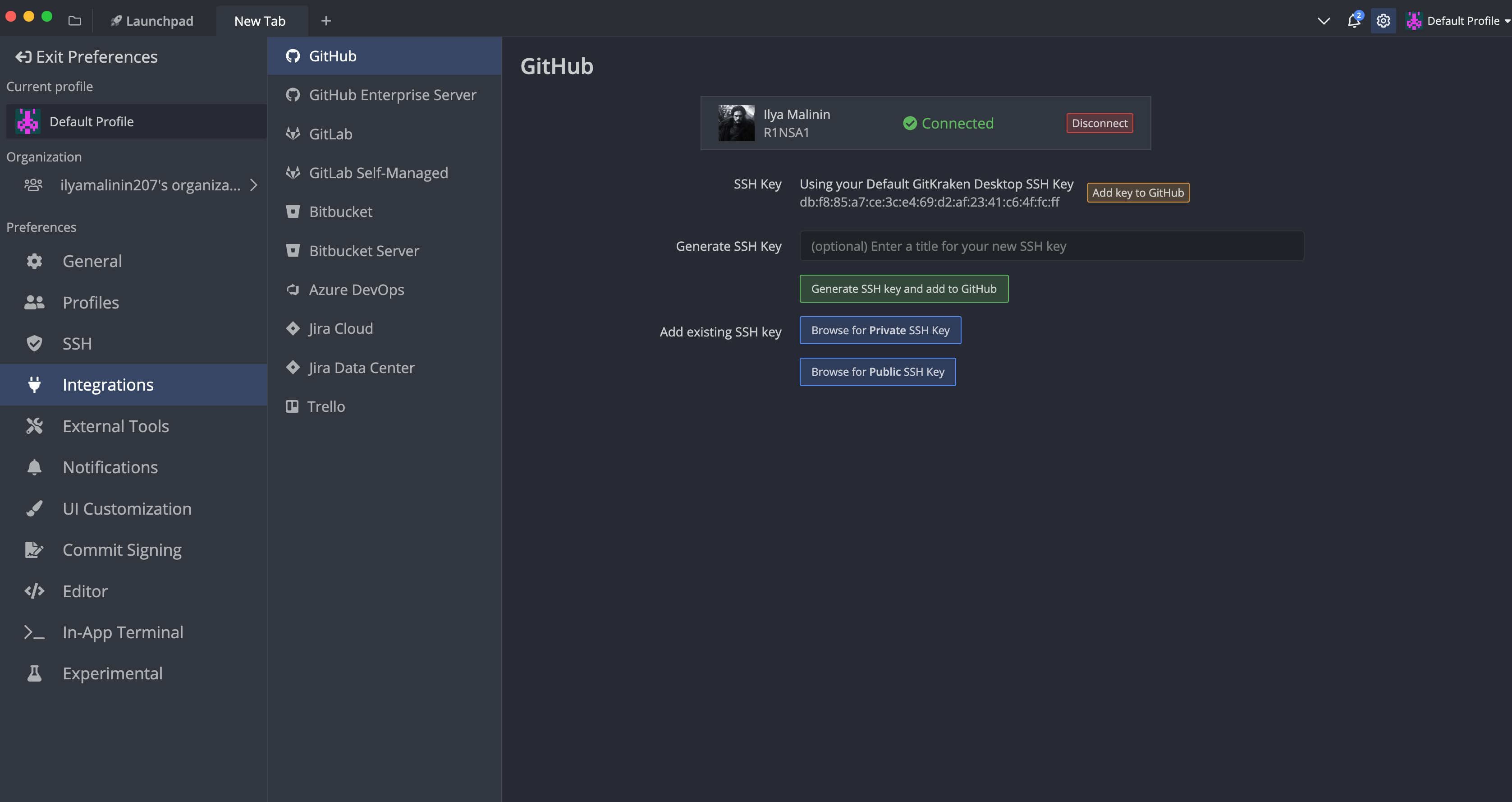
**Малинин И. А.**

**Лабораторная работа №1:**

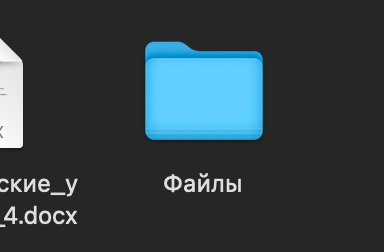
**Упражнение 1. Работа с системой контроля версий Git.**

* 1. Установить на компьютер графический клиент Git.

Я буду использовать GitKraken:



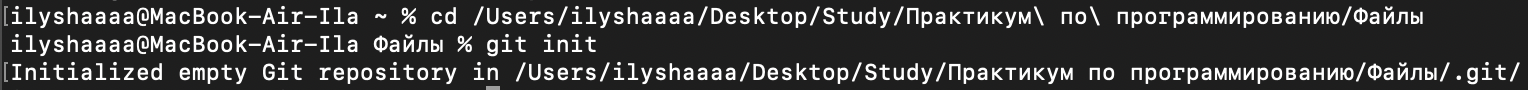
* 1. Создайте в своей домашней папке (или в любой другой на ваш выбор) каталог, который будет содержать файлы нового программного проекта.



* 1. Выберите тематику программы, которую собираетесь написать.

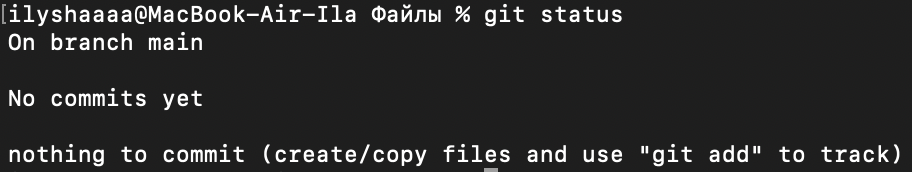
Я возьму тематику программы, связанную с управлением задачами и проектами. Это может быть простая программа для отслеживания задач, которая позволит пользователям создавать, редактировать и отмечать выполнение задач.

* 1. Инициализируйте в этой директории репозиторий гит при помощи команды git init. Обратите внимание на появление в этой папке скрытой подпапки с названием .git. Если вы ее не видите, то скорее всего, у вас отключено отображение скрытых папок.

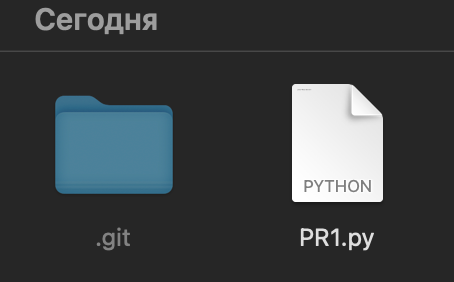
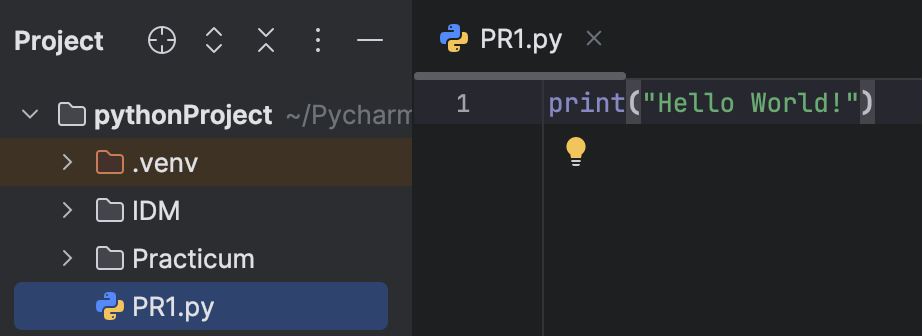


* 1. Выполните в репозитории команду git status. Проинтерпретируйте полученное сообщение.

Нахожусь на ветке main. Нет коммитов.

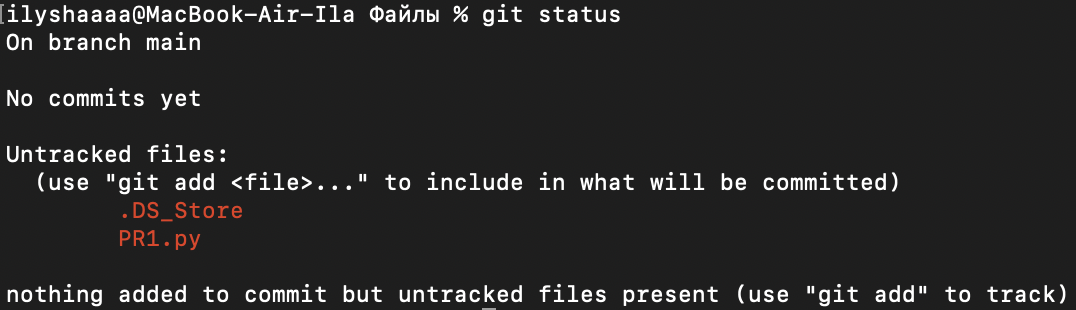


* 1. Создайте файл для исходного текста программы. Выполните команду git status.

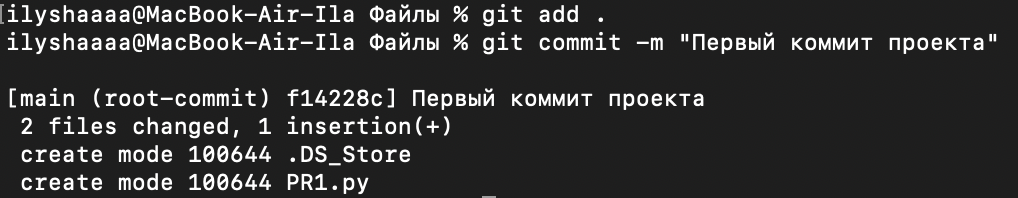


* 1. Добавьте созданный файл под версионный контроль при помощи команды git add. Еще раз выполните git status.

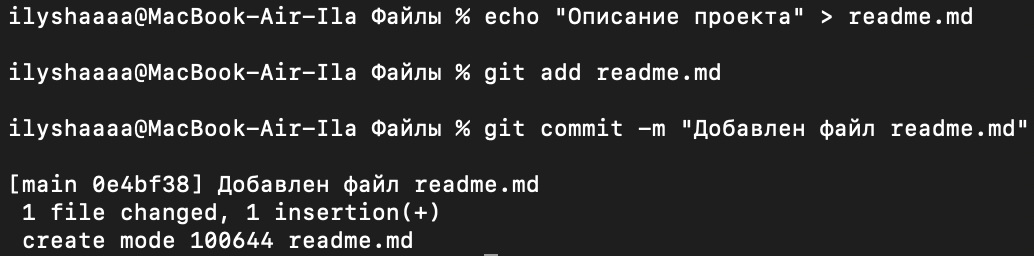
Теперь файл PR1.py виден тут:

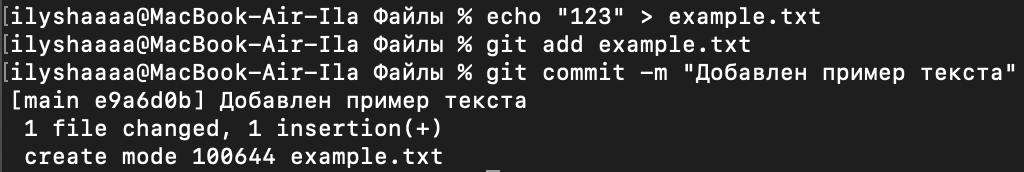


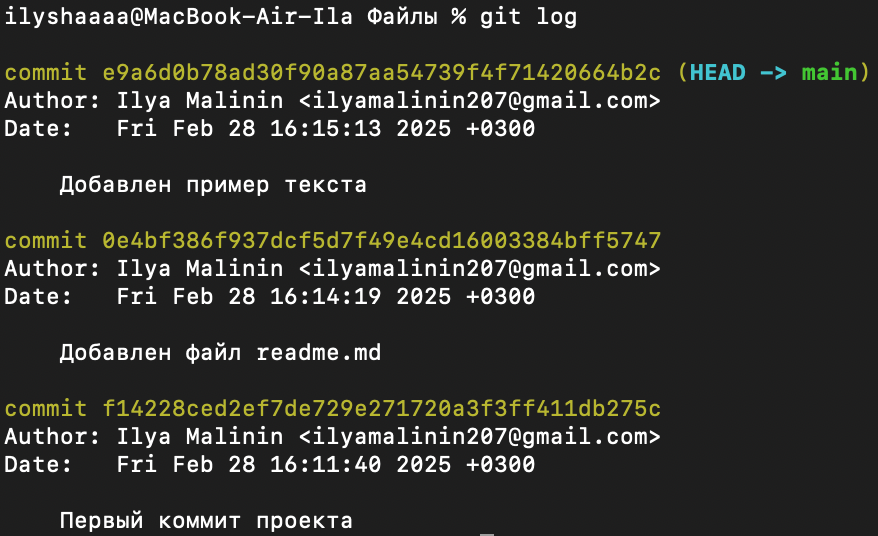
* 1. Когда все файлы готовы к сохранению, сделаем наш первый коммит — зафиксируем все сделанные изменения в «боевой версии». Делается это командой git commit c опцией -m. После -m идёт название коммита в кавычках.



* 1. Сделайте еще несколько коммитов. Выполните команду git log для просмотра истории коммитов.

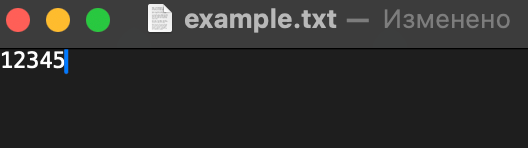


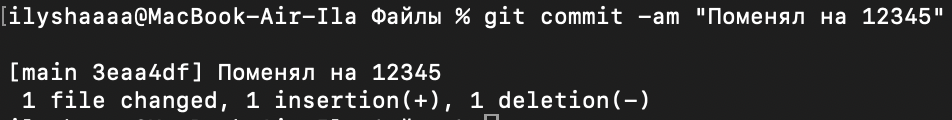




* 1. Сделайте так, чтобы при коммите измененные файлы автоматически добавлялись в коммит.

Изменяю вручную файл с 123 на 12345:

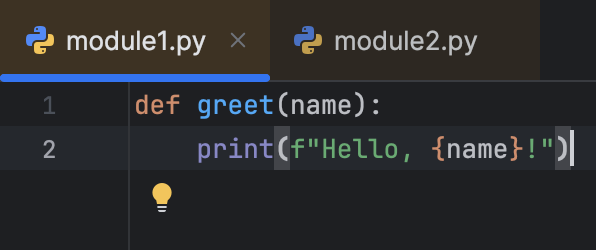


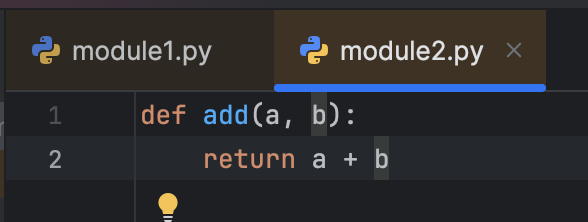


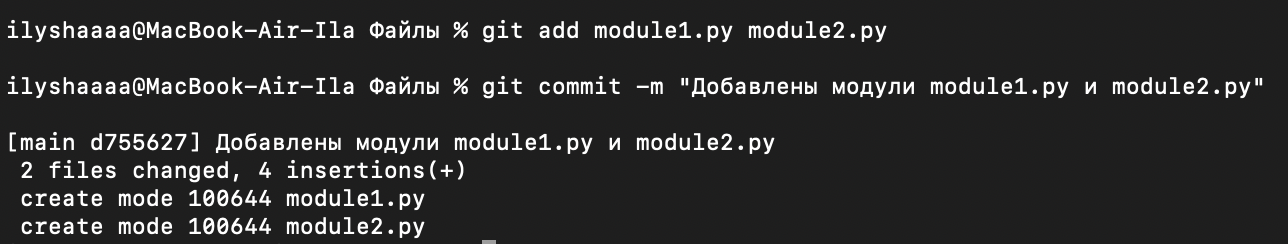
* 1. Добавьте еще несколько файлов с исходным текстом программы.



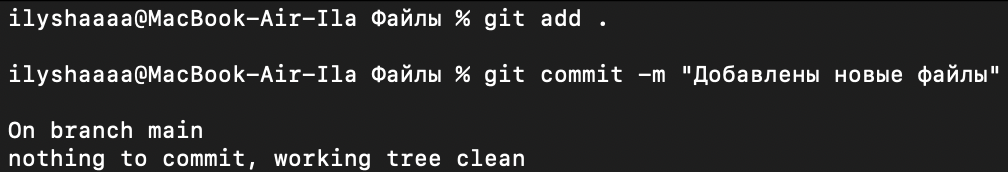
Добавляем некоторый код в данные файлы:





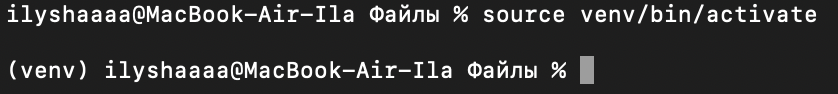


* 1. Добавьте все новые файлы под версионный контроль одной командой.

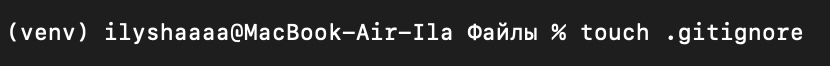


* 1. Инициализируйте в рабочей директории виртуальное окружение

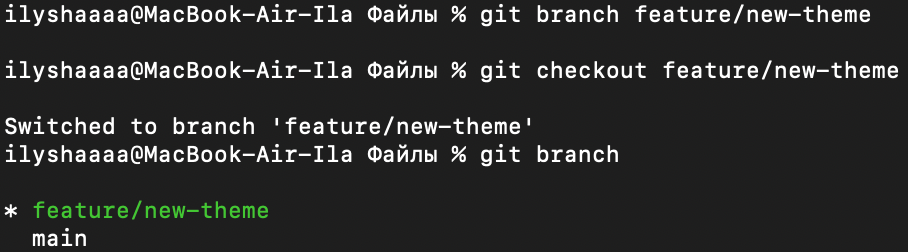




* 1. Добавьте созданную служебную папку в файл .gitignore. Проверьте, что они не добавляются в репозитории при добавлении новых файлов с исходным кодом.

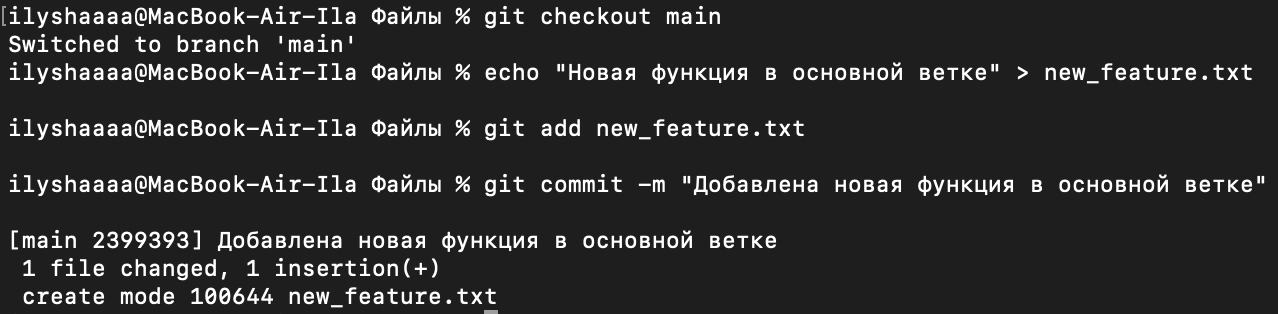


* 1. Создайте новую тематическую ветку git branch. Перейдите в нее с помощью git checkout. Выведите на экран список всех веток.

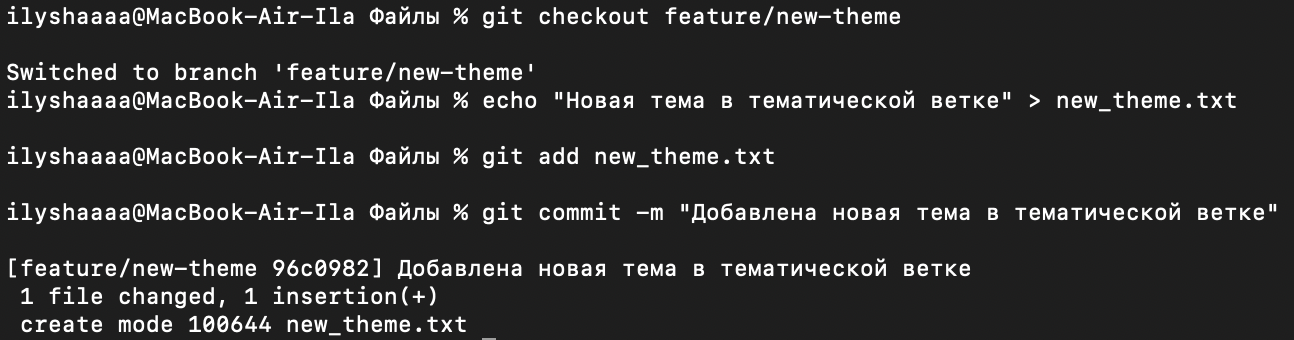


* 1. Сделайте несколько коммитов в основную и тематическую ветки.

В основную:



В тематическую:



* 1. Слейте изменения в основную ветку с помощью git merge. Если

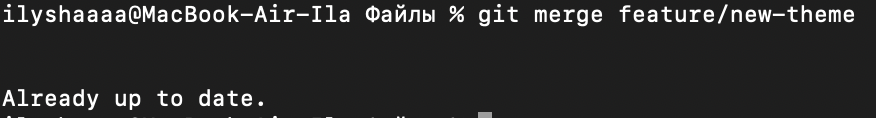
произошел конфликт слияния, разрешите его и завершите слияние с помощью git commit.

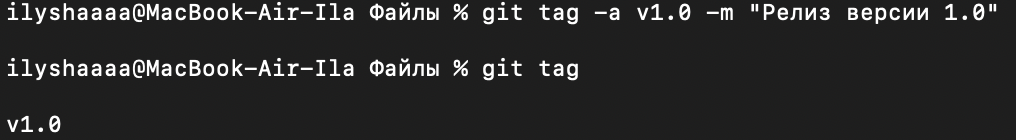






* 1. При получении в процессе разработки программы в стабильно работающем состоянии, слейте это состояние в основную ветку и добавьте к коммиту слияния пометку с номером релиза.



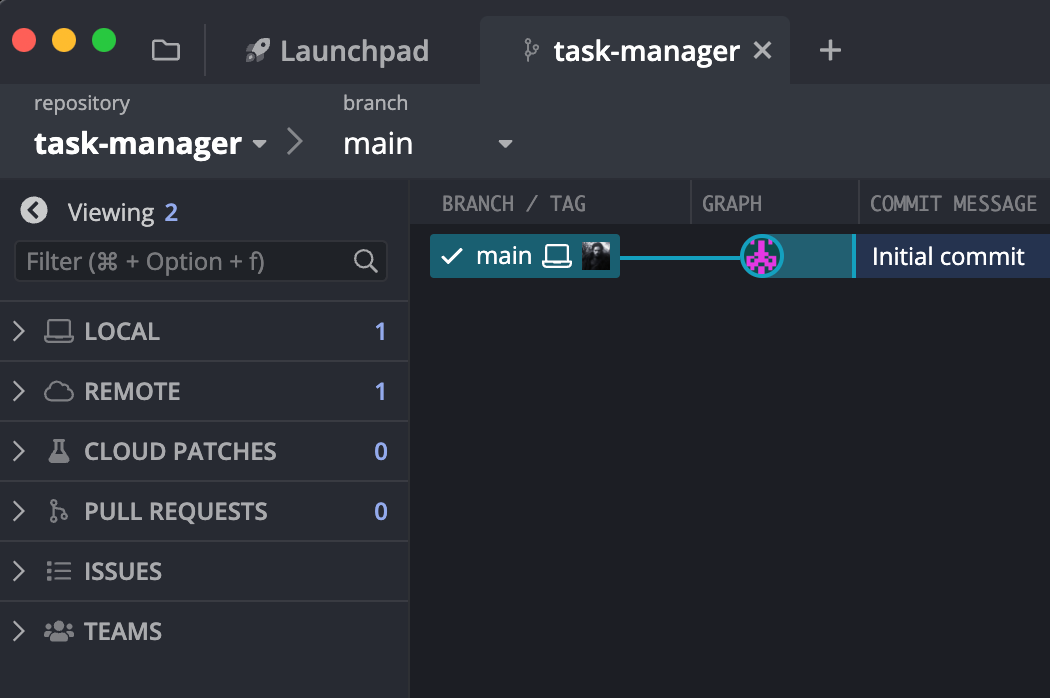




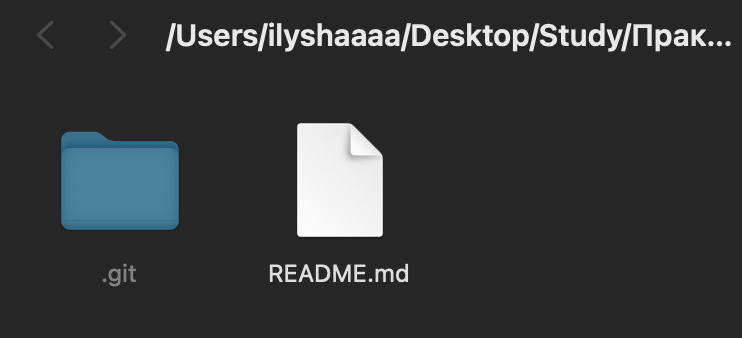
**Упражнение 2. Работа GUI git клиент GitKraken.**

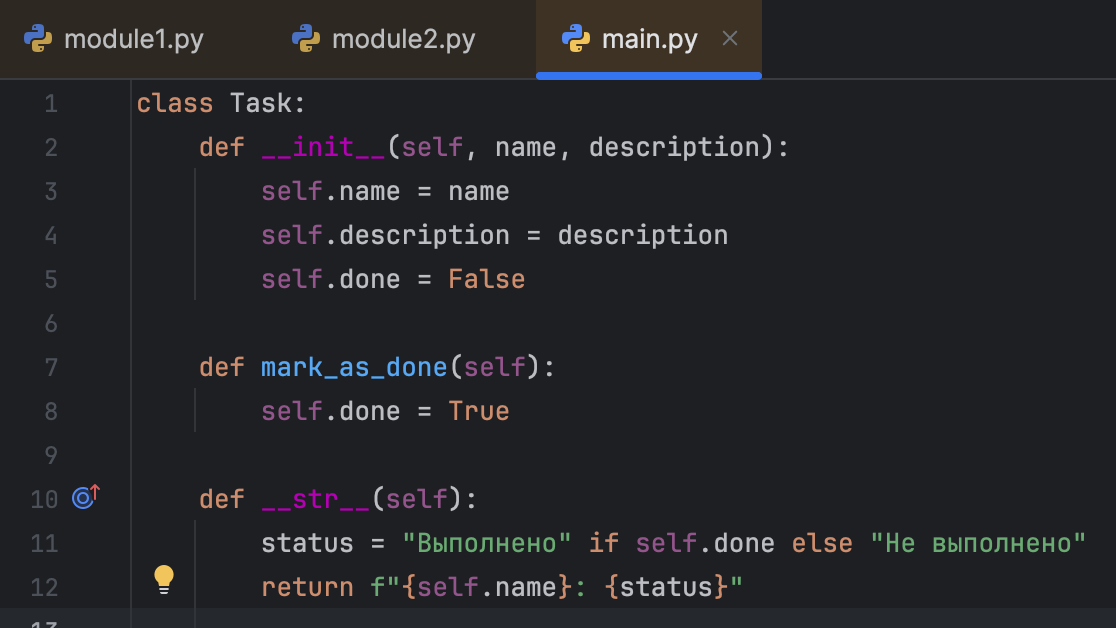
1. Выберите тематику программы, которую собираетесь написать. Создайте для нее рабочую директорию.

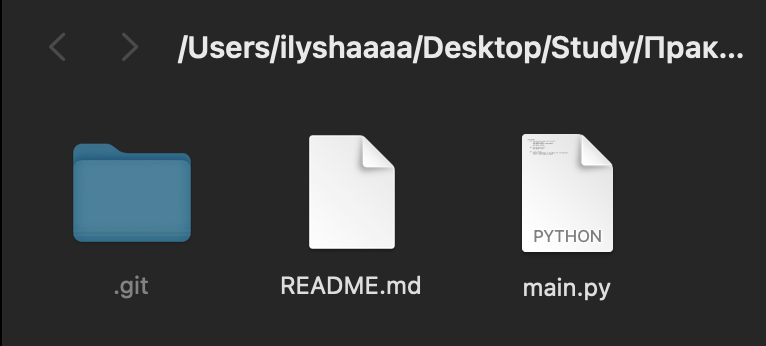
Я опять возьму “Управление задачами и проектами”. Создал директорию task-manager:



1. Инициализируйте в рабочей директории репозиторий.
2. Создайте файл для исходного текста программы.

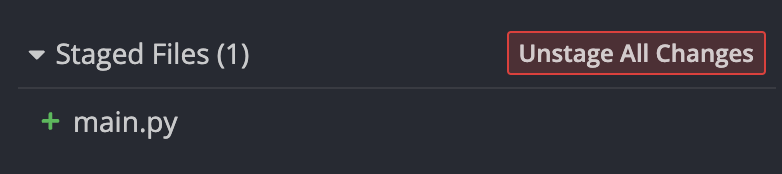


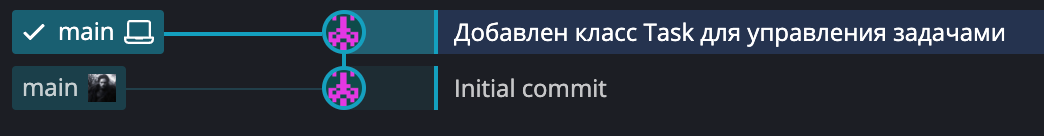




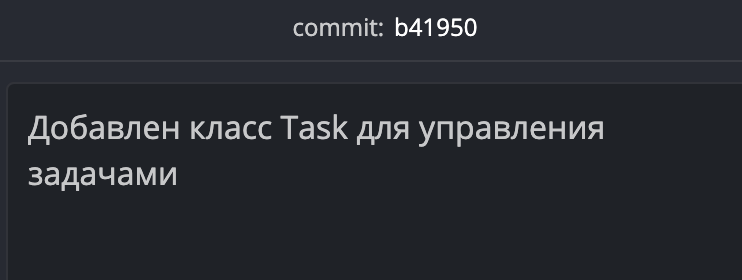
1. Добавьте созданный файл под версионный контроль.

В GitKraken находим этот файл:



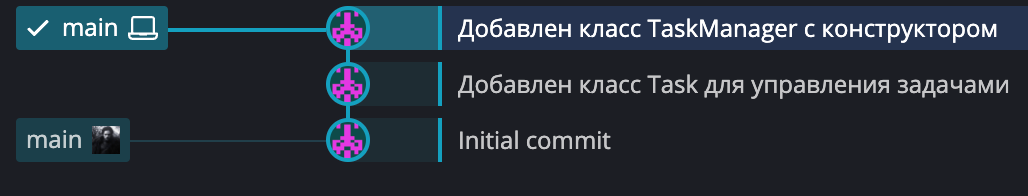


1. Сделайте начальный коммит.

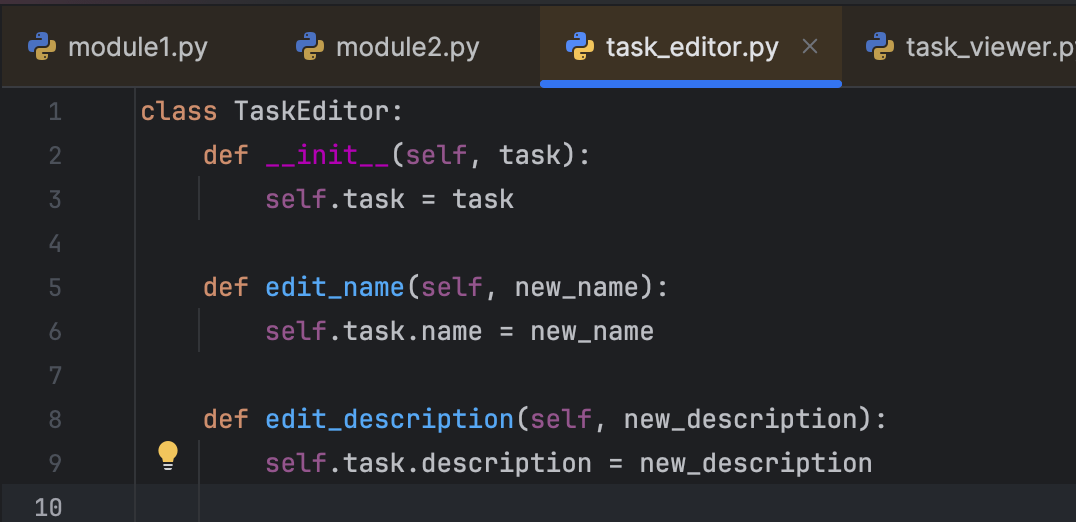


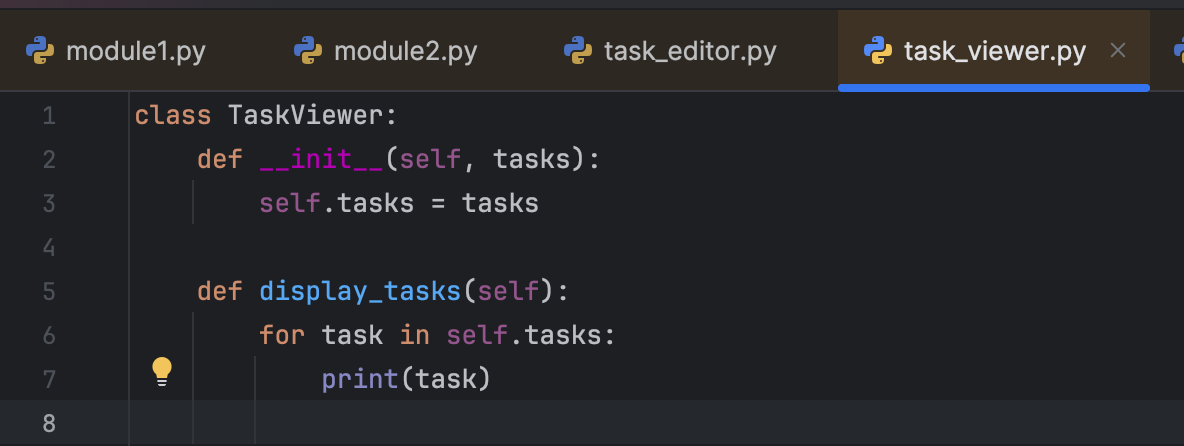
1. Сделайте еще несколько коммитов.

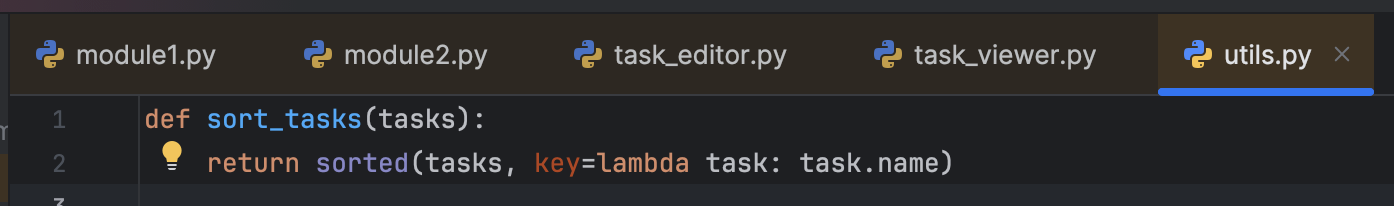
Добавляю класс в файл и создаю ещё коммит:

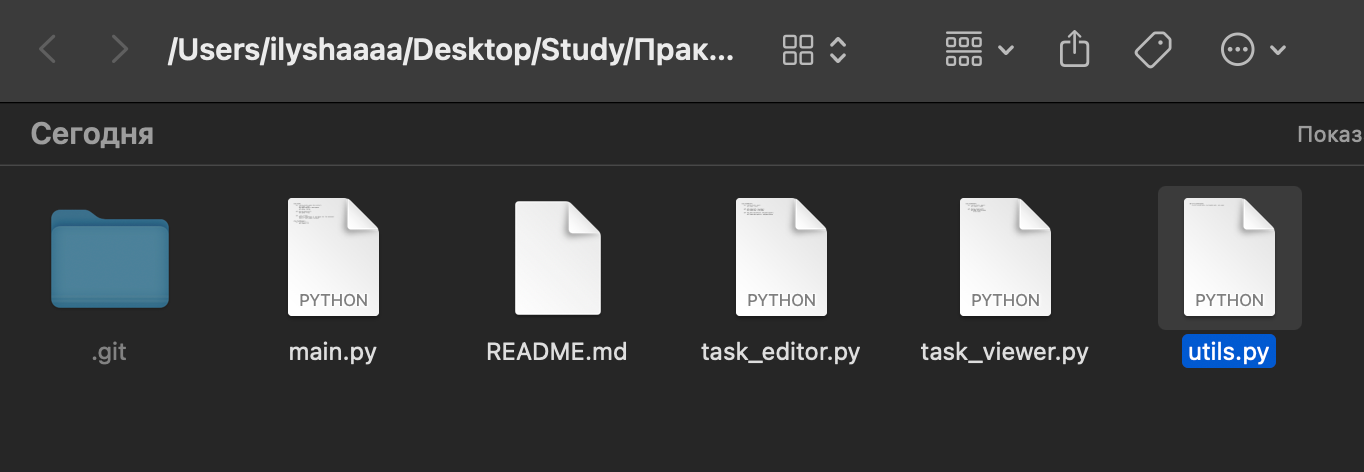


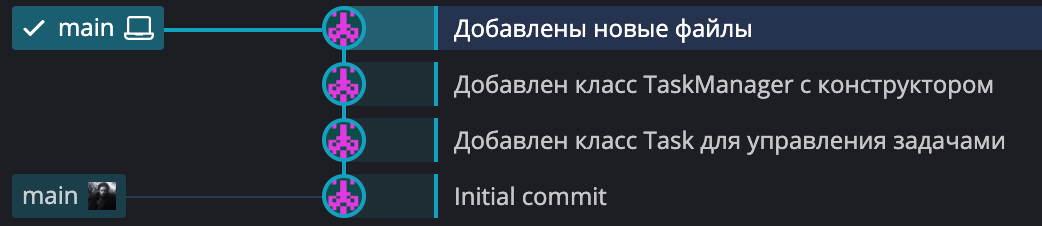
1. Добавьте еще несколько файлов с исходным текстом программы.



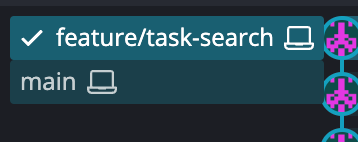








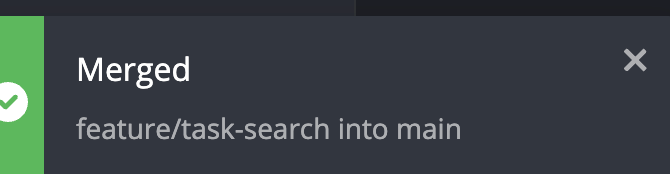
1. Создайте новую тематическую ветку.



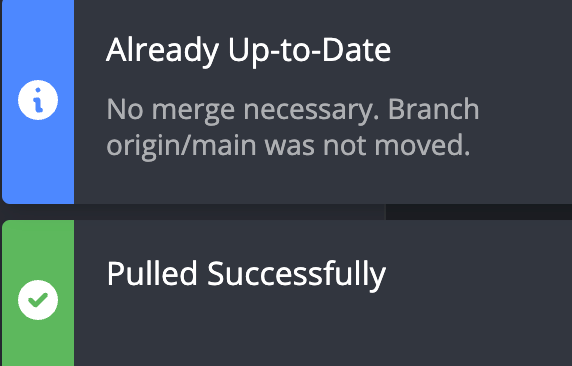
1. Сделайте несколько коммитов в основную и тематическую ветки.



1. Слейте изменения в основную ветку. Если произошел конфликт слияния, разрешите его и завершите слияние.



1. При получении в процессе разработки программы в стабильно работающем состоянии, слейте это состояние в основную ветку и добавьте к коммиту слияния пометку с номером релиза.



**Упражнение 3. Работа с удаленными репозиториями и GitHub.**

1. Зарегистрироваться на сайте github.com и создать новый пустой репозиторий



1. Создать в этом репозитории файл Readme

