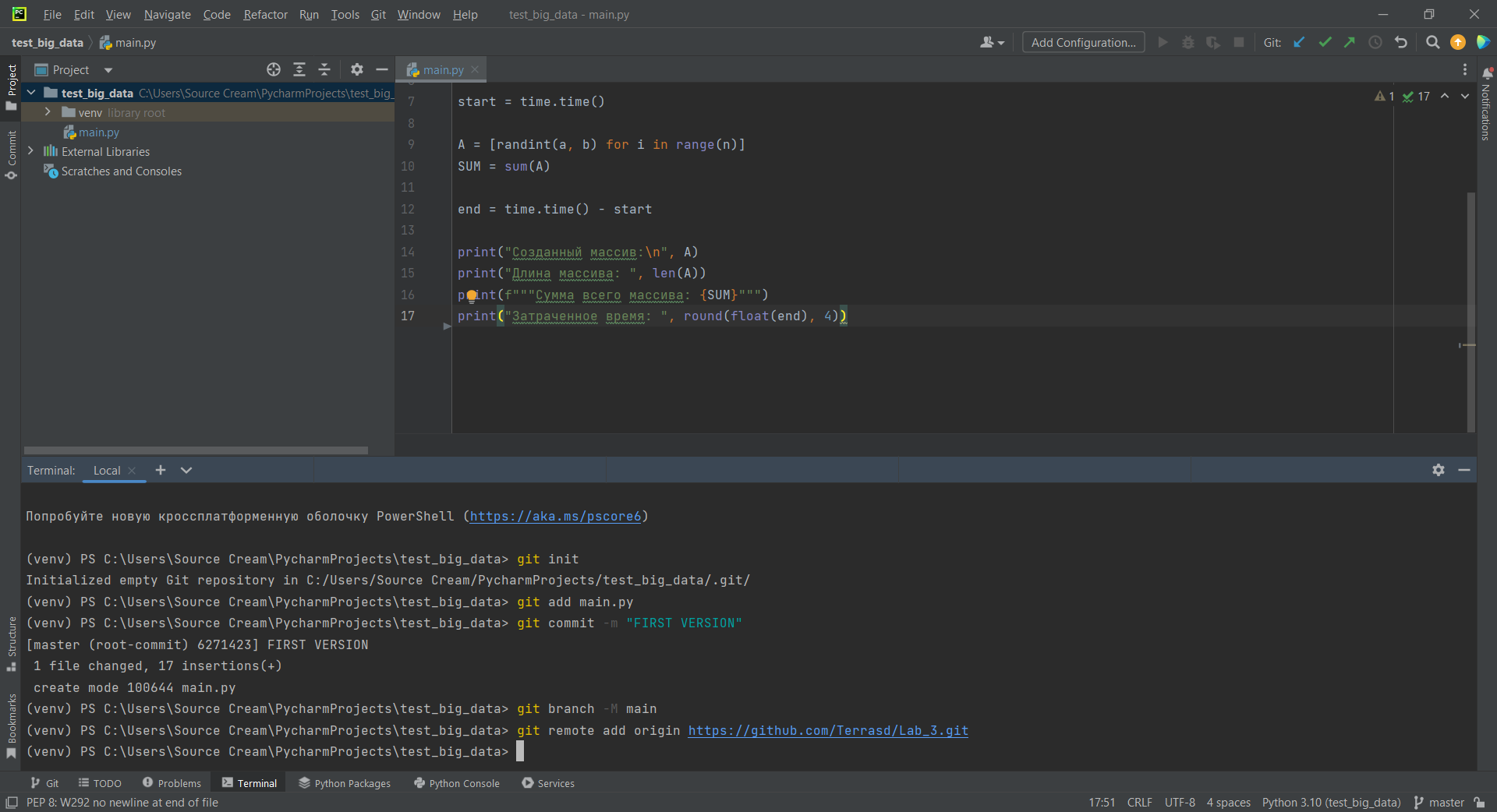
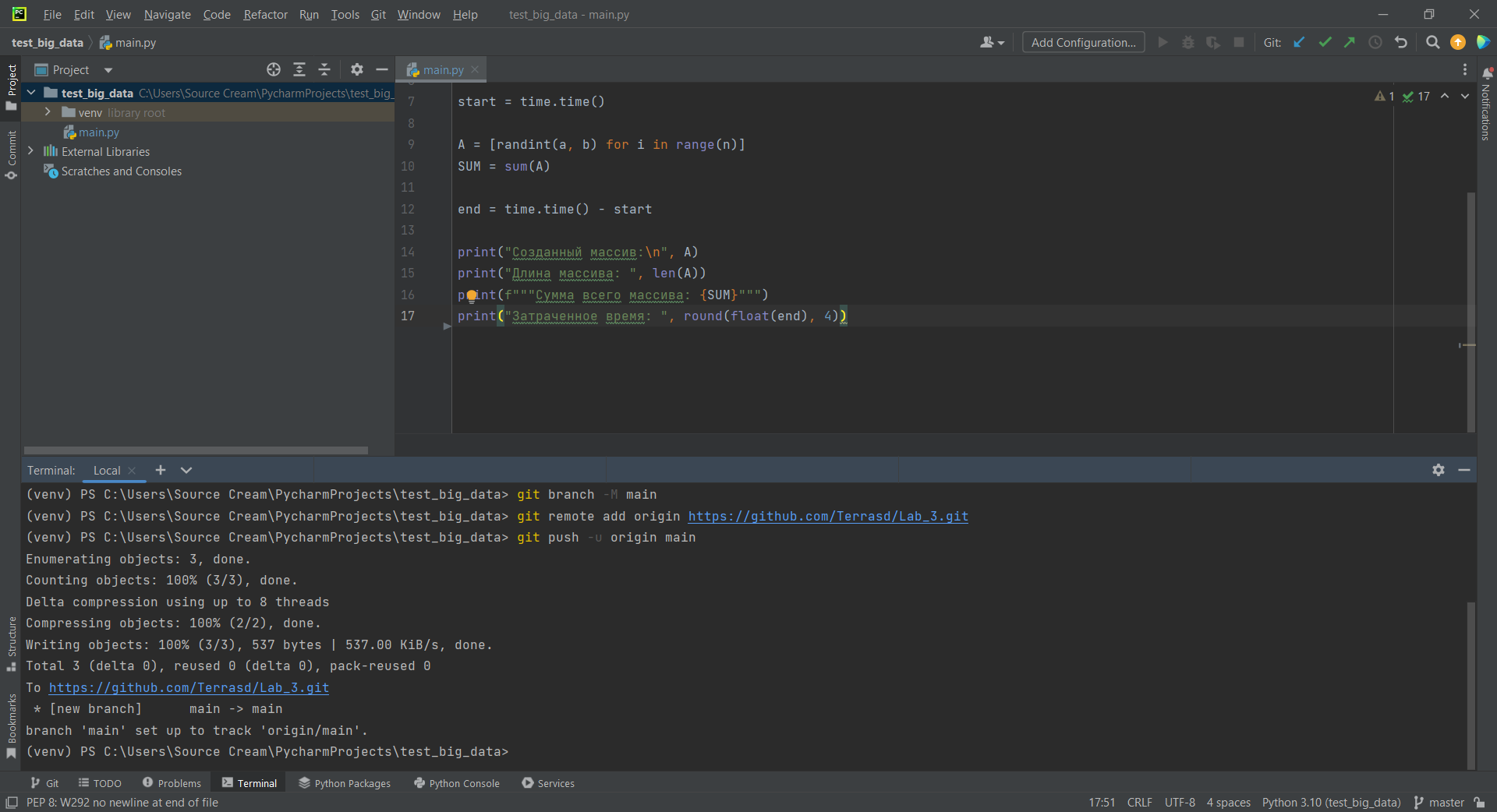
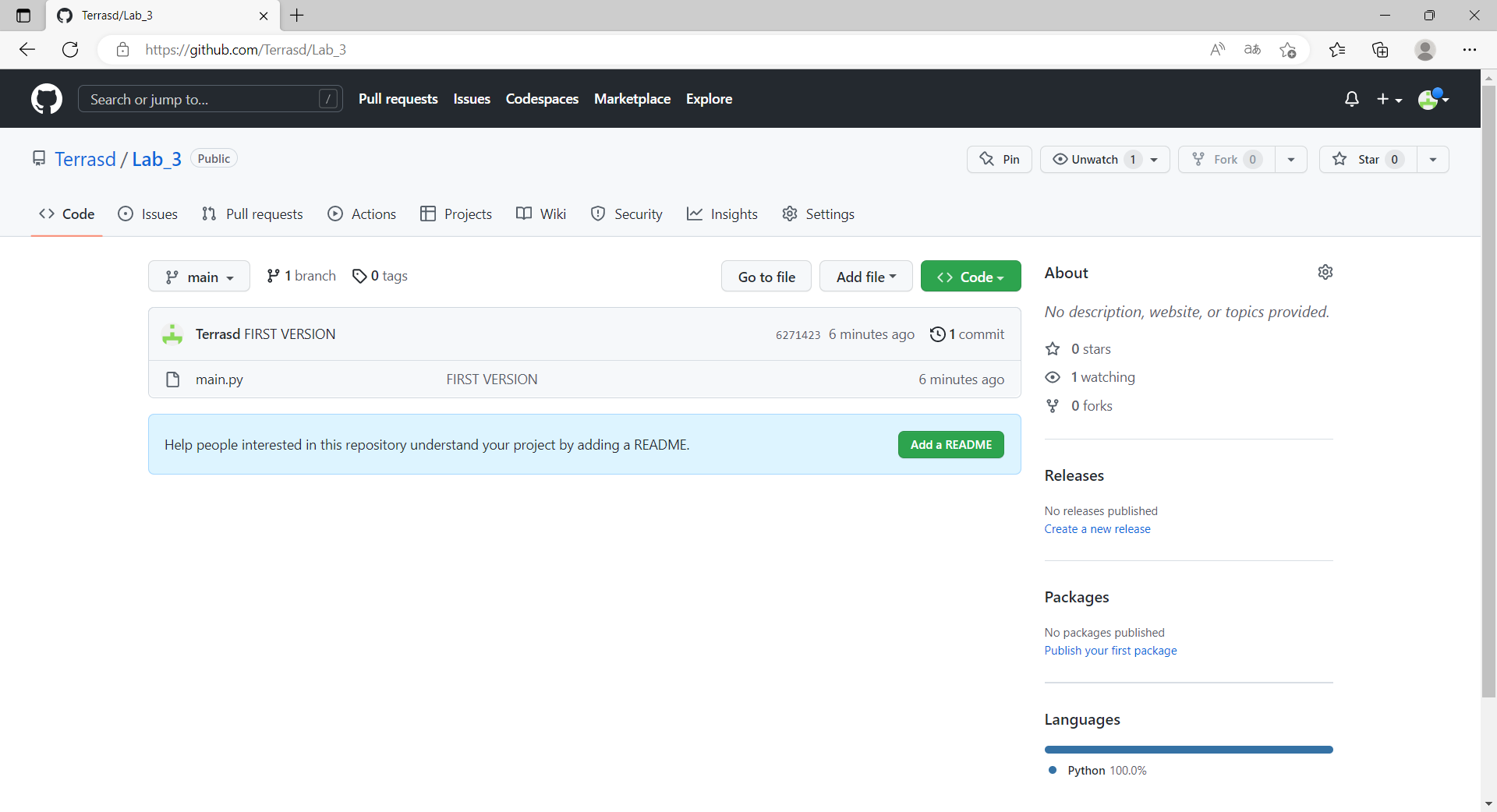
**3 Вариант**: Исправить в ходе работы не менее 5 коммитов, задать дополнительно 3-4 алиаса и использовать их.

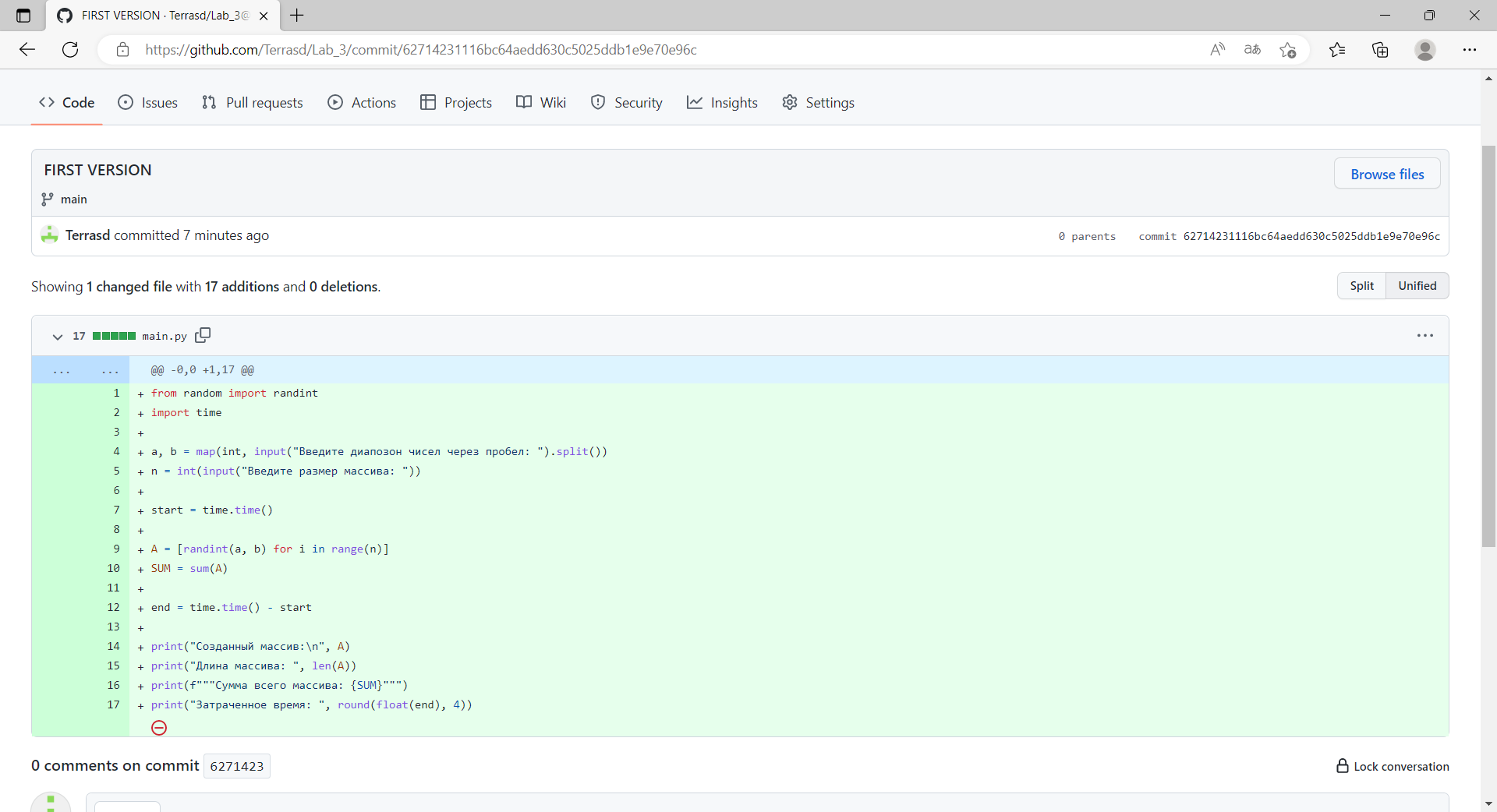
Для начала создал репозиторий в проекте, добавил первый коммит, создал новую ветку и загрузил всё в репозиторий на Github:



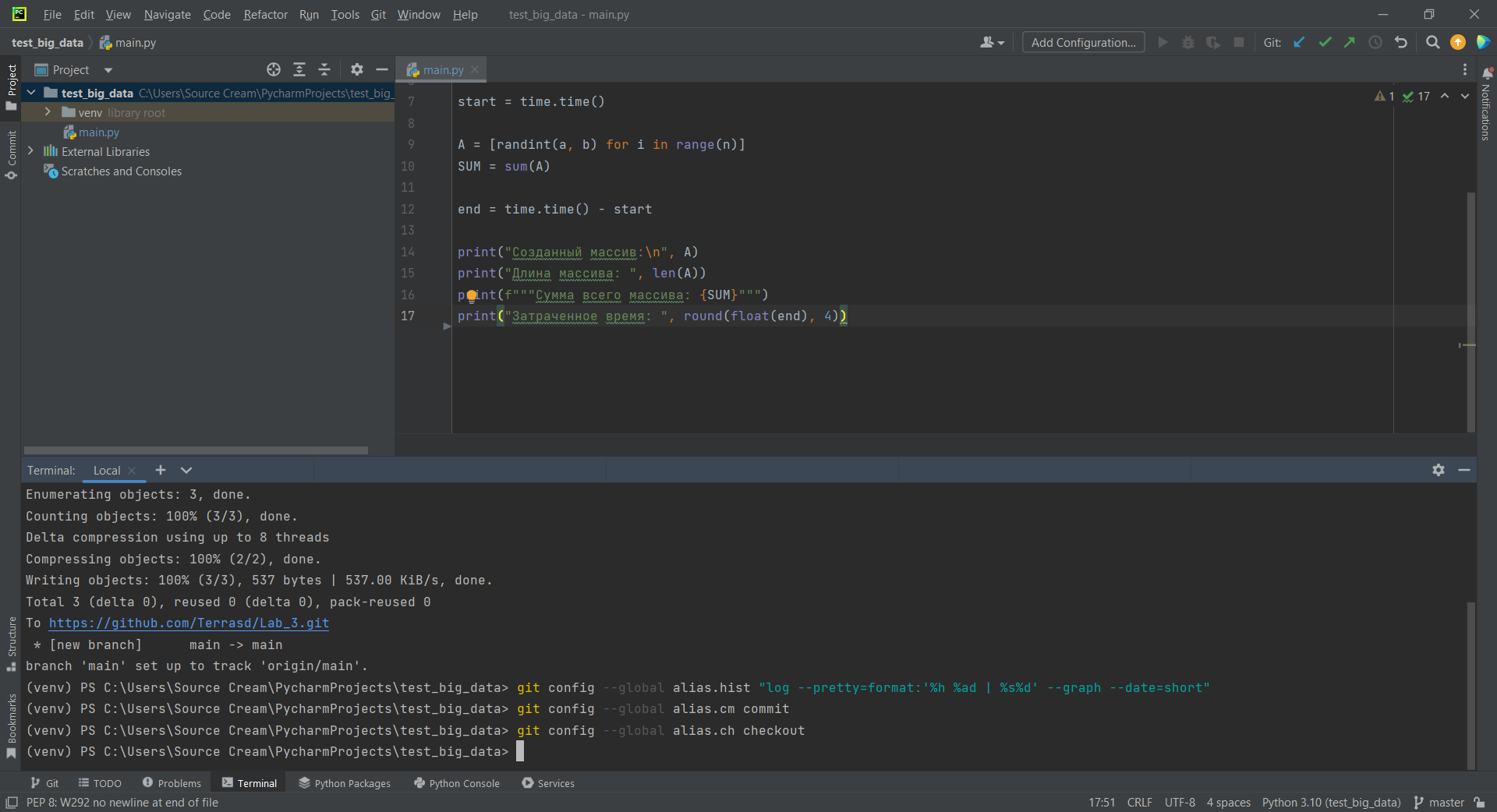


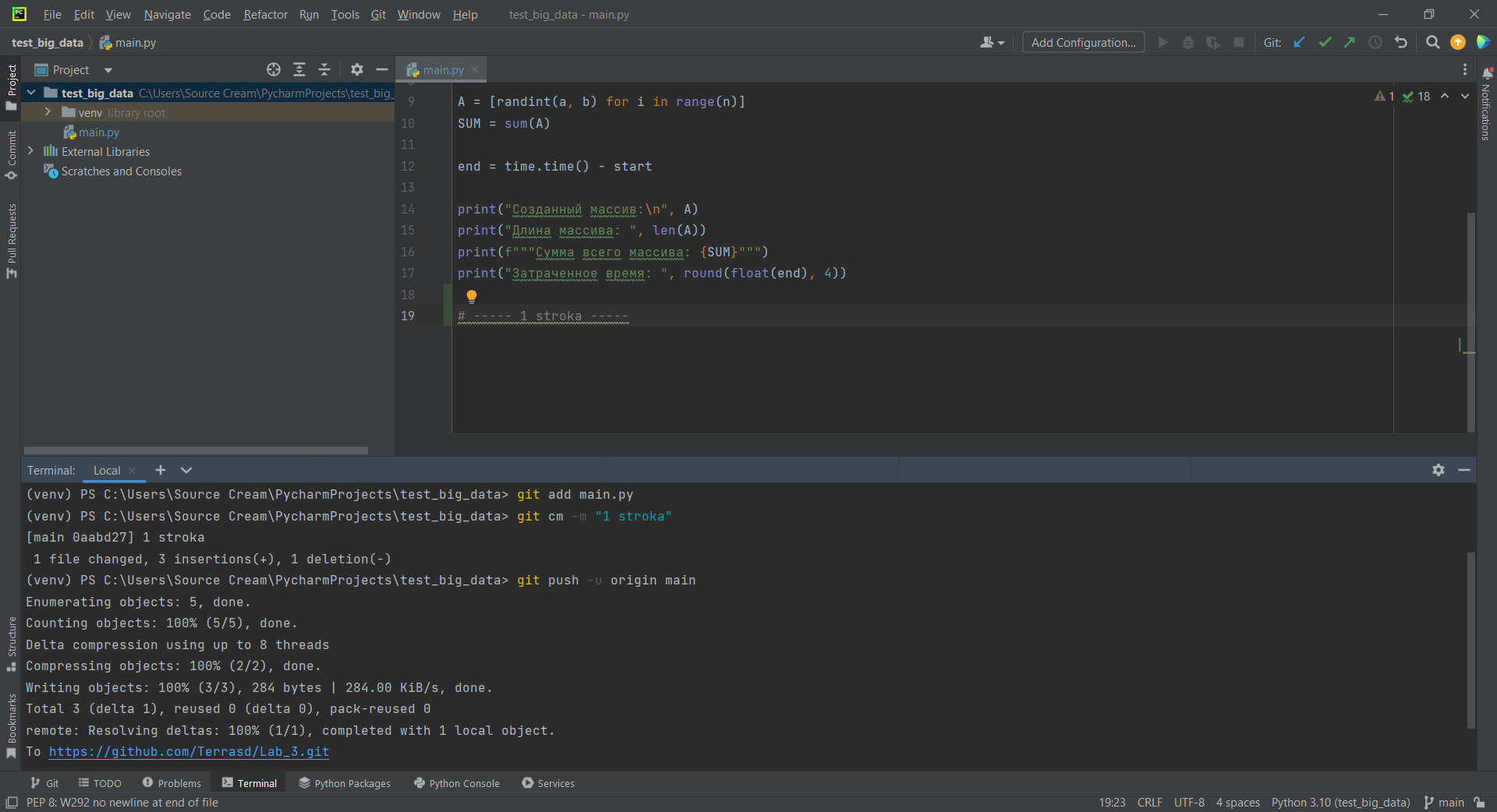
Проверил, что на Github всё появилось:



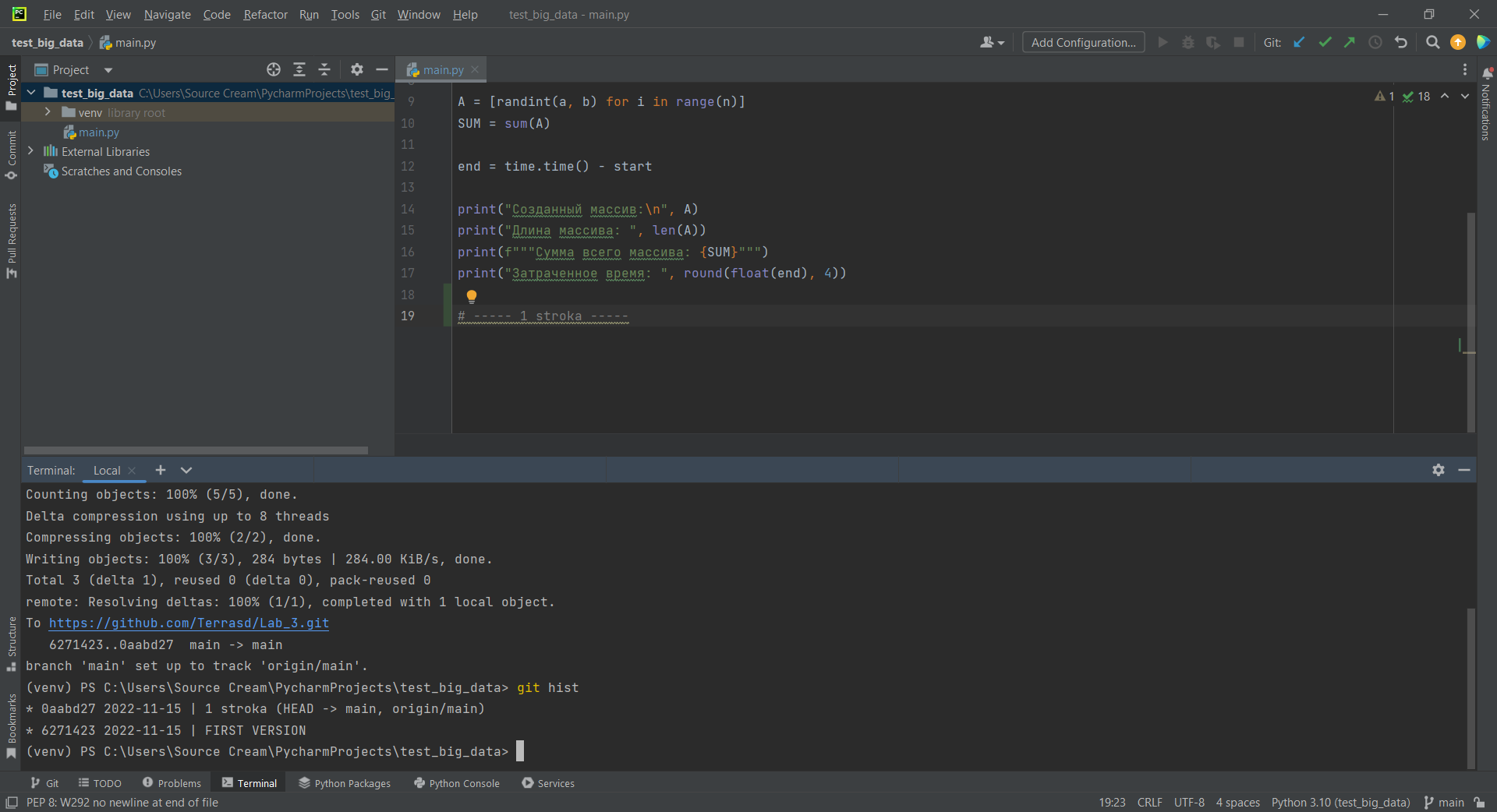


Далее создаю алиасы и начинаю несильно изменять программу, чтобы для наглядности показать применение изменений, добавляю коммиты и загружаю их на Github:

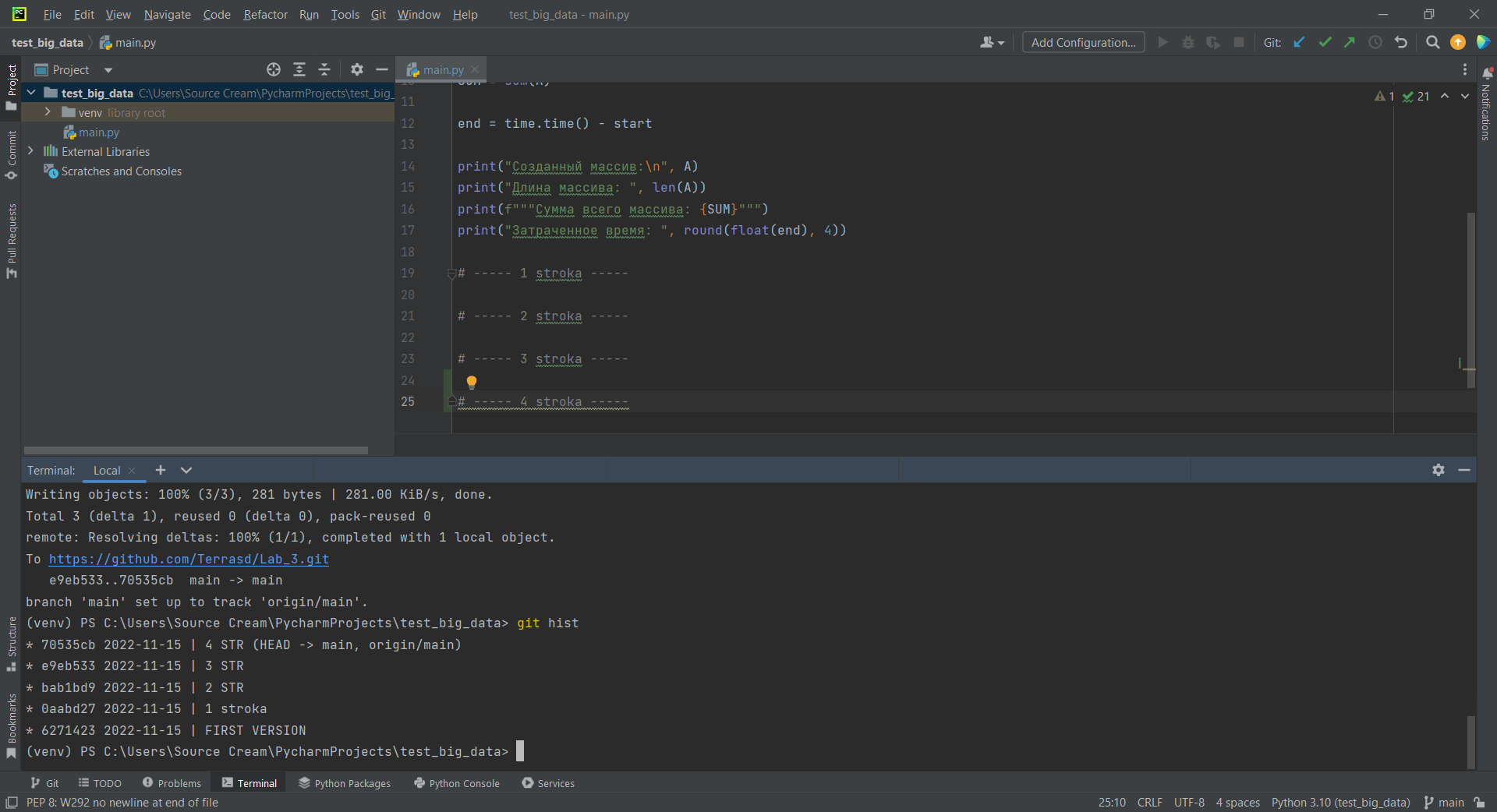




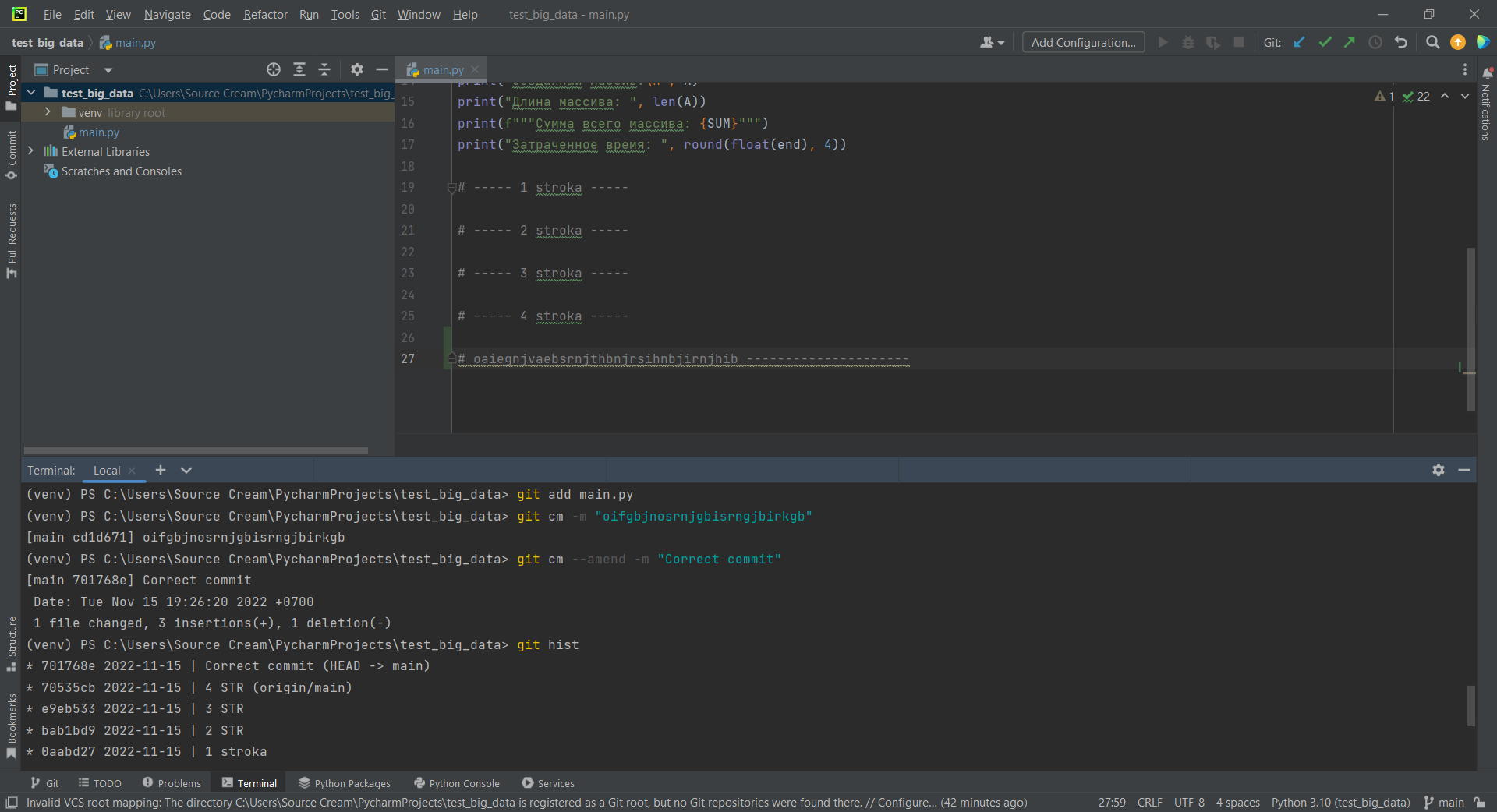
Здесь показал, что алиас hist работает корректно и наглядно показывает все коммиты файла:



Добавление последнего (пятого) коммита:



Здесь показано, как можно изменить коммит:



**Ответы на контрольные вопросы.**

1) Это система управления версиями с распределенной архитектурой.

2) Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение.

3) Коммит – это операция отправки изменений. Она выполняется тогда, когда все нужные изменения на данный момент внесены и об этом можно сообщить системе контроля версий.

4) Выполнить команду **git log**.

5) Выполнить команду **git init**.

6) Git – система управления версиями с распределенной архитектурой.

7) За создание новой ветки отвечает команда **git checkout -b НазваниеВетки**, а за слияние – **git merge НазваниеВетки**.

8) Благодаря «staging area» при коммите версий можно просмотреть все изменения и подправить, если было указано что-то не так перед отправкой на Github.

9) Выполнить команду **git diff НазваниеКоммита**.