

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
«Наследование и полиморфизм 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 созданию иерархии классов 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.

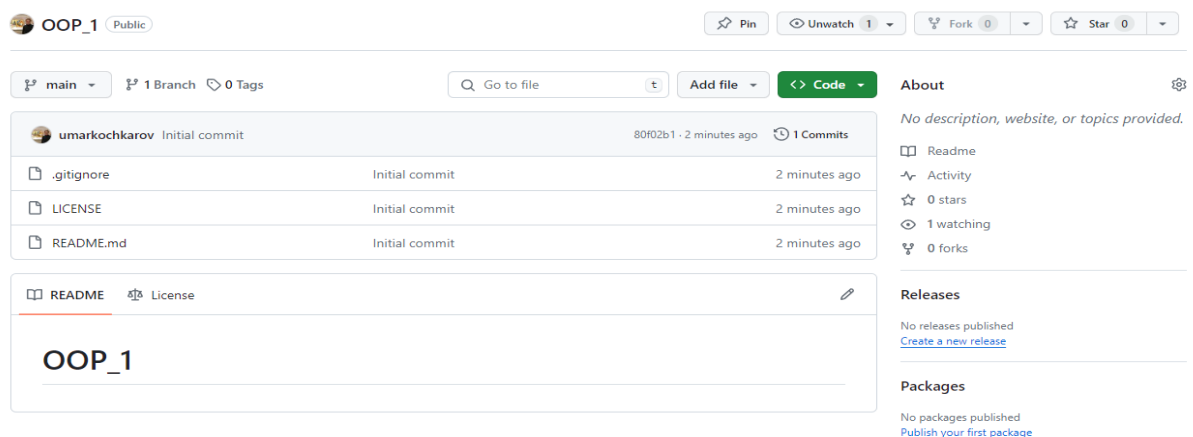


Рисунок 1. Создание репозитория

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\AppData\Local\Programs\Python\Python310\python.exe C:/Users/erken/...
Введите общее количество солдат: 12
Герой: (unique_id: 1, level: 2, team: red)
Идентификатор солдата, который следует за героем: 3
```

Рисунок 3. Пример

4. Индивидуальные задания(вариант 7):

Задание 1. Создать класс Triangle с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс Equilateral (равносторонний), имеющий поле площади. Определить метод вычисления площади.

```
C:\Users\erken\AppData\Local\Programs\Python\Python310\python.exe C:/Users/erken/Desktop/project/main.py
Введите длину первой стороны треугольника: 3
Введите длину второй стороны треугольника: 4
Введите длину третьей стороны треугольника: 5
Периметр треугольника: 18.0
Углы треугольника: (44.415308597192976, 57.12165043562251, 78.46304096718453)
Введите длину стороны равностороннего треугольника: 5
Периметр равностороннего треугольника: 15.0
Площадь равностороннего треугольника: 10.825317547305483
```

Рисунок 4. Индивидуальное задание 1

Задание 2. Создать абстрактный базовый класс Root (корень) с виртуальными методами вычисления корней и вывода результата на экран. Определить производные классы Linear (линейное уравнение) и Square (квадратное уравнение) с собственными методами вычисления корней и вывода на экран.

```
C:\Users\erken\AppData\Local\Programs\Python\Python310\python.exe C:/Users/erken/Desktop/project/main.py
Линейное уравнение:
2x + -3 = 0
Корни: [1.5]
Результат: Линейное уравнение: 2x + -3 = 0
Квадратное уравнение:
1x^2 + -3x + 2 = 0
Корни: [2.0, 1.0]
Результат: Квадратное уравнение: 1x^2 + -3x + 2 = 0
```

Рисунок 5. Индивидуальное задание 2

Контрольные вопросы:

1. Что такое наследование и как оно реализовано в языке Python?

Наследование — это когда один класс (подкласс) получает свойства и методы другого класса (суперкласса). Подкласс может наследовать все публичные атрибуты и методы своего суперкласса и добавлять свои собственные. В языке Python наследование реализуется с помощью ключевого слова `class`. Для создания подкласса нужно указать имя суперкласса в скобках

после имени подкласса. Подкласс получает все атрибуты и методы суперкласса, их можно использовать напрямую или переопределить.

2. Что такое полиморфизм и как он реализован в языке Python?

Полиморфизм — это возможность объектов разных классов иметь одно и то же имя метода, но каждый класс может предоставить свою собственную реализацию этого метода. Это позволяет использовать одинаковое имя метода для объектов различных классов, что упрощает программирование и повышает гибкость кода. В языке Python полиморфизм реализуется через наследование и переопределение методов. Если в подклассе метод с тем же именем переопределяется, то при вызове этого метода на объекте подкласса будет использоваться его реализация, а не реализация суперкласса. Это позволяет использовать одинаковые методы с разным поведением для разных классов.

3. Что такое «утиная» типизация в языке Python?

«Утиная» типизация (англ. duck typing) — это концепция в языке программирования Python, основанная на философии «если она выглядит как утка, плавает как утка и крякает как утка, то это, вероятно, и есть утка». В контексте Python утиная типизация означает, что тип объекта определяется по его возможностям и методам, а не по его явно заданному типу. Иными словами, если объект обладает определенными методами, то мы можем использовать его как экземпляр нужного типа, не задумываясь о его фактическом классе или интерфейсе.

4. Каково назначение модуля abc языка Python?

Модуль abc (аббревиатура от "Abstract Base Classes") является частью стандартной библиотеки языка Python и предоставляет средства для определения абстрактных базовых классов.

5. Как сделать некоторый метод класса абстрактным?

Необходимо декорировать его методы как абстрактные, а реализацию выносить в классы-наследники.

6. Как сделать некоторое свойство класса абстрактным?

Можно потребовать атрибут в конкретных классах, определив их с помощью `@abstractproperty`.

7. Каково назначение функции `isinstance`?

Функция `isinstance()` проверяет, является ли объект экземпляром указанного класса или его подкласса.

Вывод: приобретены навыки по созданию иерархии классов при написании программ с помощью языка программирования Python версии 3.x.