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 «Программирование на Python»

Выполнил:
Гайчук Дарья Дмитриевна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика
и вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р.А.-доцент кафедры
инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: «Исследование основных возможностей Git и GitHub»

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

Краткие теоретические сведения

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

GIT — это распределенная система управления версиями, которая широко используется для разработки программного обеспечения.

Основными понятиями в GIT являются репозитории, коммиты, ветки и слияния. Репозиторий — это хранилище всей истории изменений проекта. Коммиты — это снимки состояния проекта на определенный момент времени, включая все изменения в файлах. Каждый коммит имеет уникальный идентификатор, который позволяет легко отслеживать изменения в проекте. Ветки — это отдельные ветви разработки, которые позволяют работать над разными функциональности или исправлениями одновременно без вмешательства в другие ветки. Слияния — это процесс объединения изменений из одной ветки в другую.

GIT предлагает множество команд для управления версиями, таких как git commit, git branch, git merge и др. Эти команды позволяют разработчикам сохранять изменения, создавать новые ветки, сливать ветки и просматривать историю изменений. Кроме того, GIT обеспечивает возможность работать с удаленными репозиториями, что позволяет командам программистов сотрудничать над проектами из разных мест.

Порядок выполнения работы

1. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования

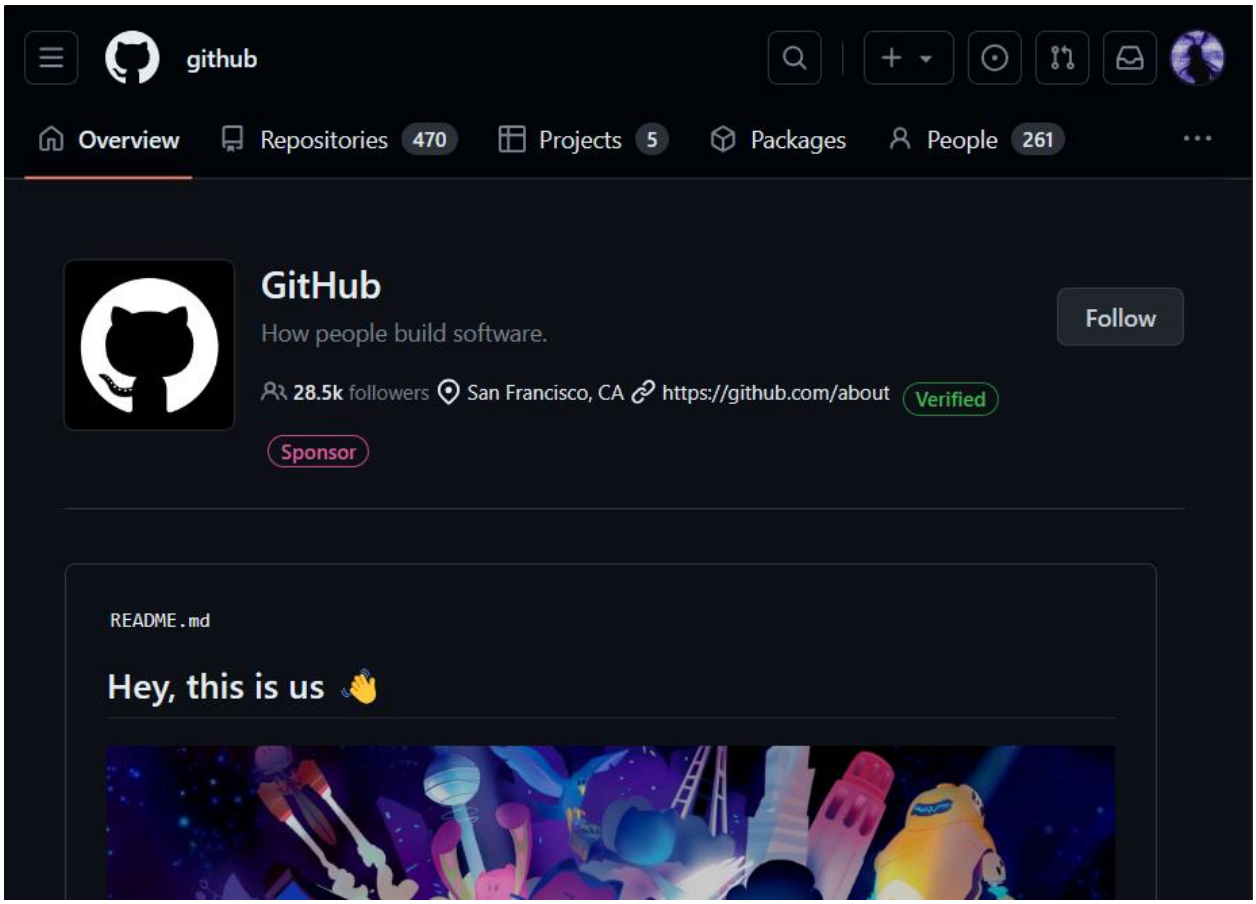

Рисунок 1. Приветственная страница GitHub

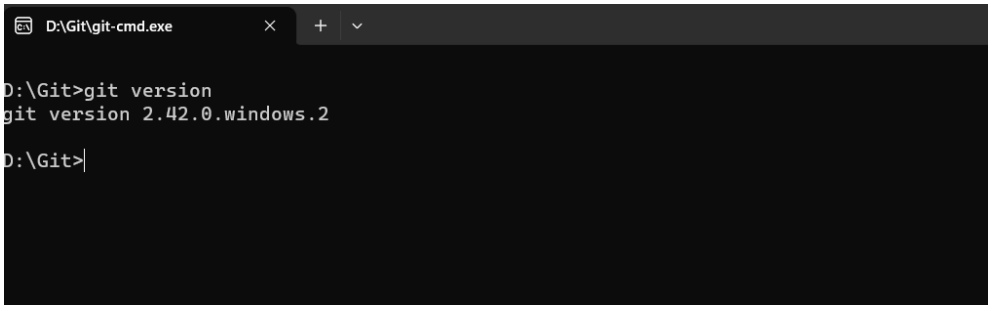

Рисунок 2. Версия Git

```
D:\Git>git config --global user.name "Ichizuchi"

D:\Git>git config --global user.email "doriskotl2@gmail.com"
```

Рисунок 3. Привязка пользователя

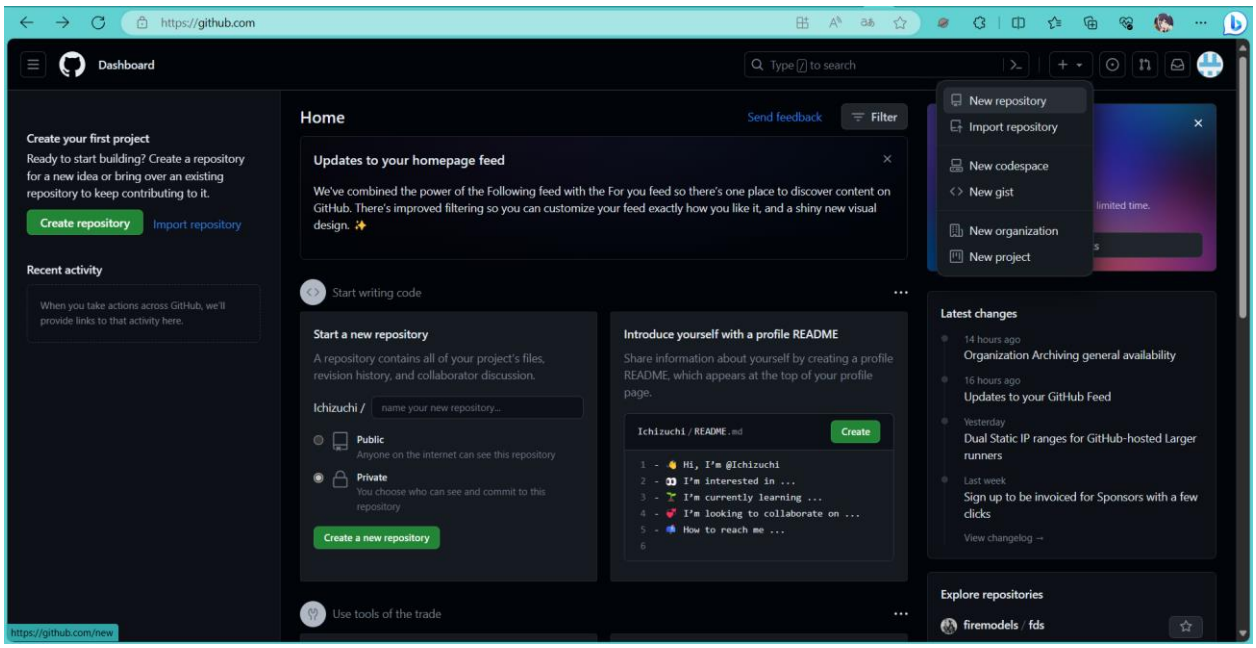

Рисунок 4. Переход к созданию репозитория

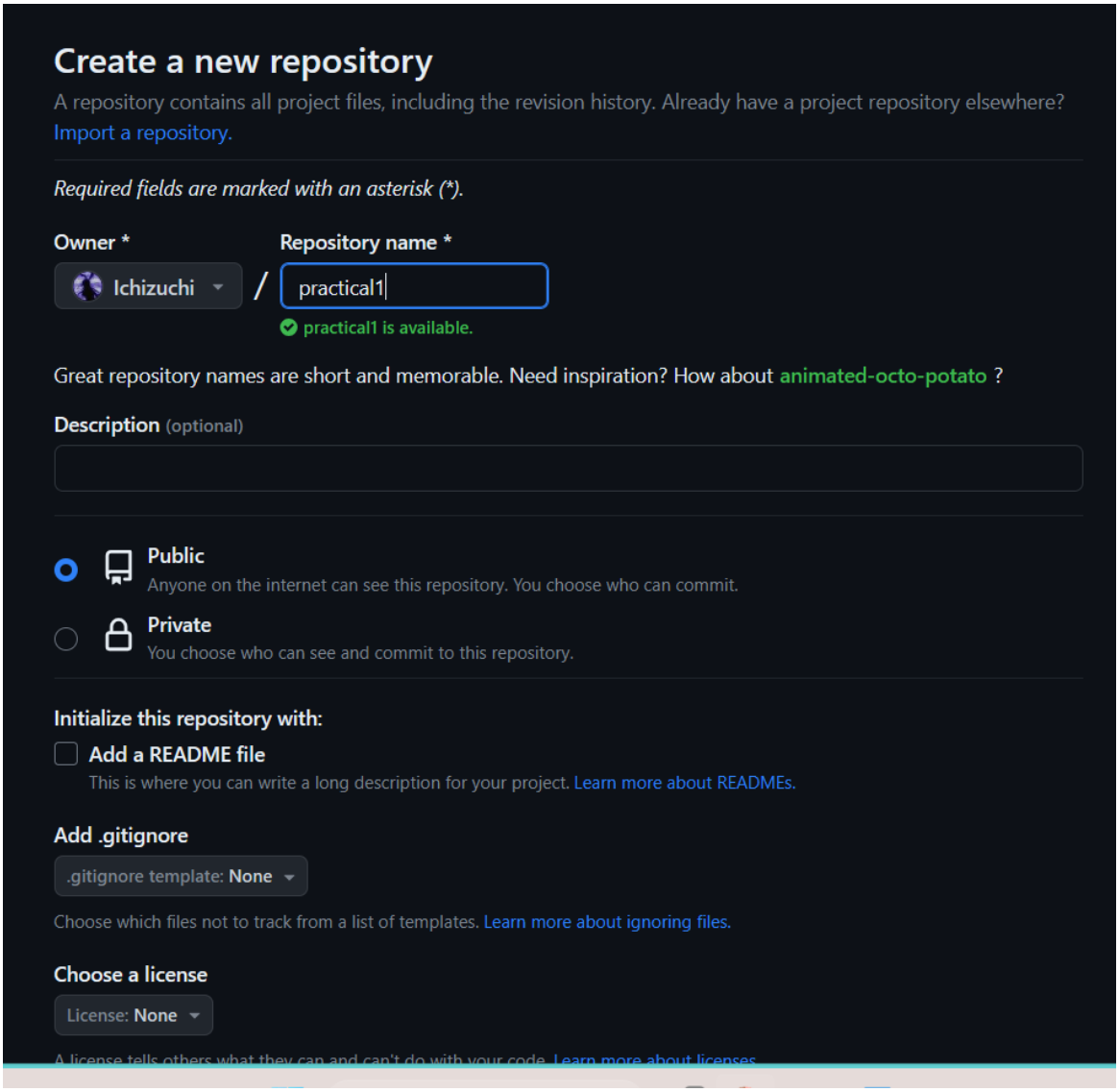

Рисунок 5. Заполнение полей

2. Выполните клонирование созданного репозитория на рабочий компьютер.

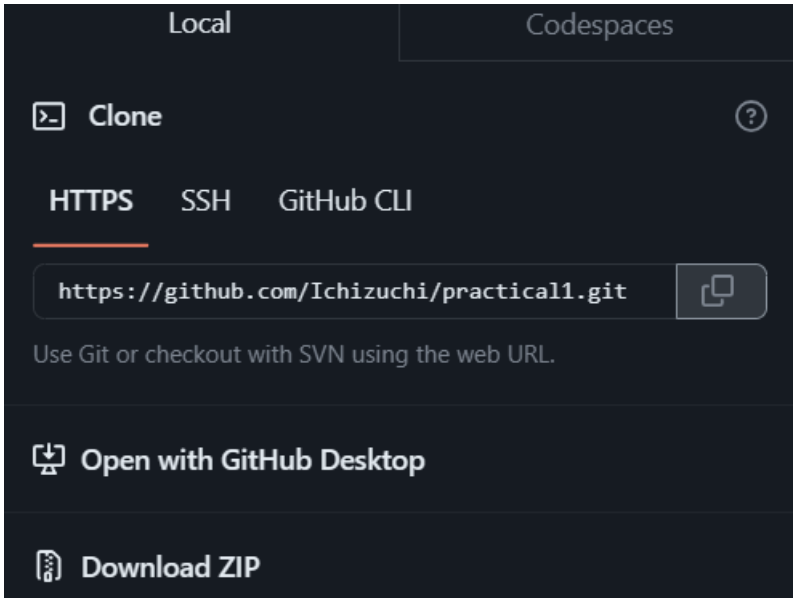

Рисунок 6. Ссылка репозитория

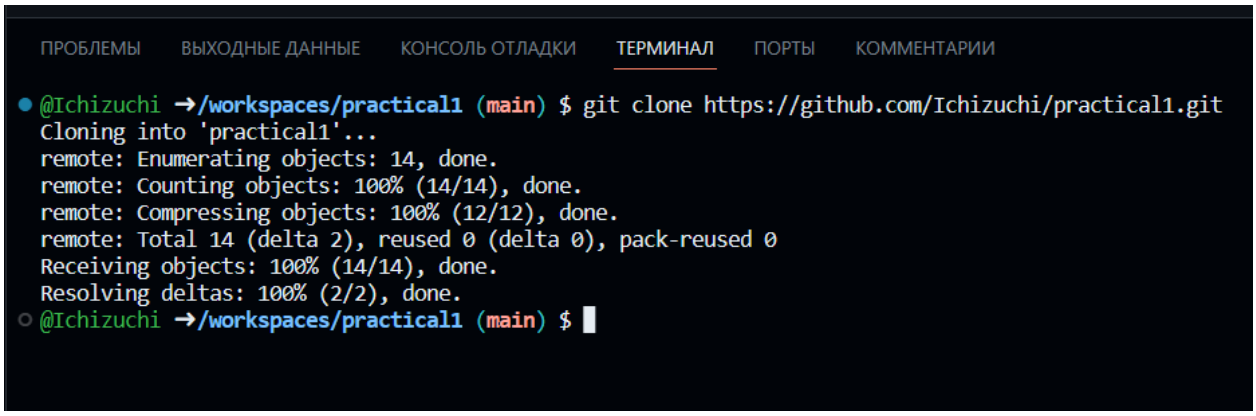

Рисунок 7. Клонирование репозитория на локальный компьютер

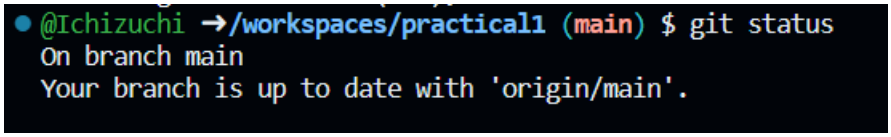

Рисунок 8. Успешное клонирование

3. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки.

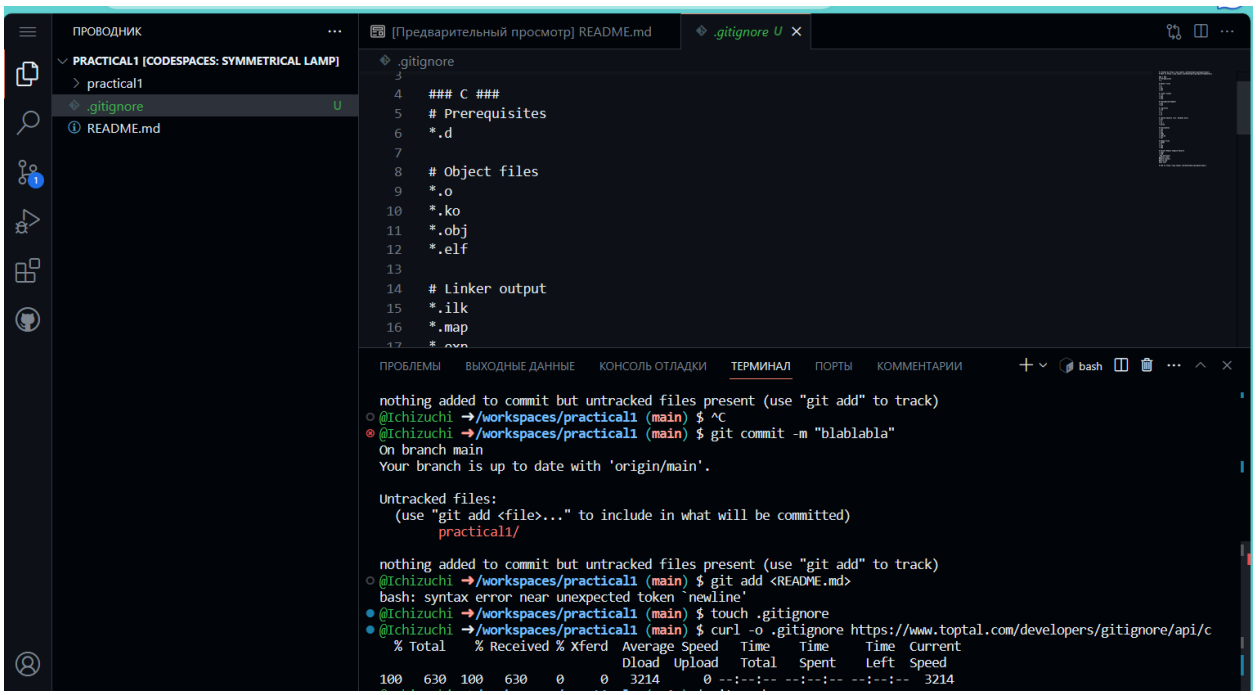

Рисунок 9. Добавление изменений в .gitignore

4. Добавьте в файл README.md информацию о группе и ФИО студента, выполняющего лабораторную работу.

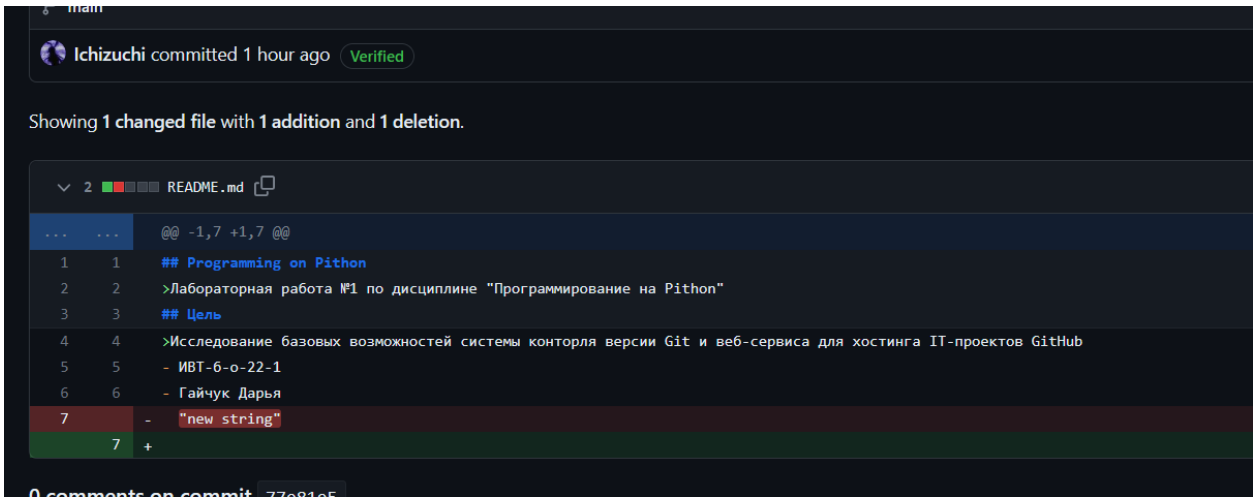

Рисунок 10. Добавление ФИО и вида работы

5. Напишите небольшую программу на выбранном Вами языке программирования. Фиксируйте изменения при написании программы в локальном репозитории. Должно быть сделано не менее 7 коммитов.

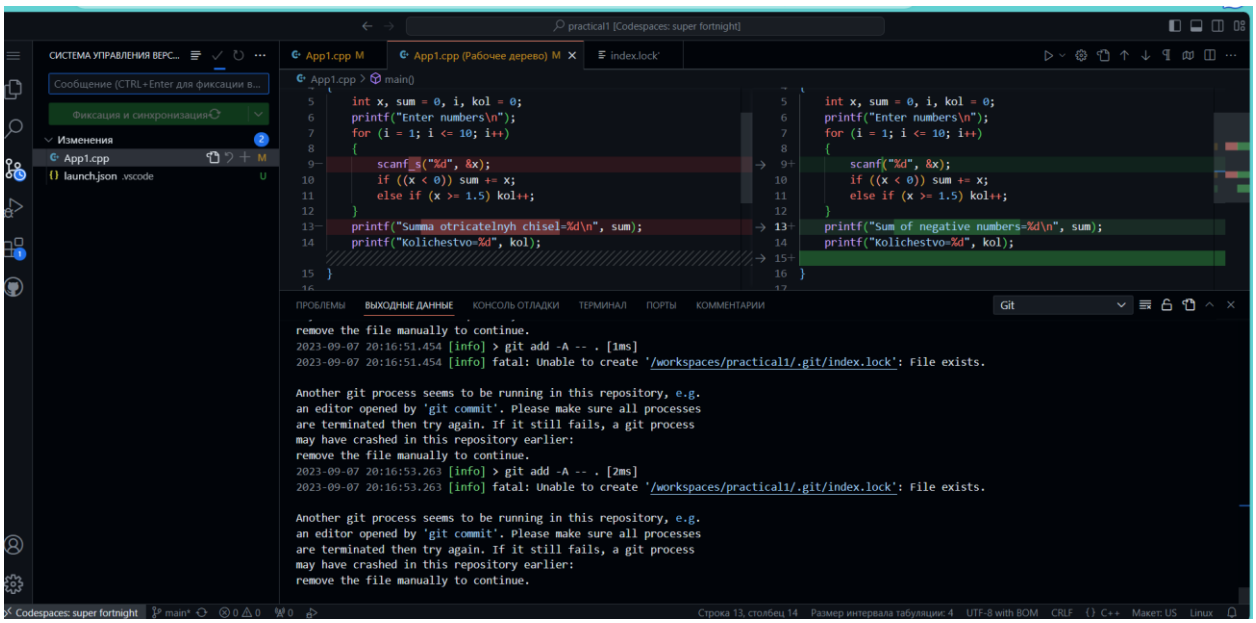

Рисунок 11. Созданная программа и изменения

Контрольные вопросы

1. Что такое СКВ и каково ее назначение?

Система контроля версий — это программный инструмент, который помогает отслеживать изменения в файлах и каталогах с течением времени. Его цель - управлять изменениями файлов и документировать их, позволяя нескольким людям совместно работать над проектом, вести историю изменений и легко возвращаться к предыдущим состояниям при необходимости.

2. В чем недостатки локальных и централизованных СКВ?

Локальный VCS предназначен для индивидуального использования, что затрудняет одновременную работу нескольких пользователей над одним и тем же проектом.

Отсутствие резервного копирования: Данные в локальном VCS могут быть легко потеряны в случае аппаратного сбоя или повреждения данных.

Централизованный VCS может стать узким местом или единой точкой отказа, если центральный сервер выйдет из строя.

Производительность может быть ниже для распределенных команд, в зависимости от скорости сети и нагрузки на сервер.

3. К какой СКВ относится Git?

Она относится к категории распределенных систем контроля версий (DVCS).

4. В чем концептуальное отличие Git от других СКВ?

Ключевое концептуальное отличие заключается в том, что Git — это распределенная система управления версиями, в то время как многие другие виртуальные машины централизованы. В Git каждый разработчик имеет полную копию репозитория на своем локальном компьютере, что обеспечивает автономную работу, более быстрое ветвление и слияние, а также большую гибкость при совместной разработке. Централизованные виртуальные машины, с другой стороны, полагаются на центральный сервер, где разработчики проверяют код, что может быть медленнее и менее гибким.

5. Как обеспечивается целостность хранимых данных в Git?

Git обеспечивает целостность хранимых данных за счет использования криптографического хэширования. Каждая часть данных в Git, включая файлы и историю коммитов, идентифицируется уникальным хэшем SHA1. Если какая-либо часть данных изменяется, хэш изменяется, предупреждая Git о потенциальной проблеме целостности.

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

Файлы в Git могут находиться в трех основных состояниях:

Измененные: файлы, которые были изменены, но еще не зафиксированы.

Поэтапные: Файлы, которые были помечены для включения в следующую фиксацию.

Зафиксированные: файлы, которые надежно хранятся в репозитории Git.

Эти состояния связаны в том смысле, что изменения вносятся в файлы в состоянии "Изменено", затем файлы "поэтапно" включаются в следующую фиксацию, и, наконец, выполняется фиксация для перемещения изменений в состояние "Зафиксировано".

7. Что такое профиль пользователя в GitHub?

Профиль пользователя в GitHub — это персональная страница, которая предоставляет информацию о пользователе GitHub. Обычно он включает в себя такие сведения, как имя пользователя, фотография профиля, биография и ссылки на репозитории, в которых он участвует. Пользователи могут настраивать свои профили, чтобы демонстрировать свою работу и вклад на GitHub.

8. Какие бывают репозитории в GitHub?

Репозиторий Git бывает локальный и удалённый.

9. Укажите основные этапы модели работы с GitHub.

Установка Git; добавление имени, фамилии и адреса электронной почты; ввод определенных команд для Git; загрузка изменений в состояние (staged); добавление коммита; отправка в репозиторий на сервис GitHub.

10.Как осуществляется первоначальная настройка Git после установки?

Добавление имени, фамилии и адреса электронной почты: git config --global user.name – указывает ваше имя, фамилию. git config --global user.email – указывает вашу электронную почту. git init – создает новый репозиторий Git.

11.Опишите этапы создания репозитория в GitHub.

Ввод имени для репозитория, добавление описания проекта (выборочно), выбор приватности данного репозитория, добавление дополнительных файлов, как README.md и .gitignore.

12.Какие типы лицензий поддерживаются GitHub при создании репозитория?

Academic Free License v3.0; Boost Software License 1.0; Creative Commons license family; Eclipse Public License 1.0; ISC; MIT и многие другие.

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

С помощью команд git clone/git push. Чтобы упростить устранение конфликтов слияния, добавление или удаление файлов и отправку больших фиксаций.

14.Как проверить состояние локального репозитория Git?

Используйте команду git status , чтобы проверить текущее состояние репозитория.

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?

Репозиторий будет находиться в начальном состоянии, то есть Git напишет о том, что вы находитесь на основной ветке и то что вам нечего коммитить, рабочая ветка пуста.

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. Примечание: описание необходимо начать с команды git clone . git clone; git remote –v; git pull;

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

Bitbucket - веб-сервис для хостинга проектов и их совместной разработки, основанный на системах контроля версий Mercurial и Git. GitHub вращается вокруг общедоступного кода, тогда как Bitbucket предназначен в основном для частных проектов. Это основное различие между GitHub и Bitbucket. GitHub — ведущее сообщество разработчиков открытого исходного кода, тогда как Bitbucket в основном используется предприятиями и предприятиями.

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