

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 №2.17**

Выполнила:  
Кузнецова Алена Валерьевна  
2 курс, группа ИВТ-б-о-21-1,  
09.03.01 «Информатика  
и вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем»,  
очная форма обучения

---

(подпись)

Ставрополь, 2022 г.

**Цель работы:** приобретение построения приложений с интерфейсом командной строки с помощью языка программирования Python.

**Выполнение работы:**

1. Создали общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

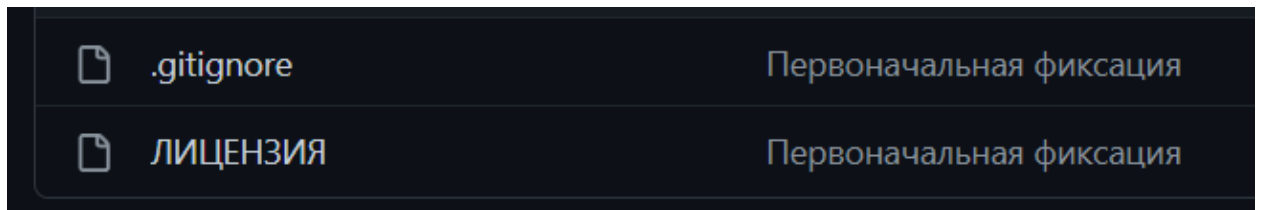


Рисунок 1 – Создание репозитория

2. Выполнили клонирование созданного репозитория.

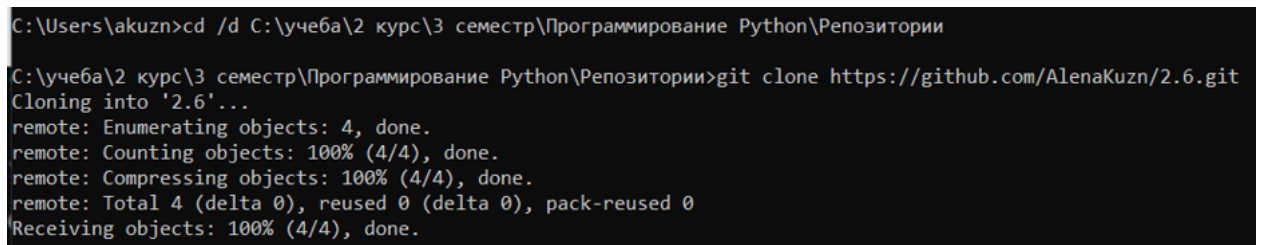


Рисунок 2 – Клонирование репозитория

3. Организовали свой репозиторий в соответствии с моделью git-flow.

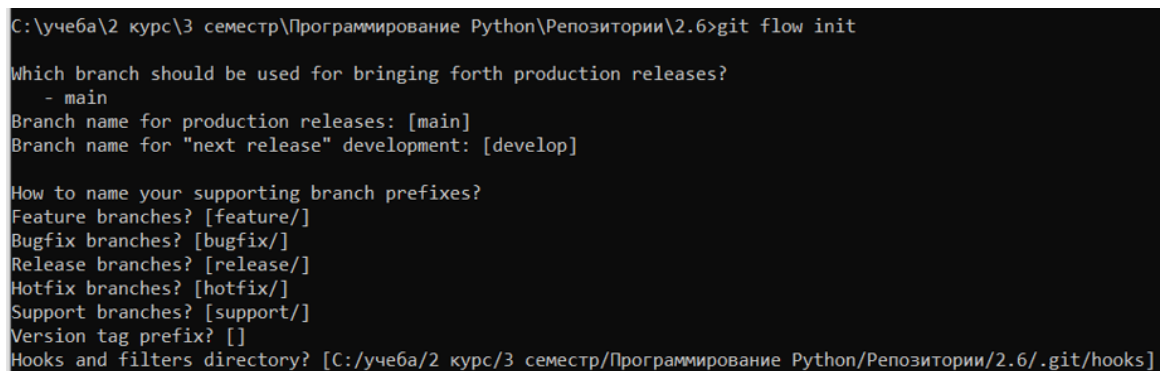


Рисунок 3 – Моделью ветвления git-flow

#### 4. Разобрали пример

```
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории\3.7>python example.py add data.json --name="Михайлов Михаил" --post="Директор" --year=2011
```

```
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории\3.7>python example.py display data.json
```

No	Ф.И.О.	Должность	Год
1	Иванов Иван	Директор	2007
2	Петров Петр	Бухгалтер	2010
3	Сидоров Сидор	Главный инженер	2012
4	Михайлов Михаил	Директор	2011

```
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории\3.7>python example.py select data.json --period=12
```

No	Ф.И.О.	Должность	Год
1	Иванов Иван	Директор	2007
2	Петров Петр	Бухгалтер	2010

Рисунок 4 – Результат

```
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории\2.17\example>python indiv.py add data.json --pynkt="Москва" --numb=3915 --samolet="Самолет"
```

```
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории\2.17\example>python indiv.py add data.json --pynkt="Ставрополь" --numb=2784 --samolet="Самолет"
```

```
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории\2.17\example>python indiv.py add data.json --pynkt="Краснодар" --numb=6127 --samolet="Почтовый"
```

```
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории\2.17\example>python indiv.py display data.json
```

No	Пункт назначения	Номер рейса	Тип
1	Москва	3915	Самолет
2	Ставрополь	2784	Самолет
3	Краснодар	6127	Почтовый

Рисунок 5 – Результат индивидуального задания

```
C:\учеба\2 курс\3 семестр\Программирование Python\Репозитории\2.17\example>python individ_hard.py -c display data.json
```

No	Пункт назначения	Номер рейса	Тип
1	Москва	3915	Самолет
2	Ставрополь	2784	Самолет
3	Краснодар	6127	Почтовый

Рисунок 6 – Задание повышенной сложности

**Вывод:** мы приобрели навыки построения приложений с интерфейсом командной строки с помощью языка программирования Python.

#### Контрольные вопросы:

1. В чем отличие терминала и консоли?

Терминал (от лат. terminus — граница) — устройство или ПО, выступающее посредником между человеком и вычислительной системой. Обычно данный термин используется, когда точка доступа к системе вынесена

в отдельное физическое устройство и предоставляет свой пользовательский интерфейс на основе внутреннего интерфейса (например, сетевых протоколов).

Консоль `console` — исторически реализация терминала с клавиатурой и текстовым дисплеем. В настоящее время это слово часто используется как синоним сеанса работы или окна оболочки командной строки. В том же смысле иногда применяется и слово “терминал”.

## 2. Что такое консольное приложение?

Консольное приложение `console application` — вид ПО, разработанный с расчётом на работу внутри оболочки командной строки, т.е. опирающийся на текстовый ввод-вывод.

## 3. Какие существуют средства языка программирования Python для построения приложений командной строки?

Python 3 поддерживает несколько различных способов обработки аргументов командной строки. Встроенный способ – использовать модуль `sys`. С точки зрения имен и использования, он имеет прямое отношение к библиотеке C (`libc`). Вторым способом – это модуль `getopt`, который обрабатывает как короткие, так и длинные параметры, включая оценку значений параметров.

## 4. Какие особенности построения CLI с использованием модуля `sys`?

Это базовый модуль, который с самого начала поставлялся с Python. Он использует подход, очень похожий на библиотеку C, с использованием `argc` и `argv` для доступа к аргументам. Модуль `sys` реализует аргументы командной строки в простой структуре списка с именем `sys.argv`.