

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Плещенко Данила Георгиевич
1 курс, группа ИТС-б-о-21-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук, доцент
кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Цель работы: исследовать базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.

Ход работы:

Создам общедоступный репозиторий на Github.

Регистрируюсь на Github.

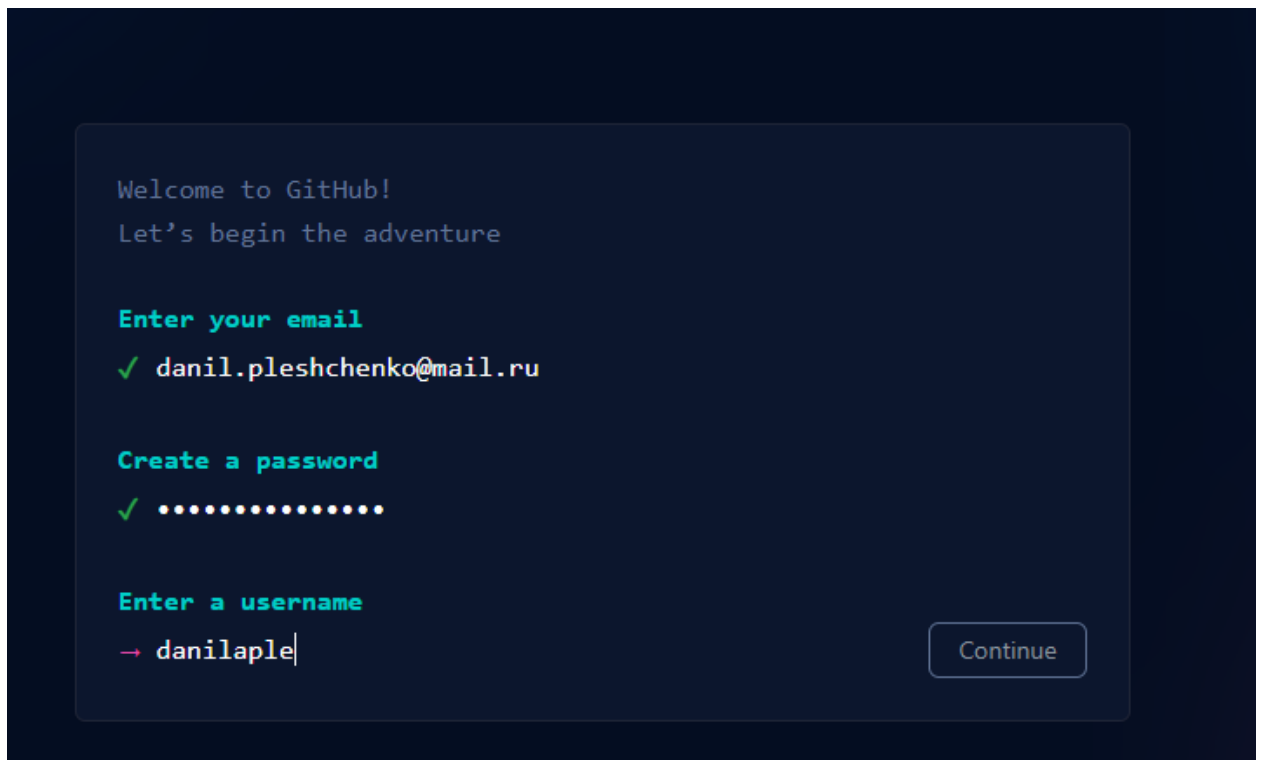



Рисунок 1. Регистрация на Github.


Создадим новый репозиторий.


Owner * Repository name *

 danilaple / 1.lab.rab ✓

Great repository names are short, lowercase, and contain only numbers, lowercase letters, and hyphens. Your new repository will be created as 1.1-b.rab. How about **refactored-fiesta**?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.


☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)



☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)


 You are creating a public repository in your personal account.

Create repository


Рисунок 2. Создание репозитория с добавлением .gitignore.
Скопирую репозиторий для клонирования на компьютер.

 **Clone** 

HTTPS SSH GitHub CLI

`https://github.com/danilaple/1.lab.rab.git` 

Use Git or checkout with SVN using the web URL.

 **Open with GitHub Desktop**


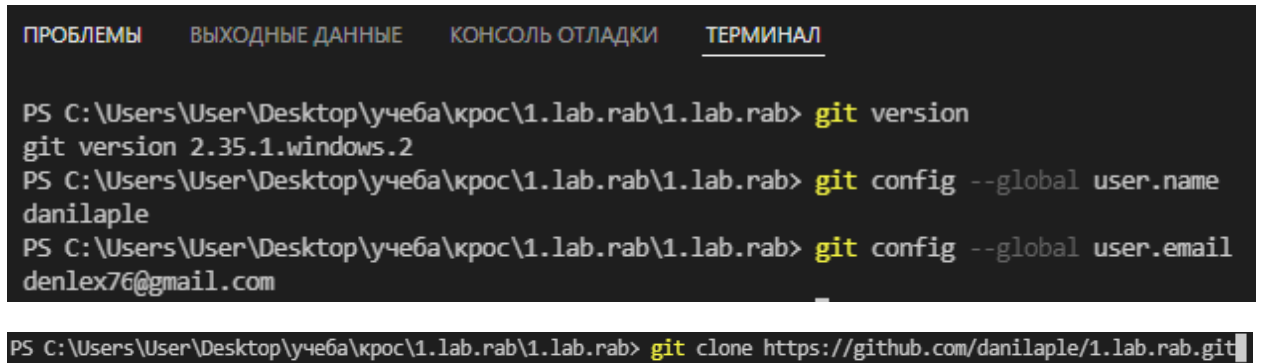
 **Download ZIP**

Рисунок 3. Ссылка на репозиторий.

После этого захожу в любой терминал и связываюсь с GitHub, а также делаю клон репозитория для работы с ним.



```
ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

PS C:\Users\User\Desktop\учеба\крос\1.lab.rab\1.lab.rab> git version
git version 2.35.1.windows.2
PS C:\Users\User\Desktop\учеба\крос\1.lab.rab\1.lab.rab> git config --global user.name
danilaple
PS C:\Users\User\Desktop\учеба\крос\1.lab.rab\1.lab.rab> git config --global user.email
denlex76@gmail.com

PS C:\Users\User\Desktop\учеба\крос\1.lab.rab\1.lab.rab> git clone https://github.com/danilaple/1.lab.rab.git
```

Рисунок 4. Подготовка репозитория для работы.

После этого создал все необходимые файлы для работы, указав их формат.

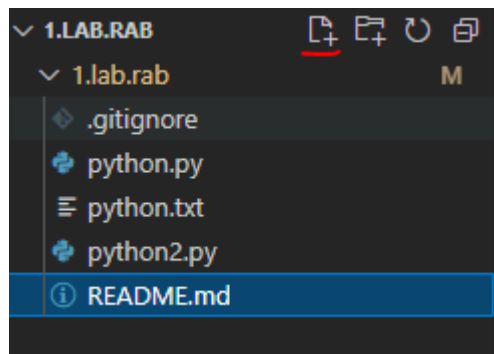


Рисунок 5. Создание файлов.

В файл «.gitignore» добавил python.txt, в котором хранятся необходимые правила для выбранного языка программирования. В моём случае это Python.

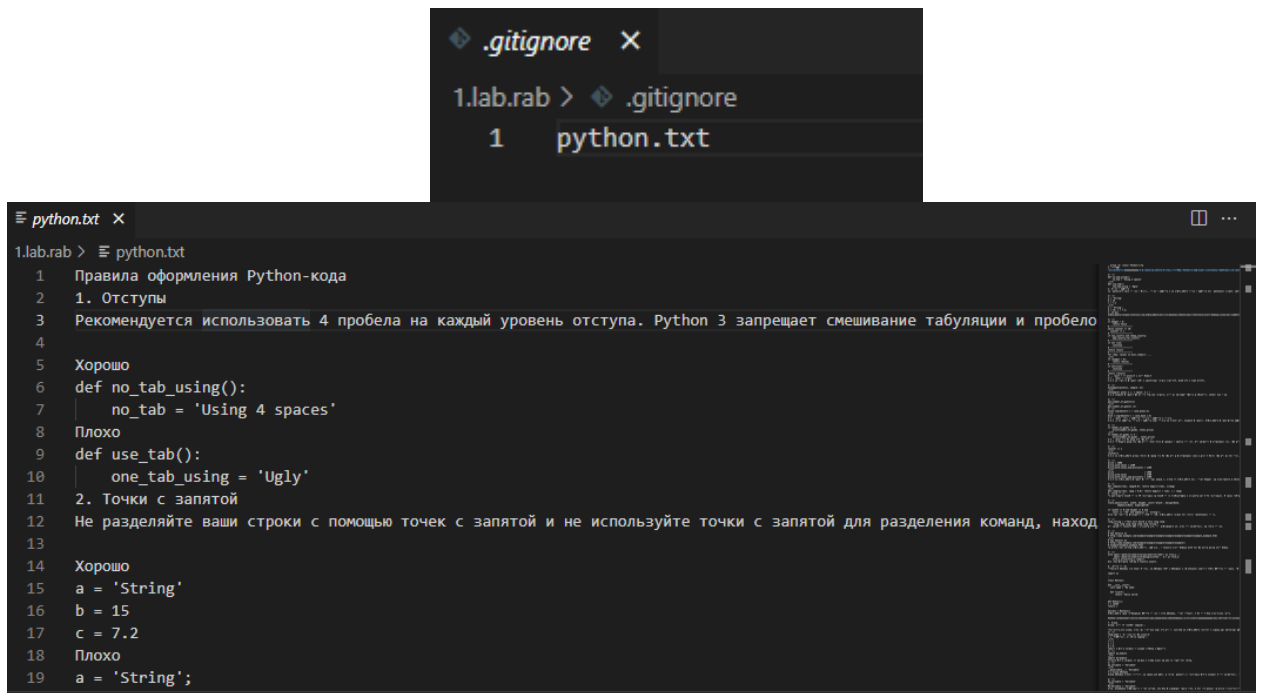


Рисунок 6. Gitignore и необходимые правила.

Далее создаю файл на Python и пишу маленькую программу. Запускаю её и у меня получился краткий диалог.

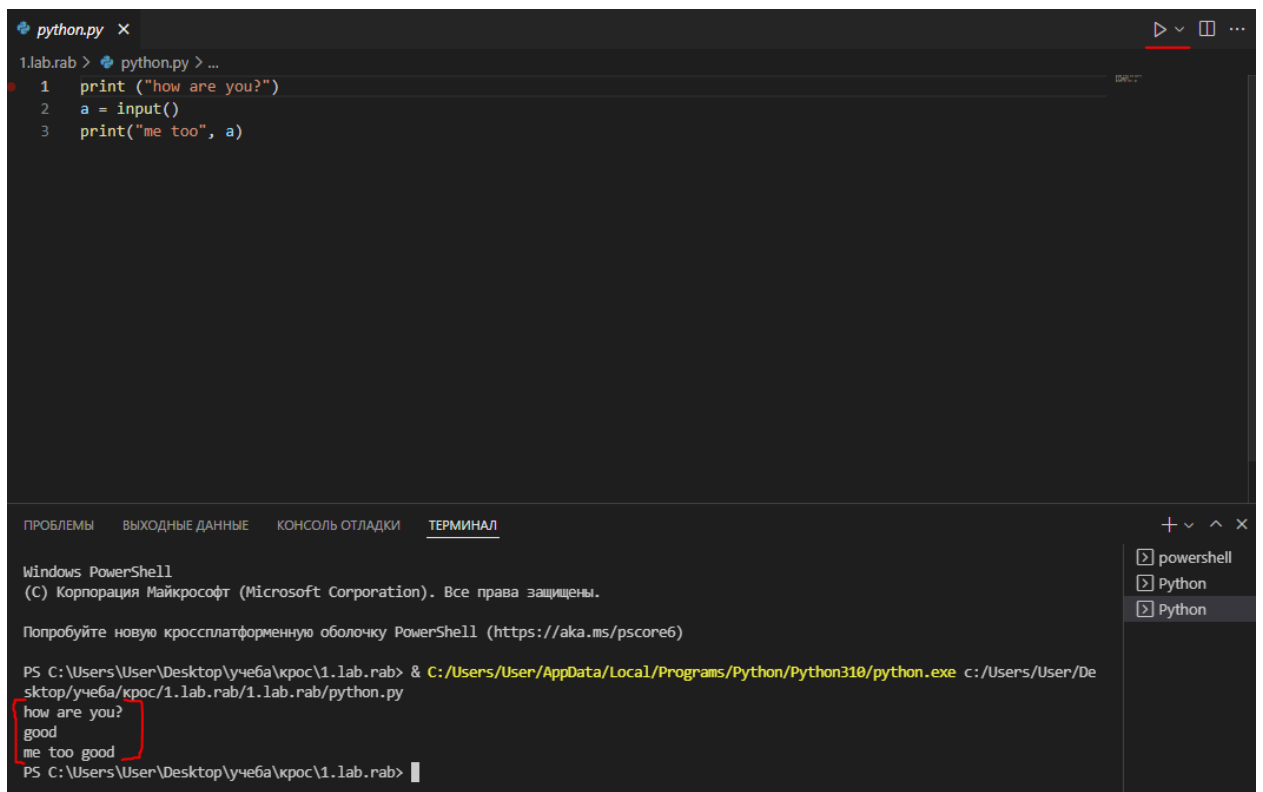
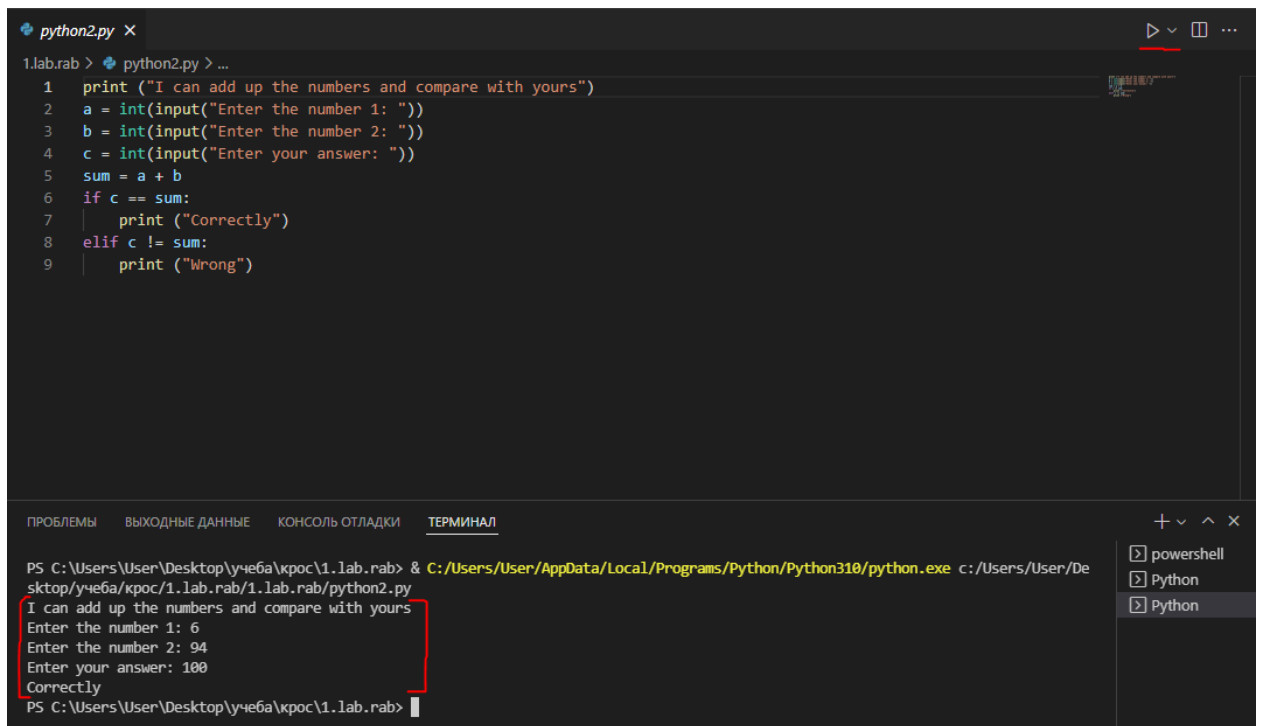


Рисунок 7. Небольшой диалог.

Следующая написанная мной программа немного тяжелее, она проверяет правильность сложения двух чисел.



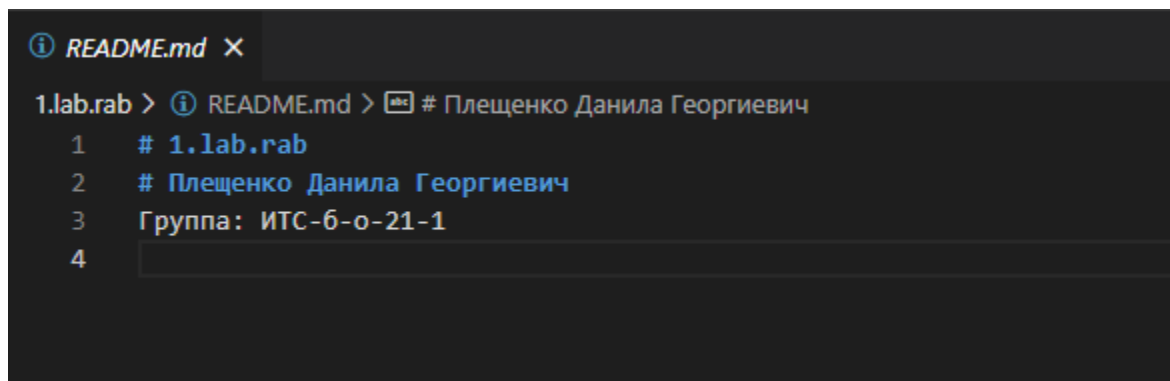
The image shows a VS Code editor with a file named `python2.py` open. The code is a simple Python program that asks the user for two numbers and their sum, then checks if the sum is correct. Below the editor, the terminal window shows the command prompt running the script. The output shows the program asking for input, receiving '6' and '94', calculating the sum '100', and printing 'Correctly'.

```
python2.py
1 lab.rab > python2.py > ...
1 print ("I can add up the numbers and compare with yours")
2 a = int(input("Enter the number 1: "))
3 b = int(input("Enter the number 2: "))
4 c = int(input("Enter your answer: "))
5 sum = a + b
6 if c == sum:
7     print ("Correctly")
8 elif c != sum:
9     print ("Wrong")

PS C:\Users\User\Desktop\учеба\крос\1.lab.rab> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe c:/Users/User/Desktop/учеба/крос/1.lab.rab/1.lab.rab/python2.py
I can add up the numbers and compare with yours
Enter the number 1: 6
Enter the number 2: 94
Enter your answer: 100
Correctly
PS C:\Users\User\Desktop\учеба\крос\1.lab.rab>
```

Рисунок 8. Сложение двух чисел.

В файл README.md ввожу данные согласно методическим указаниям.



The image shows a VS Code editor with a file named `README.md` open. The file contains a header with a hash symbol and the name 'Плещенко Данила Георгиевич', followed by a list of items: '# 1.lab.rab', '# Плещенко Данила Георгиевич', 'Группа: ИТС-6-о-21-1', and an empty line.

```
1 lab.rab > README.md > # Плещенко Данила Георгиевич
1 # 1.lab.rab
2 # Плещенко Данила Георгиевич
3 Группа: ИТС-6-о-21-1
4
```

Рисунок 10. Заполненный файл README.md

После всех изменений сохраняем их командой «**git add .**», а после сохранения отправляю все наработки в GitHub командой «**git push**».

Обновляю страницу репозитория и вижу все изменения.

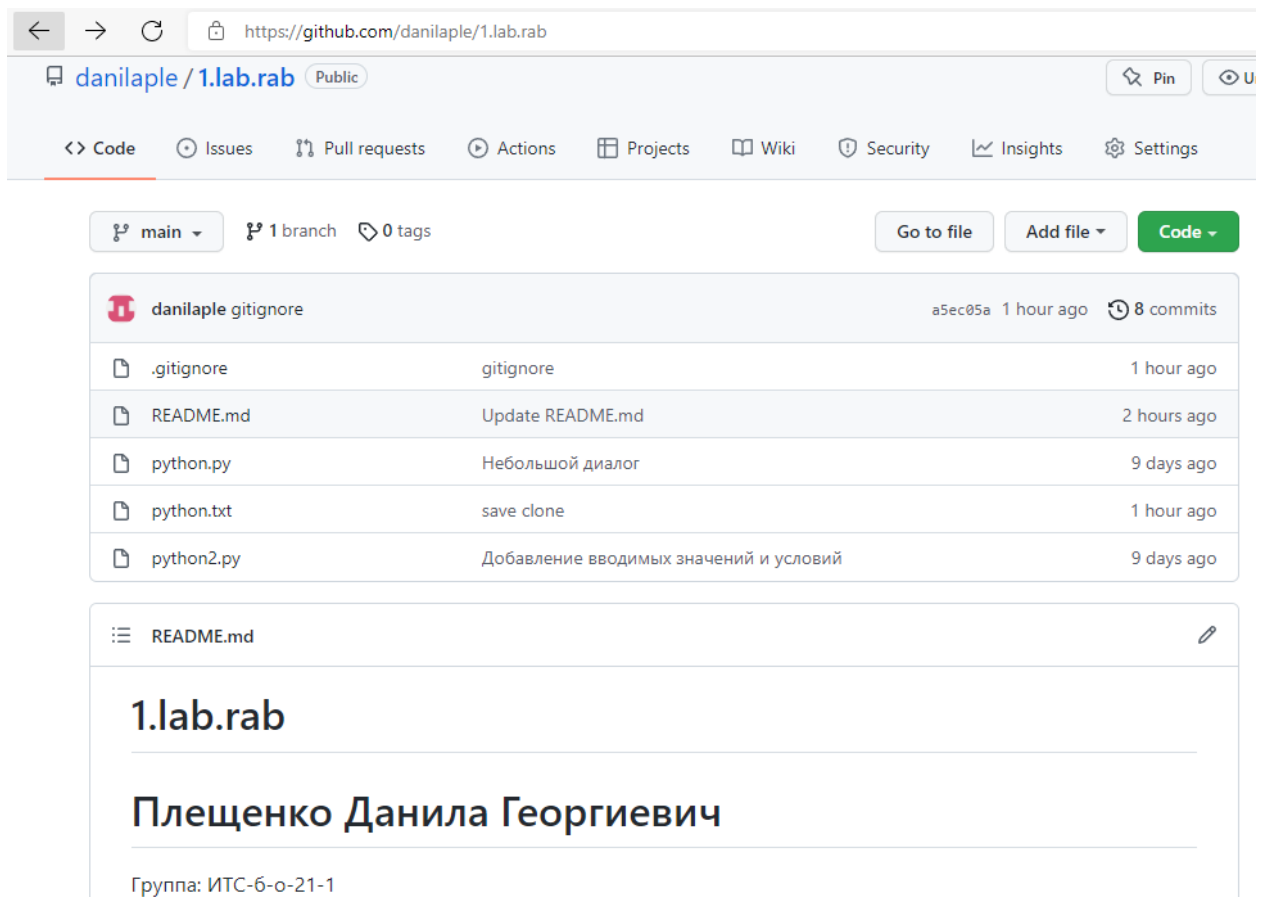


Рисунок 11. Все данные находятся на GitHub.

Контрольные вопросы:

1. Что такое СКВ и каково ее назначение?

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах этих файлов.

2. В чем недостатки локальных и централизованных СКВ?

Это единая точка отказа, представленная централизованным сервером. Если этот сервер выйдет из строя на час, то в течение этого времени никто не сможет использовать контроль версий для сохранения изменений, над которыми работает, а также никто не сможет обмениваться этими изменениями с другими разработчиками.

3. К какой СКВ относится Git?

К распределённым системам контроля версий

4. В чем концептуальное отличие Git от других СКВ?

Git не хранит и не обрабатывает данные таким же способом как другие СКВ.

5. Как обеспечивается целостность хранимых данных в Git?

В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме.

6. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?

- 1) Зафиксированный значит, что файл уже сохранён в вашей локальной базе.
- 2) К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы.
- 3) Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит.

7. Что такое профиль пользователя в GitHub?

Профиль - это наша публичная страница на GitHub, как и в социальных сетях. В нем другие пользователи могут посмотреть ваши работы.

8. Какие бывают репозитории в GitHub?

- 1) инструменты тестирования;
- 2) фрагменты кода;
- 3) советы;
- 4) концепции программирования;
- 5) базы знаний;
- 6) примеры;
- 7) справочники;
- 8) руководства;
- 9) шаблоны;
- 10) демонстрации;
- 11) книги;
- 12) среды разработки;
- 13) чеклисты;
- 14) библиотеки.

9. Укажите основные этапы модели работы с GitHub.

- 1) Регистрация.
- 2) Создание репозитория.
- 3) Клонирование репозитория.
- 4) Добавление новых файлов.

10. Как осуществляется первоначальная настройка Git послеустановки?

Убедимся, что Git установлен используя команду: `git version`. Перейдём в папку с локальным репозиторием используя команду: `cd /d <Расположения папки на компьютере>`. Свяжем локальный репозиторий и удалённый командами: `gitconfig --globaluser.name<YOUR_NAMEgitconfig --globaluser.email<EMAIL>`

11. Опишите этапы создания репозитория в GitHub.

1) В правом верхнем углу, рядом с аватаром есть кнопка с плюсиком, нажимая которую мы переходим к созданию нового репозитория.

2) В результате будет выполнен переход на страницу создания репозитория. Наиболее важными на ней являются следующие поля: Имя репозитория. Описание (Description). Public/private. “Initialize this repository with a README” .gitignore и LICENSE

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Microsoft Reciprocal License, The Code Project Open License

(CPOL), The Common Development and Distribution License (CDDL), The Microsoft Public License (Ms-PL), The Mozilla Public License 1.1 (MPL 1.1), The Common Public License Version 1.0 (CPL), The Eclipse Public License 1.0, The MIT License, The BSD License, The Apache License, Version 2.0, The Creative Commons Attribution-ShareAlike 2.5 License, The zlib/libpng License, A Public Domain dedication, The Creative Commons Attribution 3.0 Unported License, The Creative Commons)

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

После создания репозитория его необходимо клонировать на ваш компьютер. Для этого на странице репозитория необходимо найти кнопку Clone или Code и щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования

Откройте командную строку или терминал и перейдите в каталог, куда вы хотите скопировать хранилище. Затем напишите `git clone` и введите адрес.

14. Как проверить состояние локального репозитория Git?

`git status`

15. Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/измененного файла под версионный контроль с помощью команды `git add`; фиксации (коммита) изменений с помощью команды `git commit` и отправки изменений на сервер с помощью команды `git push`?

Файлы обновятся на репозитории.

16. У Вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы можете осуществлять работу над некоторым проектом с использованием этого репозитория. Опишите последовательность команд, с помощью которых оба локальных репозитория, связанных с репозиторием GitHub будут находиться в синхронизированном состоянии.

`git clone.`

`git pull`

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.

1) GitLab — альтернатива GitHub номер один. GitLab предоставляет не только веб-сервис для совместной работы, но и программное обеспечение с открытым исходным кодом.

2) BitBucket — это служба хостинга репозитория и управления версиями от Atlassian. Она тесно интегрирована с другими инструментами Atlassian — Jira, HipChat и Confluence.

18. Интерфейс командной строки является не единственным и далеко не самым удобным способом работы с Git. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git?

Приведите как реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств.

GitHub Desktop это совершенно бесплатное приложение с открытым исходным кодом, разработанное GitHub. С его помощью можно взаимодействовать с GitHub (что и не удивительно), а также с другими платформами (включая Bitbucket и GitLab).

Вывод: исследовал базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.