

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Кафедра инфокоммуникаций
«Элементы объектно-ориентированного программирования в языке
Python»»»
Отчет по лабораторной
работе по дисциплине
«Объектно-ориентированное
программирование»

Выполнил студент группы ИВТ-б-о-21-1

Кочкаров Умар Ахматович.

«17» ноября 2023г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: приобретение навыков по работе с классами и объектами при написании программ с помощью языка программирования

Python версии 3.x.

Ход работы:

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

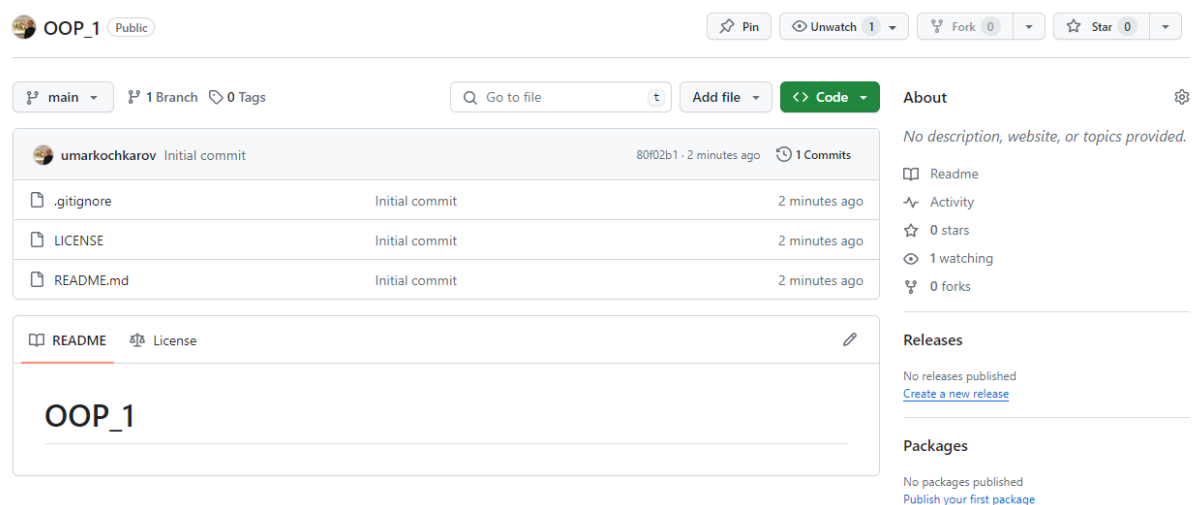


Рисунок 1. Создание репозитория

2. Организовать репозиторий в соответствии с Git-Flow init.

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/OOP_1 (main)
$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/erken/Desktop/python/OOP_1/.git/hooks]
```

Рисунок 2. Организация в соответствии с Git-Flow init

3. Проработка примеров лабораторной работы:

```
C:\Users\erken\Desktop\python\Примеры\2.10\venv\Scripts\python.exe C:/Users/erken/Desktop/py
3/4
Введите обыкновенную дробь: 3/4
3/4
3/2
0/1
9/16
1/1
```

Рисунок 3. Пример

4. Индивидуальные задания:

Задание 1. Поле `first` — дробное число; поле `second` — целое число, показатель степени. Реализовать метод `power()` — возведение числа `first` в степень `second`. Метод должен правильно работать при любых допустимых значениях `first` и `second`.

```
C:\Users\erken\Desktop\python\Примеры\2.10\venv\Scripts\python.exe C:/Users/erken/Desktop
first: 3
second: 0.5
first: 7.5
second: 1.25
```

Рисунок 4. Индивидуальное задание 1

Задание 2. Создать класс `Point` для работы с точками на плоскости. Координаты точки — декартовы. Обязательно должны быть реализованы: перемещение точки по оси `X`, перемещение по оси `Y`, определение расстояния до начала координат, расстояния между двумя точками, преобразование в полярные координаты, сравнение на совпадение и несовпадение.

```
C:\Users\erken\Desktop\python\Примеры\2.10\venv\Scripts\python.exe C:/Users/erken/
x: 10
y: 5
direction: horizontal
size: 5
is_visible: True
x: 12
y: 8
direction: horizontal
size: 5
is_visible: True
x: 12
y: 8
direction: vertical
size: 5
is_visible: True
x: 12
y: 8
direction: vertical
size: 10
is_visible: True
x: 12
y: 8
direction: vertical
size: 10
is_visible: False
x: 12
y: 8
direction: vertical
size: 10
is_visible: True
```

Рисунок 5. Индивидуальное задание 2

Контрольные вопросы:

1. Как осуществляется объявление класса в языке Python?

Классы объявляются с помощью ключевого слова `class` и имени класса:

```
# class syntax
```

```
class MyClass:
```

```
var = ... # некоторая переменная
```

```
def do_smt(self):
```

```
# какой-то метод
```

2. Чем атрибуты класса отличаются от атрибутов экземпляра?

Атрибут класса - это атрибут, общий для всех экземпляров класса.

Атрибуты класса определены внутри класса, но вне каких-либо методов.

Их значения одинаковы для всех экземпляров этого класса. Так что вы можете рассматривать их как тип значений по умолчанию для всех наших объектов.

Атрибуты экземпляра определяются в методах и хранят информацию, специфичную для экземпляра.

3. Каково назначение методов класса?

Классы позволяют определять данные и поведение похожих объектов.

Поведение описывается методами. Метод похож на функцию тем, что это блок кода, который имеет имя и выполняет определенное действие. Методы, однако, не являются независимыми, поскольку они определены внутри класса.

4. Для чего предназначен метод `init()` класса?

Метод `init` является конструктором. Конструкторы - это концепция объектно-ориентированного программирования. Класс может иметь один и только один конструктор. Если `init` определен внутри класса, он автоматически вызывается при создании нового экземпляра класса.

Метод `init` указывает, какие атрибуты будут у экземпляров нашего класса.

5. Каково назначение `self` ?

Аргумент `self` представляет конкретный экземпляр класса и позволяет получить доступ к его атрибутам и методам. Важно использовать параметр `self` внутри метода, если мы хотим сохранить значения экземпляра для последующего использования.

В большинстве случаев нам также необходимо использовать параметр `self` в других методах, потому что при вызове метода первым аргументом, который ему передается, является сам объект.

6. Как добавить атрибуты в класс?

Например, мы хотим видеть информацию о всех видах наших питомцев. Мы могли бы записать ее в самом классе с самого начала или создать переменную следующим образом:

```
Pet.all_specs = [tom.spec, avocado.spec, ben.spec]
```

7. Как осуществляется управление доступом к методам и атрибутам в языке Python?

Хорошим тоном считается, что для чтения/изменения какого-то атрибута должны использоваться специальные методы, которые называются

getter/setter, их можно реализовать, но ничего не мешает изменить атрибут напрямую. При этом есть соглашение, что метод или атрибут, который начинается с нижнего подчеркивания, является скрытым, и снаружи класса трогать его не нужно (хотя сделать это можно).

8. Каково назначение функции `isinstance` ?

Встроенная функция `isinstance(obj, Cls)` , используемая при реализации методов арифметических операций и операций отношения, позволяет узнать что некоторый объект `obj` является либо экземпляром класса `Cls` либо экземпляром одного из потомков класса `Cls`.

Вывод: приобретены навыки по работе с классами и объектами при написании программ с помощью языка программирования Python версии 3.x.