Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный** **исследовательский политехнический университет»**

Факультет: Прикладной математики и механики

Кафедра: Вычислительной математики, механики и биомеханики

Направление бакалавриата: 09.03.02 Информационные системы и технологии

Профиль: «Информационные системы и технологии»

**О Т Ч Е Т**

по лабораторной работе

по дисциплине

**«Информатика в приложении к отрасли»**

Лабораторная работа №**1**

Тема работы: «**Работа с Git**»

**Выполнил** студент гр. ИСТ-20-1б

**Поплаухин Иван**

**Принял** ассистент кафедры ВММБ

**Георгий Владимирович**

**Пермь 2022**

**Задание №1.** Создание виртуального окружения, установка модулей в виртуальное окружение при помощи менеджера пакетов pip.

1. Создайте проект в PyCharm. При создании проекта создайте виртуальное окружение.
2. Установите в виртуальное окружение модуль **numpy**
3. Установите в виртуальное окружение модуль **matplotlib**
4. Напишите программу, которая вычисляет N первых чисел Фибоначчи, записывает их в массив и строит график: величина числа Фибоначчи от его номера.

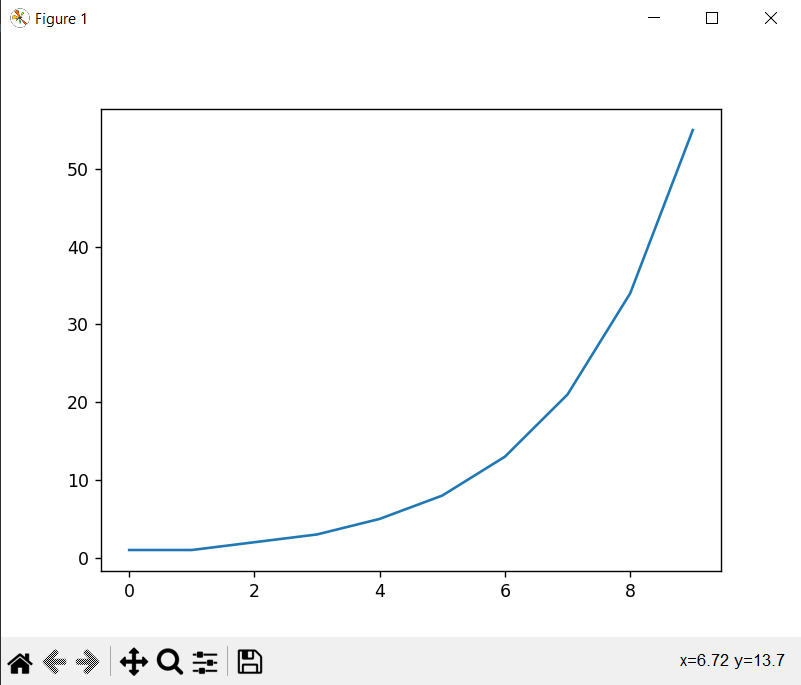


Рис 1. График значений числе Фибоначчи

от номера в последовательности

**Задание №1.** Создание репозитория на сервисе Github через браузер.

1. Создайте репозиторий **lab01** на сервисе Github через браузер.
2. Клонируйте репозиторий на компьютер при помощи GitKraken
3. Создайте проект в PyCharm в данном репозитории
4. Добавьте папку с виртуальным окружением **venv** в исключения системы контроля версий
5. Создайте файл **main.py** в директории проекта с программой, которая реализует алгоритм Евклида – алгоритм нахождения наибольшего общего делителя (НОД) пары целых чисел.
6. Создайте коммит, отправьте его на сервер Github
7. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.

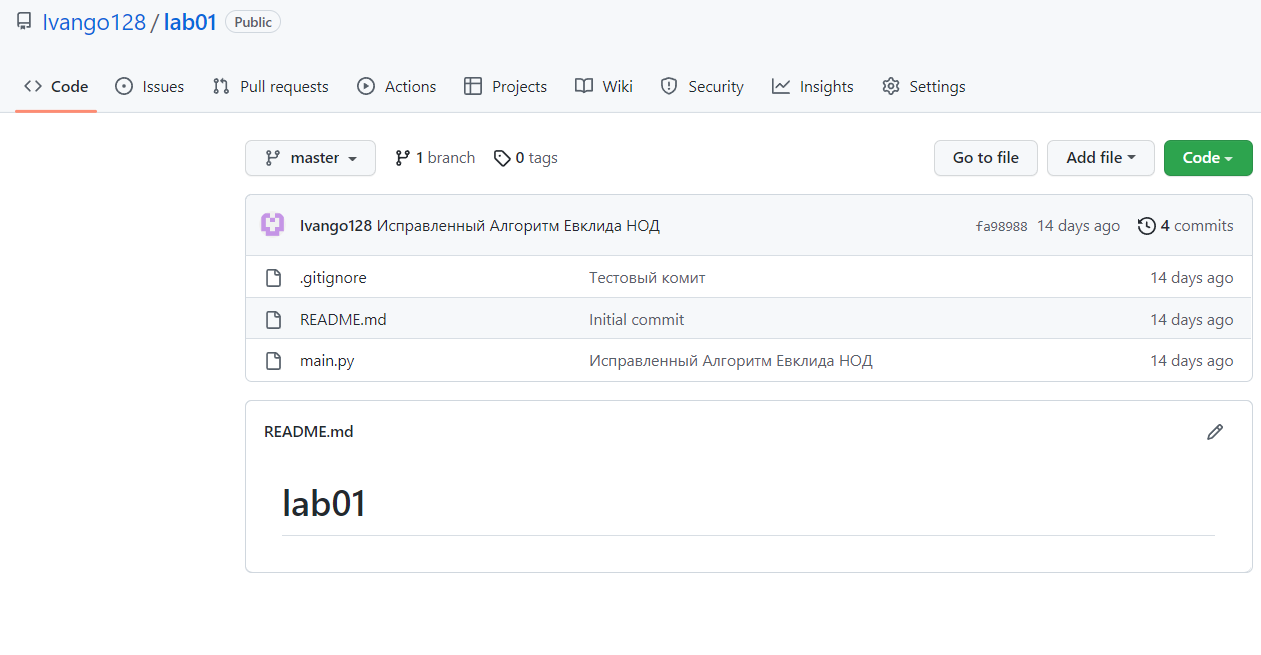


Рис 2. Созданный репозиторий.

**Задание №2.** Создание репозитория на сервисе Github через программу GitKraken.

1. Создайте репозиторий **lab02** на сервисе Github через программу GitKraken.
2. Создайте проект в PyCharm в данном репозитории
3. Добавьте папку с виртуальным окружением **venv** в исключения системы контроля версий
4. Создайте файл **main.py** в директории проекта с программой, которая реализует алгоритм решето Эратосфена – алгоритм нахождения простых чисел до заданного натурального числа путем постепенного отсеивания составных чисел.
5. Создайте коммит, отправьте его на сервер Github
6. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории. Убедитесь, что в облако не попала папка venv.

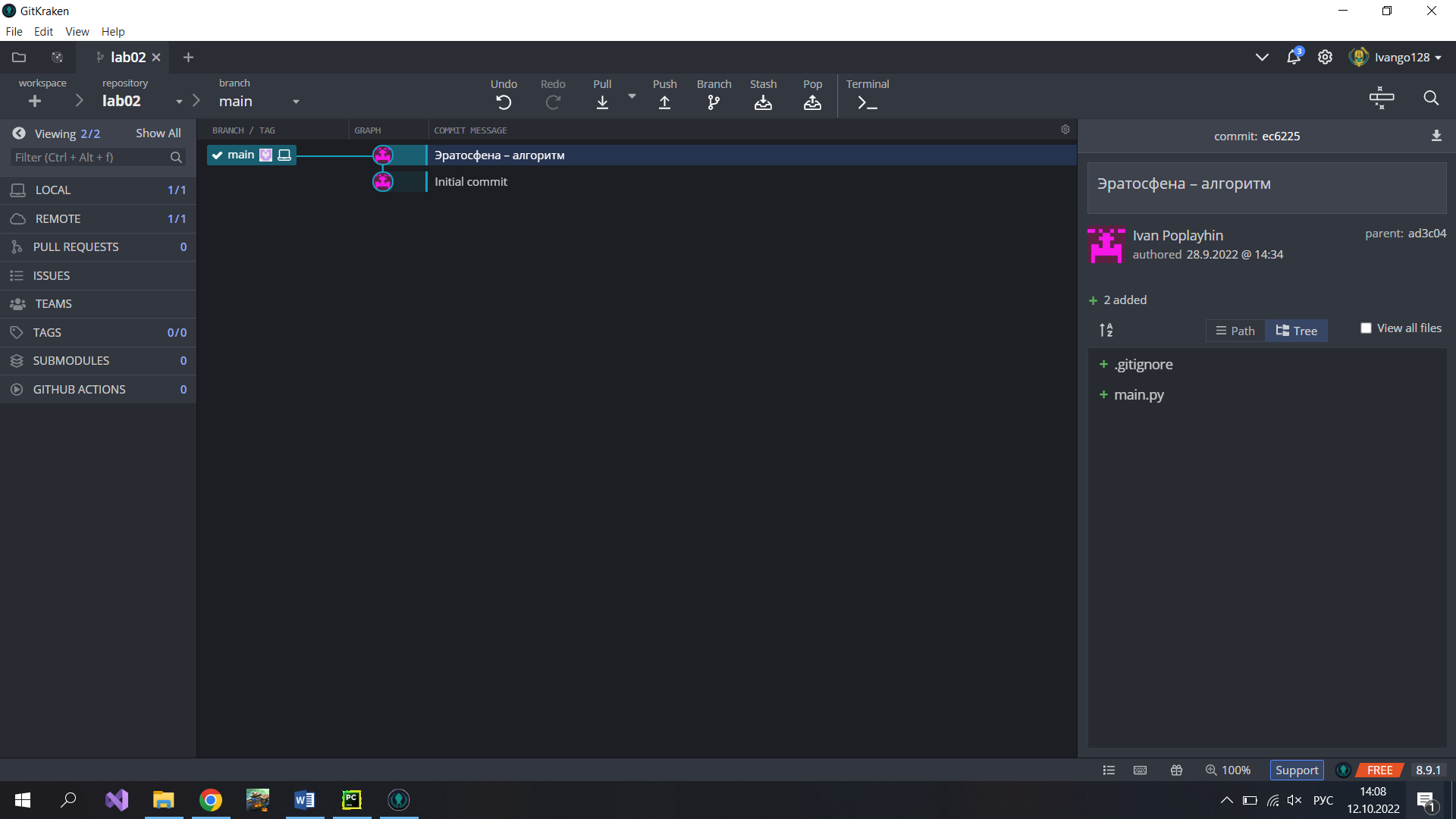


Рис 3. Созданный репозиторий в GitKraken.

**Задание №3.** Основы работы с ветками

1. Создайте репозиторий **lab03** на сервисе Github через программу GitKraken.
2. Создайте проект в PyCharm в данном репозитории
3. Добавьте папку с виртуальным окружением **venv** в исключения системы контроля версий
4. Создайте файл **main.py** в директории проекта с программой, которая запрашивает с клавиатуры n чисел и сортирует их при помощи алгоритма сортировки пузырьком по возрастанию и выводит их на экран.
5. Создайте коммит, отправьте его на сервер Github
6. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.
7. Создайте ветку от текущего коммита. Назовите ветку **new-feature,** переключитесь на неё.
8. Измените программу так, чтобы она после чтения чисел запрашивала направление сортировки (по возрастанию/по убыванию) и производила сортировку соответствующим образом.
9. Создайте коммит, отправьте его на сервис Github.
10. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.
11. При помощи программы GitKraken, слейте изменения из ветки **new-feature** в основную ветку вашего репозитория.
12. Отправьте изменения на сервер GitHub.
13. Откройте в браузере репозиторий, убедитесь, что внесенные изменения находится в репозитории.

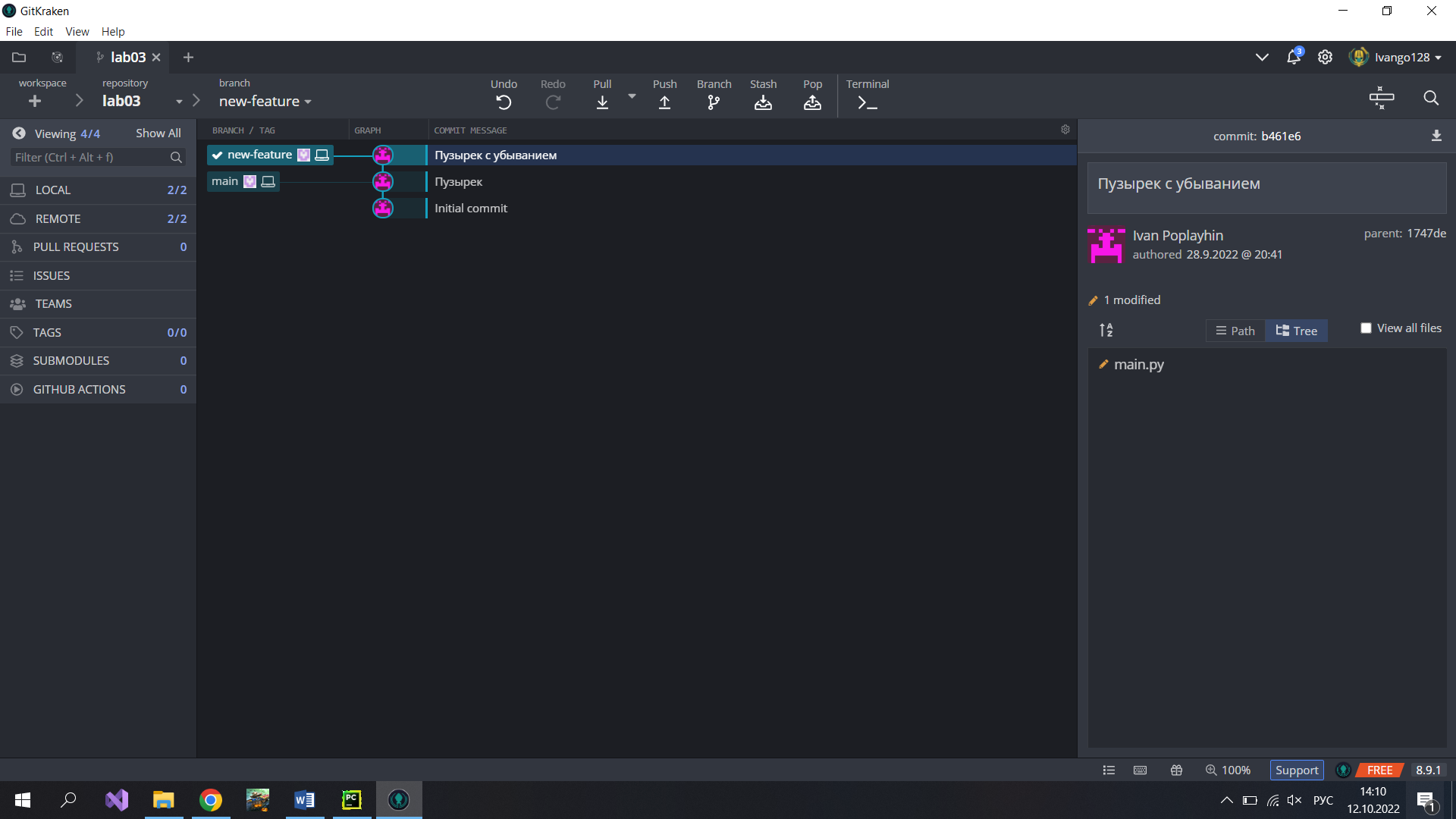


Рис 4. Репозиторий с ветвлением.

**Задание №4а.** Основы работы с pull-requests.

1. Создайте репозиторий **lab04** на сервисе Github через программу GitKraken.
2. Создайте проект в PyCharm в данном репозитории
3. Добавьте папку с виртуальным окружением **venv** в исключения системы контроля версий
4. Создайте файл **lib.py** в директории проекта с функцией, которая принимает на вход **N** списков и возвращает количество одинаковых элементов в них.
5. Создайте коммит, отправьте его на сервер Github
6. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.

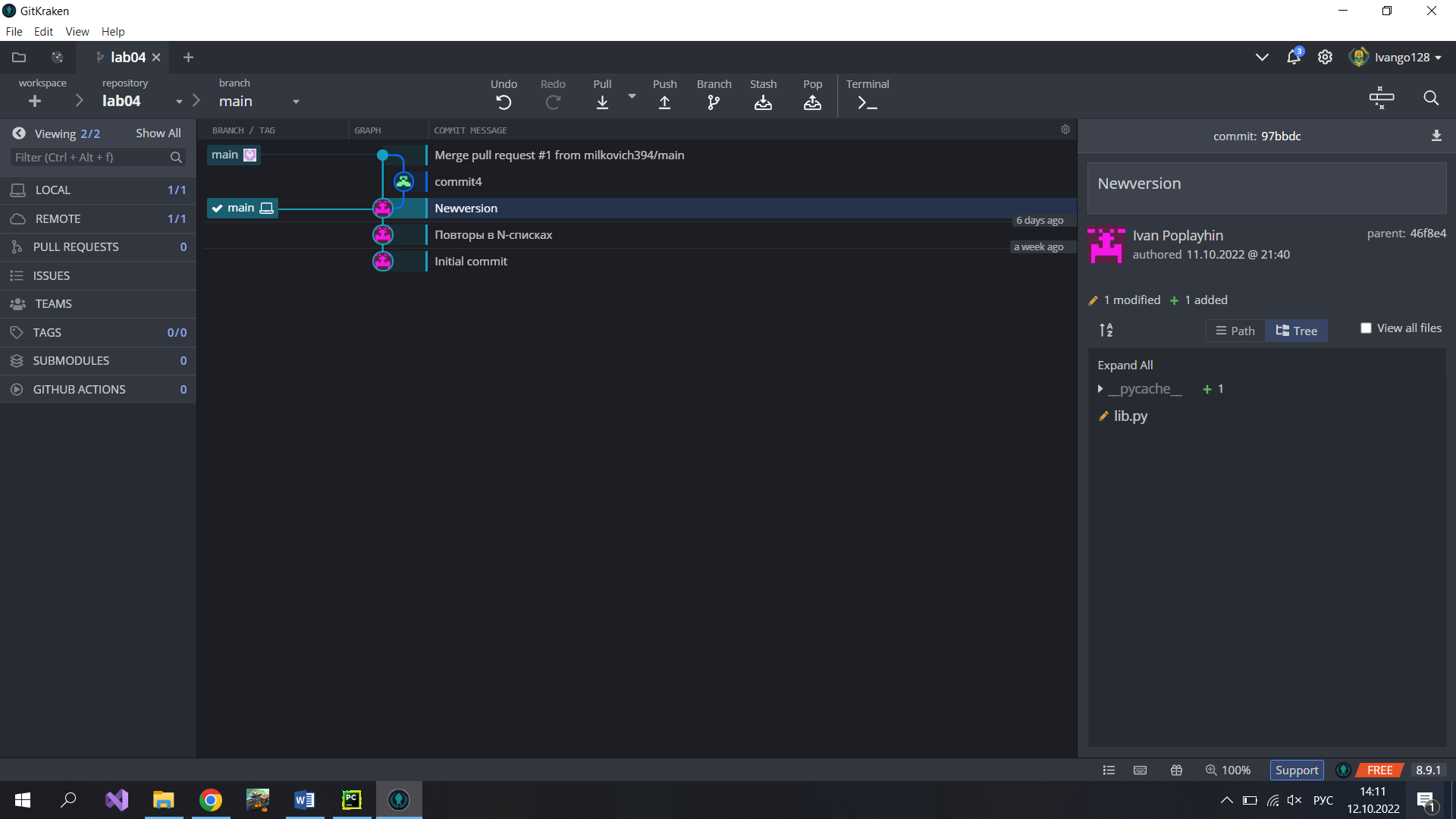


Рис 5.Собственый репозиторий в котом сделан форк.

**Задание №4б.**

1. Откройте на сайте Github проект коллеги **lab04** и сделайте **fork**.
2. Выгрузите на компьютер созданный репозиторий.
3. Добавьте программу для тестирования модуля в файл **test.py**.
4. Создайте коммит и отправьте его на сервис **Github.**
5. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.
6. Создайте запрос на слияние в исходный репозиторий.
7. Убедитесь, что слияние произошло успешно, обновленная программа находится в исходном репозитории.

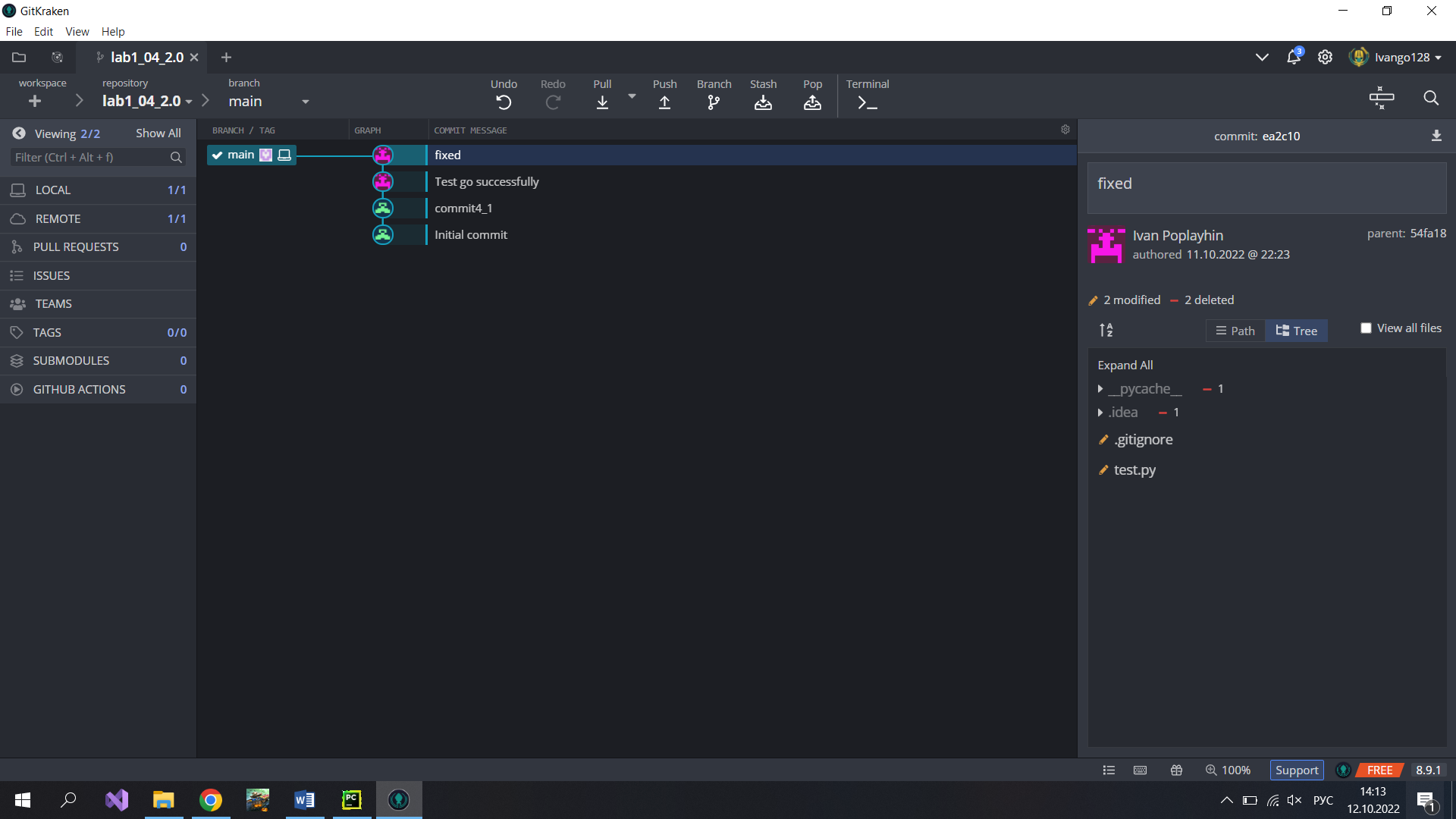


Рис 6. Репозитории коллеги в котором сделан форк.

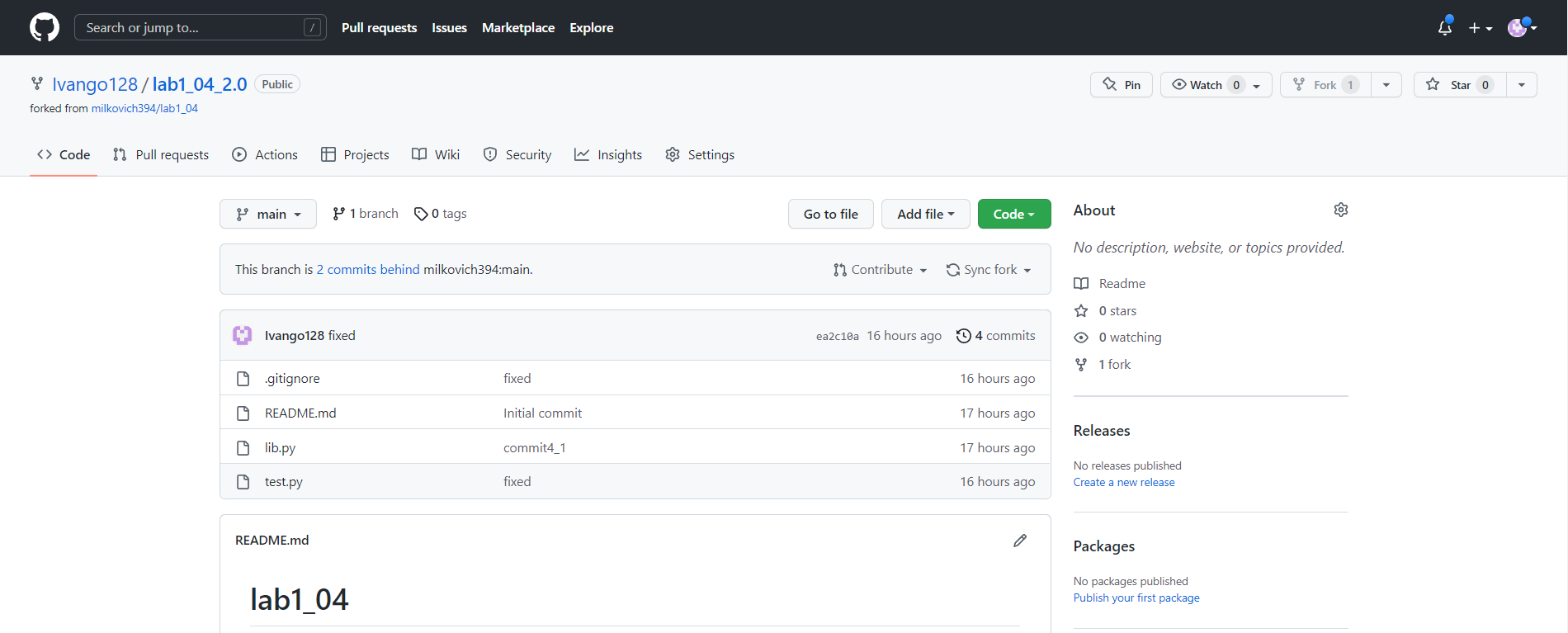


Рис 7. Репозитории коллеги в котором сделан форк.