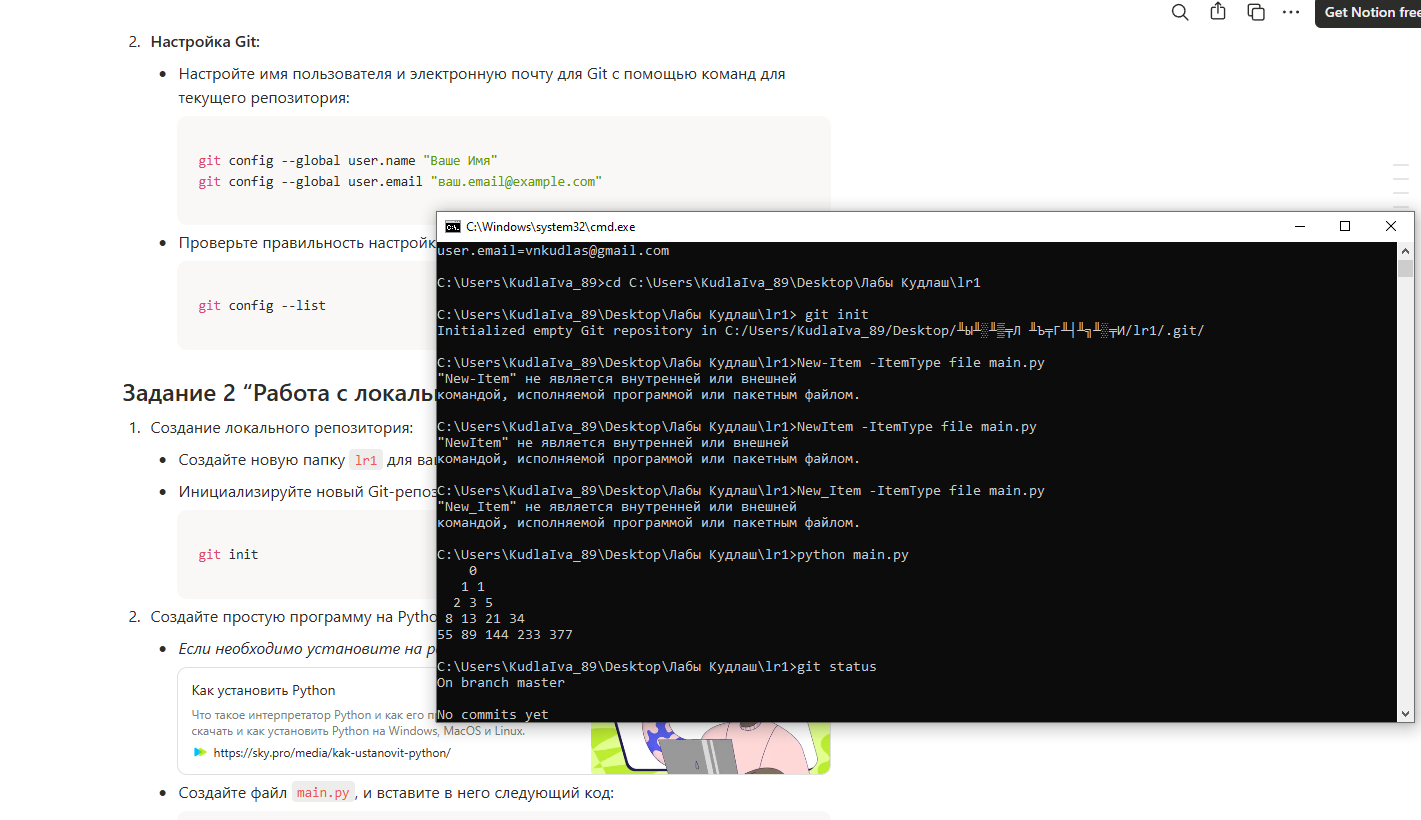
**# Укажите количество строк пирамиды**

**rows = 5**

**print\_fibonacci\_pyramid(rows)**

****

**Выполняем данный код с помощью команды python main.py**

****

**Проверяем состояние репозитория командой git status**

**Добавляем файл в индекс с помощью команды git add main.py**

**Проверяем состояние репозитория командой git status**

**Выполняем первый коммит с описанием "Initial commit". git commit -m "Initial commit"**

**​**

**Проверяем состояние репозитория командой git status**

**Изменяем кол-во строк выводимых программой main.py на свой порядковый номер в журнале**

**# Укажите количество строк пирамиды**

**rows = # Указать свой номер в журнале**

**​Проверяем состояние репозитория командой**

**git status**

**Сохраняем изменения в репозитории**

**git add main.py**

**git commit -m "Сюда вставить описание изменений"**

**Создаем новую ветку с названием "feature" с помощью команды:**

**git branch feature**

**Переключаемся на новую ветку:**

**git checkout feature**

**Изменяем файл main.py, добавив функцию для вычисления факториала:**

**def factorial(n):**

**if n == 0 or n == 1:**

**return 1**

**else:**

**return n \* factorial(n-1)**

**# Добавьте вызов функции в конец файла**

**print(f"Факториал числа {rows}: {factorial(rows)}")**

**​**

**Сохраняем изменения и создаем коммит:**

**git add main.py**

**git commit -m "Добавлена функция вычисления факториала"**

Возвращаемся в основную ветку:

git checkout main

​

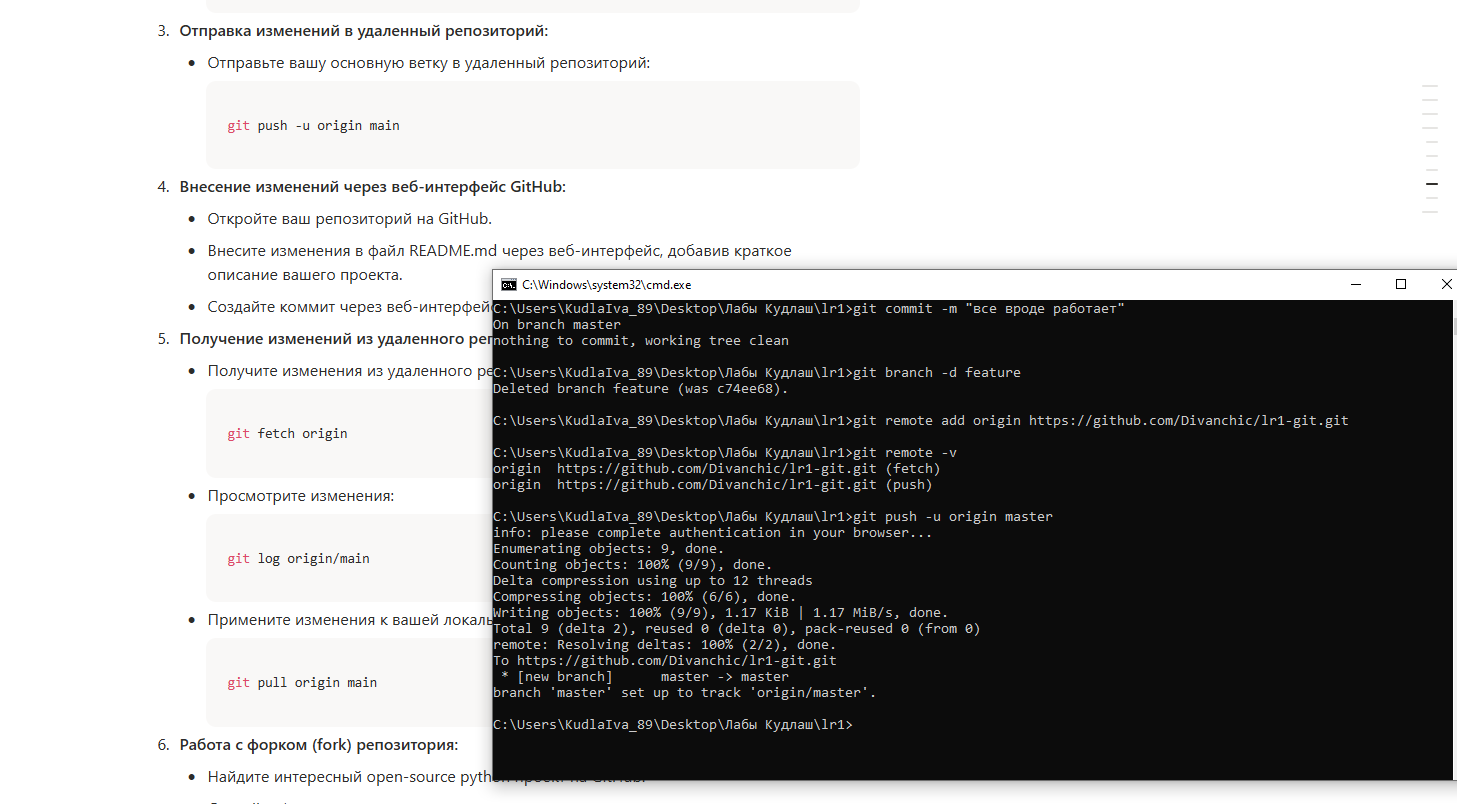
Выполняем слияние ветки "feature" с основной веткой:

git merge feature

​​

После успешного слияния удаляем ветку "feature":

git branch -d feature

****

Войдем в аккаунт на GitHub

Создаем новый репозиторий на GitHub с названием "lr1-git".

Добавьте удаленный репозиторий к вашему локальному репозиторию:

git remote add origin https://github.com/Divanchic/lr1-git.git

Проверяем, что удаленный репозиторий успешно добавлен:

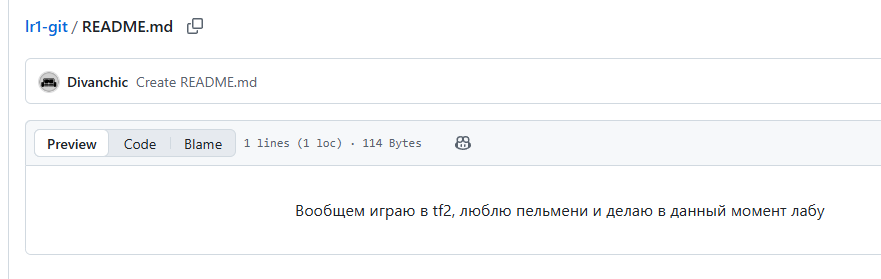
git remote -v

Отправляем основную ветку в удаленный репозиторий:

git push -u origin main

Откроем наш репозиторий на GitHub.

Внесем изменения в файл README.md через веб-интерфейс, добавив краткое описание проекта.



Создаем коммит через веб-интерфейс.

Получите изменения из удаленного репозитория:

git fetch origin

​

Просмотрите изменения:

git log origin/main

​

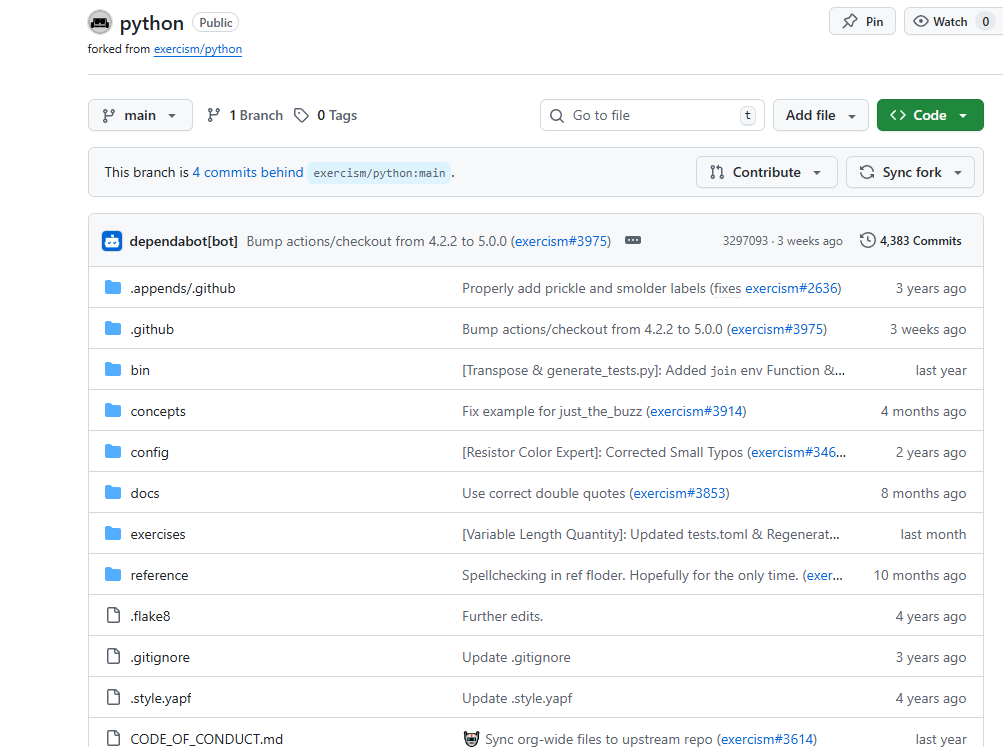
Примените изменения к вашей локальной ветке:

git pull origin main

Находим интересный open-source python проект на GitHub.

Делаем форк этого репозитория.

Клонируйте форк на локальную машину:

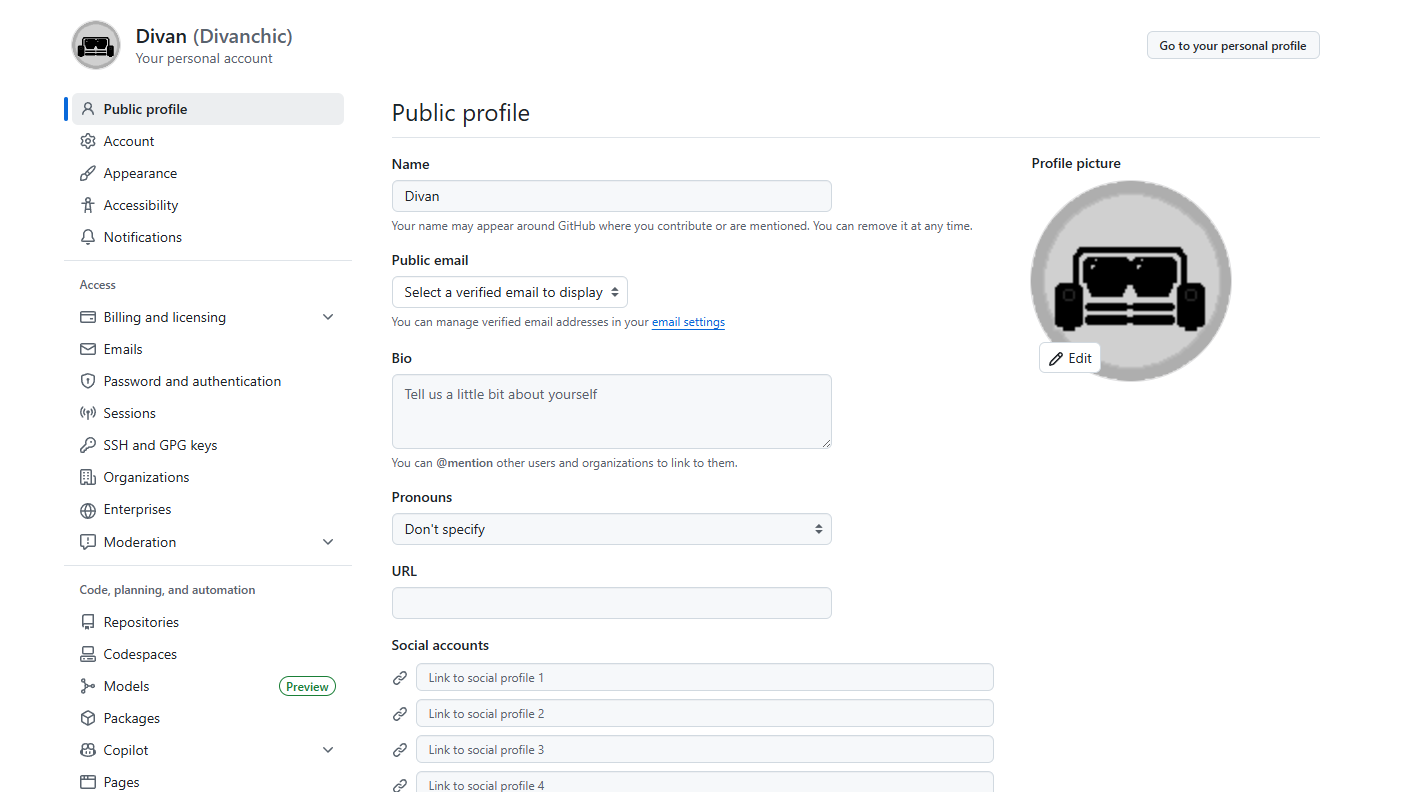


Входим в свой аккаунт на GitHub.

Переходим в настройки профиля (Settings).

Редактируем основную информацию

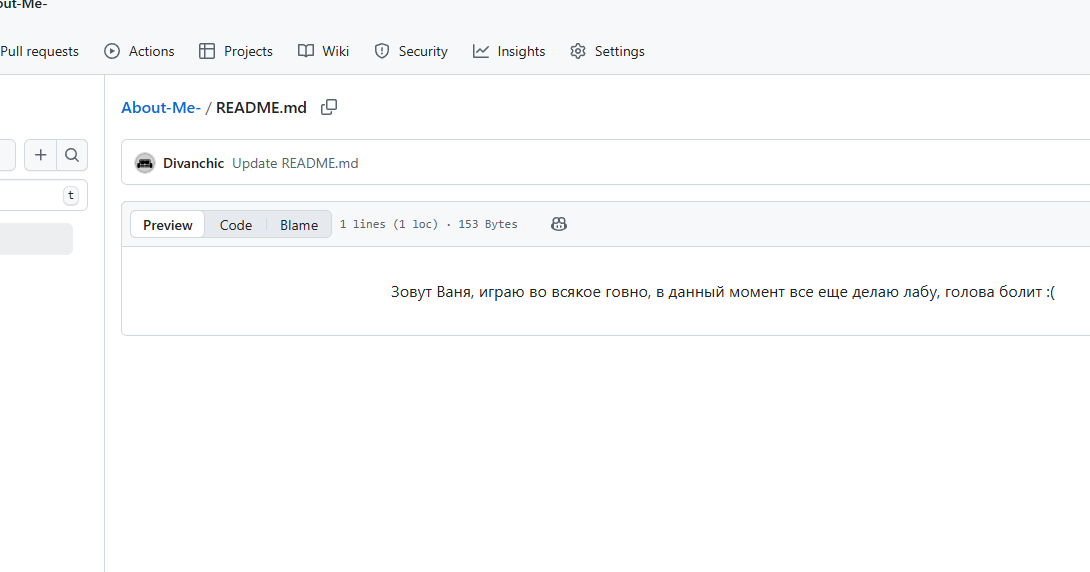
Сохраняем изменения.



Создаем новый репозиторий с именем, совпадающим с нашим именем пользователя на GitHub.

Инициализируем репозиторий с README файлом.

Редактируем README.md, добавив информацию о себе.



Контрольные вопросы:

1. Git init
2. Тег – метка конкретного коммика, git tag
3. git add -all
4. git config –global user.name “имя”, git config –global user.email имейл
5. zip -r repo.zip .-x “\*.git\*”
6. git revert
7. git branch
8. git log
9. git clone <https://github.com/ваш-username/название-форкнутого-репозитория.git>
10. Вручную исправить конфликты, прописать git add, git commit
11. Git checkout, git merge feature-branch
12. Git checkout
13. Git commit -m
14. Git status
15. Командная строка – интерфейс для ввода текстовых команд. Для работы с git прописать git/git help
16. Git add
17. Git branch -d
18. Git – распределенная система контроля версий, позволяющая отслеживать изменения в коде, создавать ветки и управлять историей проекта
19. Git checkout