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 №4**

**дисциплины «Основы программной инженерии»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Яблоновский Дмитрий Николаевич  2 курс, группа ПИЖ-б-о-22-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Богданов С.С., ассистент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** Основы языка Python.

**Цель работы:** исследование процесса установки и базовых возможностей языка Python версии 3.x.

**Порядок выполнения работы**

1. Создал репозиторий GitHub.

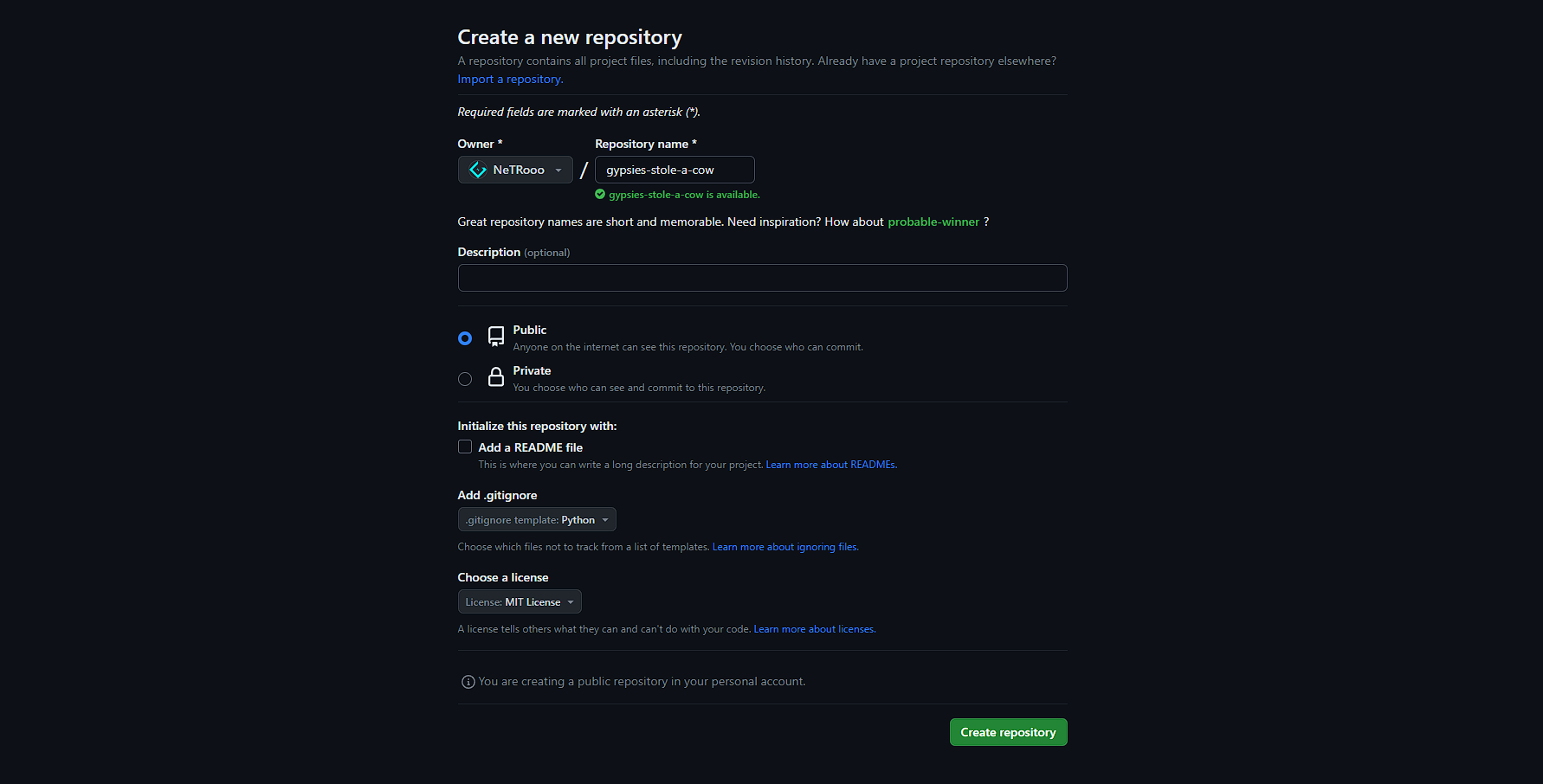


Рисунок 1- Создание репозитория

1. Проработал примеры из лабораторной работы.



Рисунок 2 – Выполнил клонирование созданного репозитория

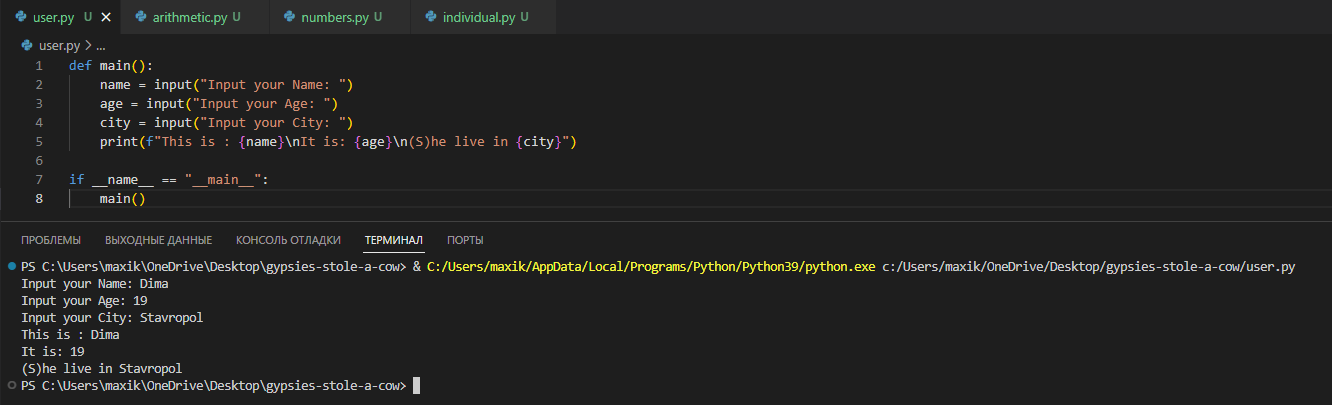


Рисунок 3 – Создал файл «user.py» и написал программу

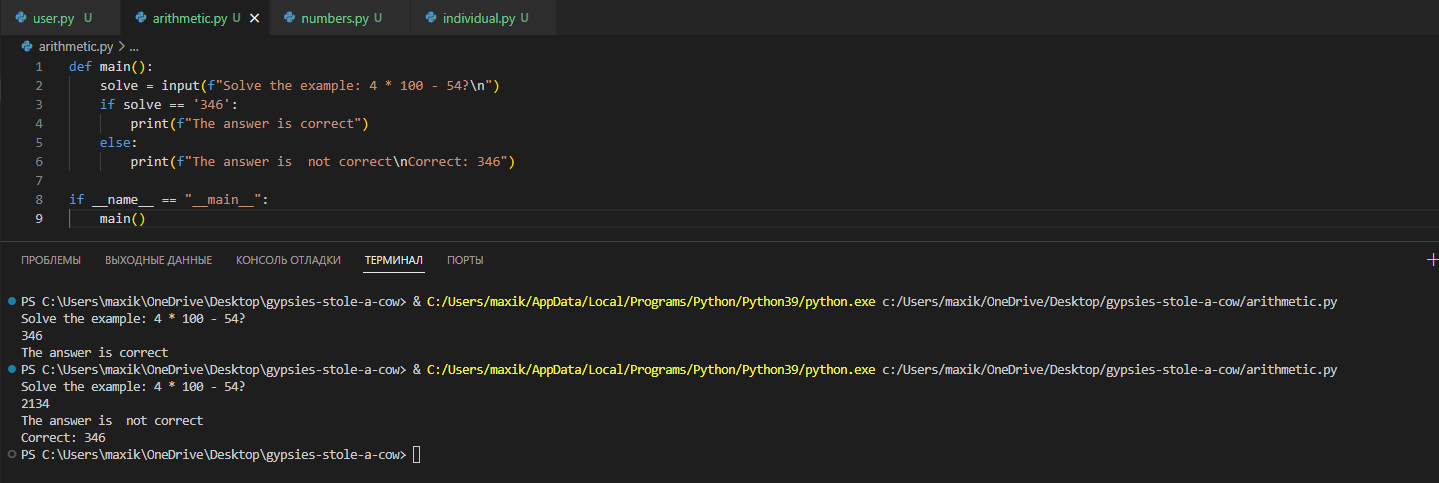


Рисунок 4 – Создал файл «arithmetic.py» и написал программу

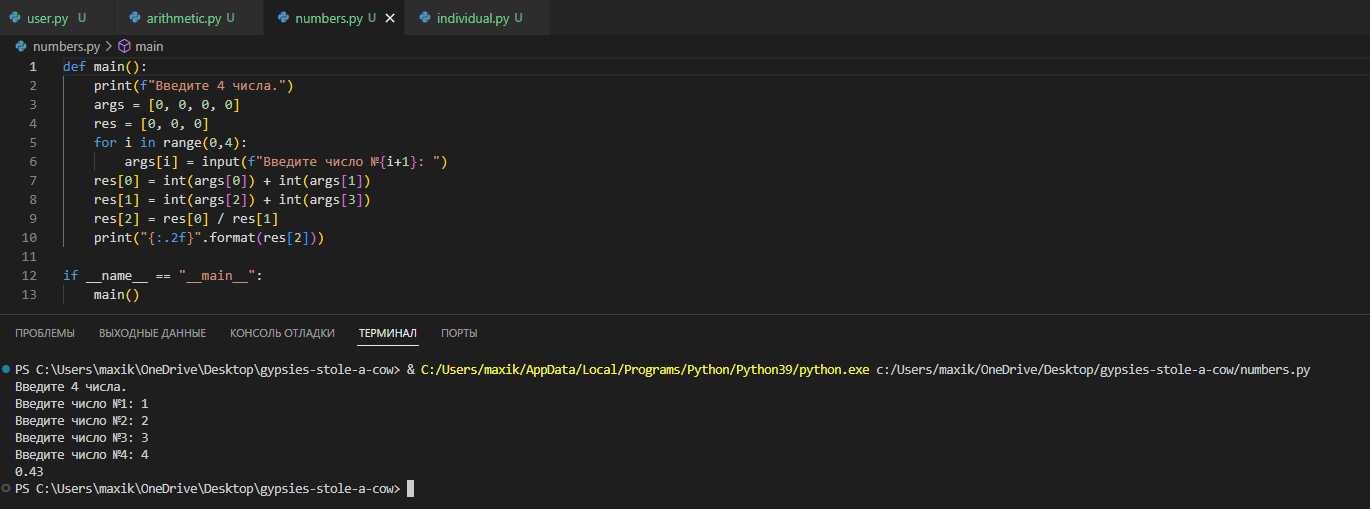


Рисунок 5 – Создал файл «numbers.py» и написал программу



Рисунок 6 – Создал файл «individual.py» и написал программу

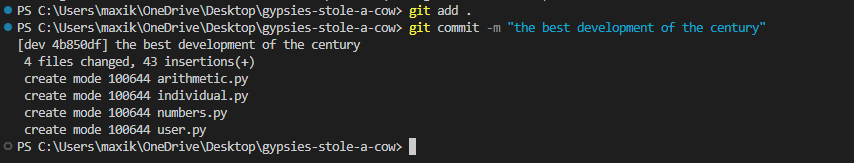


Рисунок 7 – Выполнил коммит в ветке «dev»

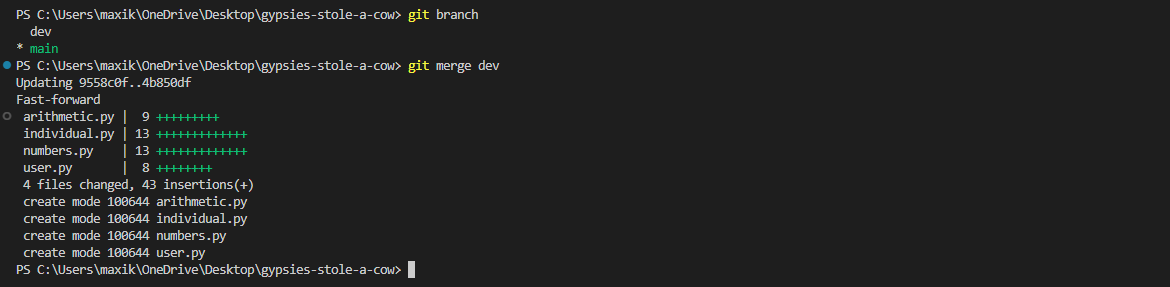


Рисунок 8 – Перешёл в ветку «main» и выполнил слияние с веткой «dev»

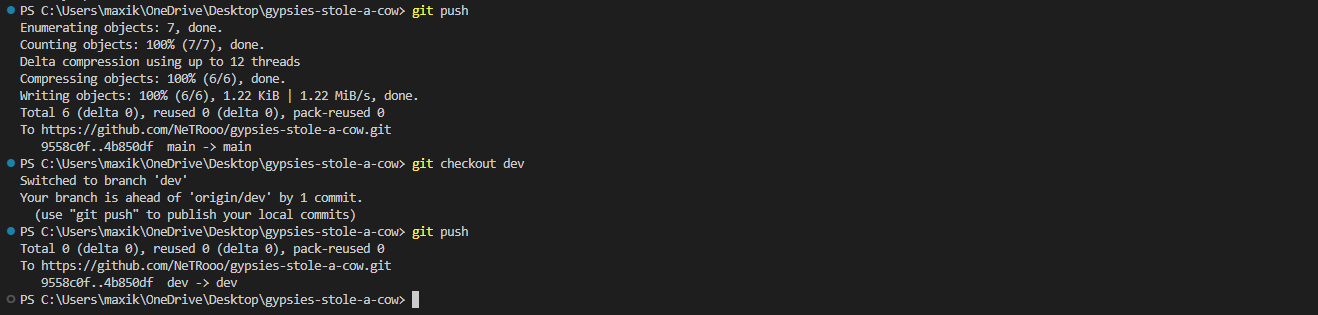


Рисунок 9 – Отправил коммиты в удалённый репозиторий GitHub

Вывод: В данной работе мы научились использовать инструменты git для разработки в нескольких ветках, а после использовать слияние. Научились работать с языком программирования Python, а также его правилами PEP8

**Ответы на контрольные вопросы**

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

С помощью команды git log. Существуют множество дополнительных опций вот некоторые из них:

* + - * -p или -patch – показывает разницу, внесенную в каждый коммит;
      * --stat – позволяет увидеть сокращенную статистику;
      * --pretty – эта опция меняет формат вывода;
      * --since и –until – опции для ограничения вывода по времени

1. Как ограничить вывод при просмотре истории коммитов?

С помощью команды git log и его аргументов, например –since.

1. Как внести изменения в уже сделанный коммит?

С помощью команды git commit –amend.

1. Как отменить индексацию файла в Git?

С помощью команды git reset HEAD.

1. Как отменить изменения в файле?

С помощью команды git checkout -- <file>.

1. Что такое удаленный репозиторий Git?

Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.

1. Как выполнить просмотр удаленных репозиториев данного локального репозитория?

С помощью команды git remote -v.

1. Как добавить удаленный репозиторий для данного локального репозитория?

С помощью команды git remote add <shortname> <url>.

1. Как выполнить отправку/получение изменений с удаленного репозитория?

Получение изменение – git fetch [remote-name], отправка изменений git push <remote-name> <branch-name>.

1. Как выполнить просмотр удаленного репозитория?  
   С помощью команды git remote -v.
2. Каково назначение тэгов Git?

Тэги Git - это ссылки на определенные коммиты в истории разработки. Они используются для пометки определенных версий или моментов в вашем проекте. Тэги обычно используются для обозначения релизов или важных этапов в разработке.

1. Как осуществляется работа с тэгами Git?

Для просмотра тэгов – git tag, для создания аннотированного тэга – git tag – a v1.4 -m “сообщение”, для отправки тэга на удаленный сервер – git push origin <tagname>, для удаления тэгов – git tag -d <tagname>.

1. . Самостоятельно изучите назначение флага --prune в командах git fetch и git push . Каково назначение этого флага?

Флаг --prune в командах git fetch и git push используется для удаления удаленных веток или тэгов, которые больше не существуют на удаленном репозитории. Если выполнить команду git fetch --prune, Git удалит локальные ссылки на удаленные ветки и тэги, которые были удалены на удаленном репозитории. Если выполнить команду git push --prune, Git удалит удаленные ветки и тэги на удаленном репозитории, которых больше нет в вашем локальном репозитории.