

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 № 3
по дисциплине «Объектно-ориентированное программирование»
Вариант №22

Выполнил студент группы ИВТ-б-о-23-2

Сумин Никита Сергеевич.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2025

Цель работы: приобретение навыков по созданию иерархии классов при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

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

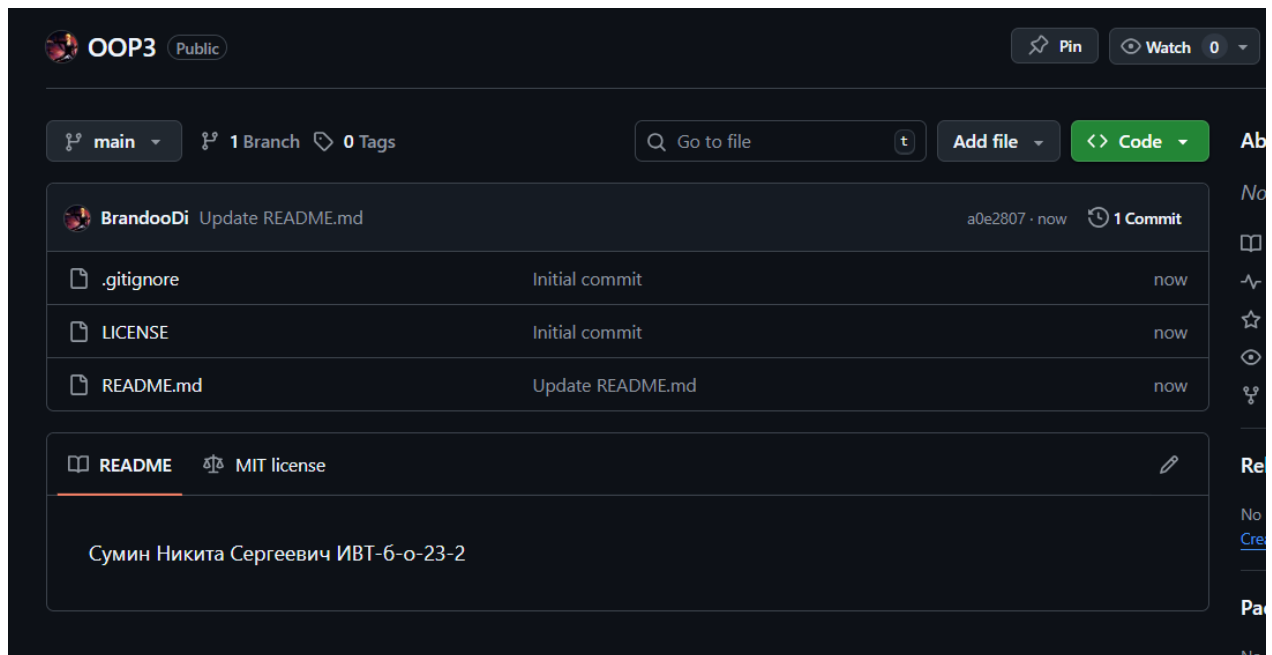


Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

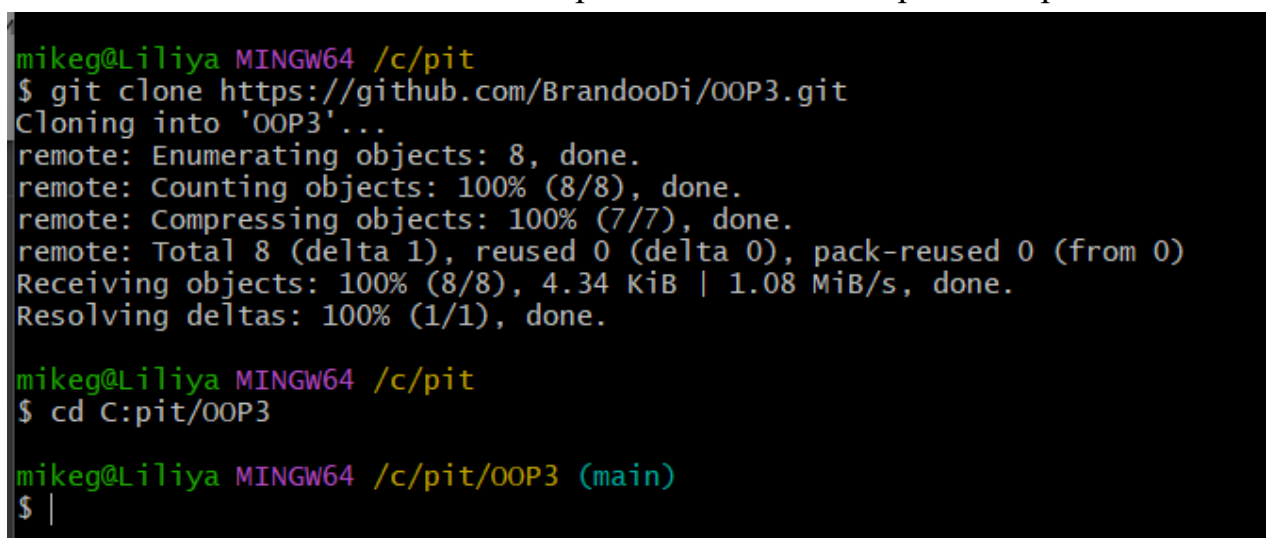


Рисунок 2 - Клонирование репозитория

3. Проработать примеры лабораторной работы.

```
C:\Users\mikeg\AppData\Local\Programs\Python\Python311\python.exe C:\pit\00P3\Project\primer1.py
3/4
Введите обыкновенную дробь: 3/4
3/4
3/2
0/1
9/16
1/1

Process finished with exit code 0
|
```

Рисунок 3 - Результат выполнения примера 1

```
C:\Users\mikeg\AppData\Local\Programs\Python\Python311\python.exe C:\pit\00P3\Project\primer2.py
I have 3 sides
I have 4 sides
I have 5 sides
I have 6 sides

Process finished with exit code 0
```

Рисунок 4 - Результат выполнения примера 2

```
C:\Users\mikeg\AppData\Local\Programs\Python\Python311\python.exe C:\pit\00P3\Project\primer3.py
I can walk and run
I can crawl
I can bark
I can roar

Process finished with exit code 0
|
```

Рисунок 5 - Результат выполнения примера 3

```
C:\Users\mikeg\AppData\Local\Programs\Python\Python311\python.exe C:\pit\00P3\Project\task1.py
Pair 1: (3, 5)|
Pair 2: (2, 7)
Pair 1 after change: (4, 6)
Pair (4, 6) is greater than Pair (2, 7)
Fraction 1: (1, 0.5)
Fraction 2: (2, 0.3)
Fraction 1 after change: (3, 0.8)
None

Process finished with exit code 0
```

Рисунок 6 - Результат выполнения задания 1

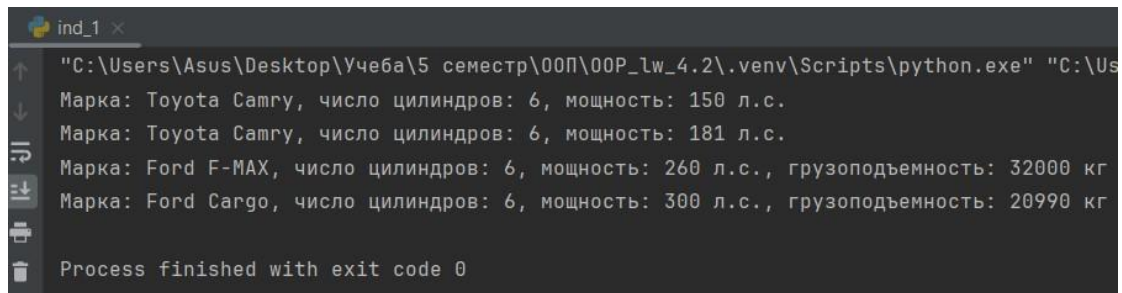
6. Выполнить индивидуальные задания.

Задание 1. Вариант 22 (2)

Класс Pair (пара чисел); определить методы изменения полей или сравнения пар: пара p1 больше пары p2, если (first.p1 > first.p2) или (first.p1 = first.p2) и (second.p1 > second.p2).

Определить класс-наследник Fraction с полями: целая часть числа и дробная часть числа.

Определить полный набор методов сравнения.



```
ind_1 x
"C:\Users\Asus\Desktop\Учеба\5 семестр\00П\00P_lw_4.2\.venv\Scripts\python.exe" "C:\Us
Марка: Toyota Camry, число цилиндров: 6, мощность: 150 л.с.
Марка: Toyota Camry, число цилиндров: 6, мощность: 181 л.с.
Марка: Ford F-MAX, число цилиндров: 6, мощность: 260 л.с., грузоподъемность: 32000 кг
Марка: Ford Cargo, число цилиндров: 6, мощность: 300 л.с., грузоподъемность: 20990 кг
Process finished with exit code 0
```

Рисунок 7 - Результат выполнения индивидуального задания 1

Задание 2. Вариант 22 (4)

Нужно сделать абстрактный класс Triangle, хранящий две стороны и угол между ними, с виртуальными методами площадь и периметр.

Затем реализовать три конкретных класса для частных случаев треугольника (прямоугольный, равнобедренный, равносторонний), переопределив в них методы вычисления.

```
C:\Users\mikeg\AppData\Local\Programs\Python\Python311\python.exe C:\pit\00P3\Project\task2.py
=== Работа с треугольниками ===

Треугольник 1: Прямоугольный треугольник: катеты 3 и 4
Площадь: 6.00
Периметр: 12.00

Треугольник 2: Равнобедренный треугольник: сторона=5, угол=30°
Площадь: 6.25
Периметр: 12.59

Треугольник 3: Равносторонний треугольник: сторона=6
Площадь: 15.59
Периметр: 18.00

=== Детальный пример ===
Прямоугольный треугольник: катеты 5 и 12
Площадь: 30.0
Периметр: 30.00

Равносторонний треугольник:
Равносторонний треугольник: сторона=10
Площадь: 43.30
```

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