

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

Кафедра инфокоммуникаций
«Перегрузка операторов в языке Python»

Отчет по лабораторной работе № 4.2
по дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-21-1
Толубаев Рамиль Ахметович

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.

Порядок выполнения работы:

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

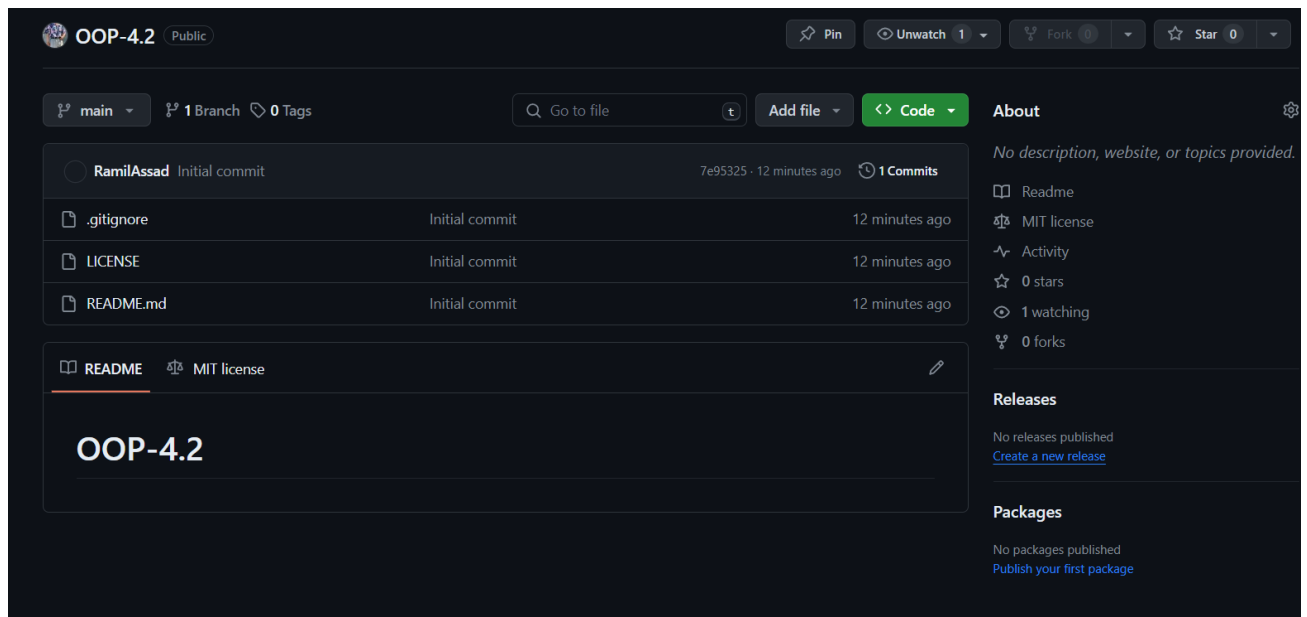


Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

```
PS C:\Users\Asus\OneDrive\Рабочий стол\учёба 3 курс\ООП> git clone https://github.com/RamilAssad/OOP-4.2.git
Cloning into 'OOP-4.2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
PS C:\Users\Asus\OneDrive\Рабочий стол\учёба 3 курс\ООП>
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\Users\Asus\Desktop\Учеба\5 семестр\ООП\ООР_lw_4.1>git checkout -b develop
Switched to a new branch 'develop'

C:\Users\Asus\Desktop\Учеба\5 семестр\ООП\ООР_lw_4.1>
```

Рисунок 3 - Ветвление по модели git-flow

4. Проработать примеры лабораторной работы.

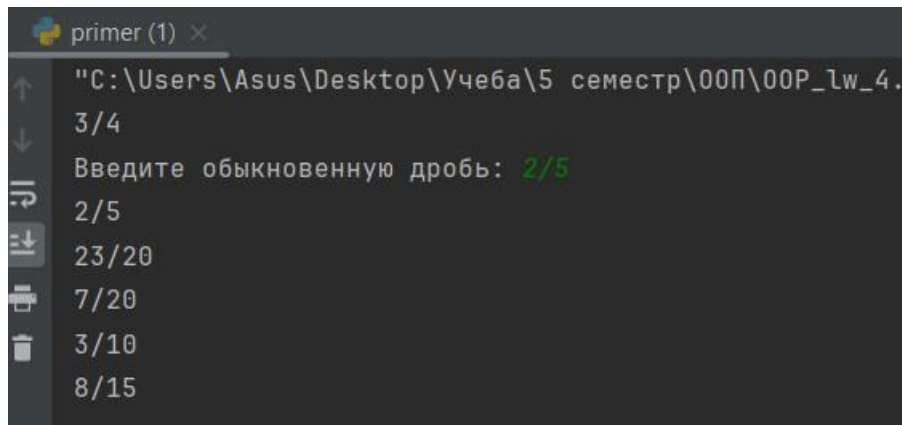


Рисунок 4 - Результат выполнения примера

5. Выполнить индивидуальные задания.

Задание 1. Вариант 16

Выполнить индивидуальное задание 1 лабораторной работы 4.1, максимально задействовав имеющиеся в Python средства перегрузки операторов.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

class Pair:
    def __init__(self, first, second):
        if not isinstance(first, int) or not isinstance(second, float):
            raise ValueError("Некорректные значения аргументов")
        self.first = first
        self.second = second

    def __mul__(self, number):
        if not isinstance(number, float):
            raise ValueError("Некорректное значение аргумента")
        return Pair(self.first * number, self.second * number)

    def __str__(self):
        return f"first: {self.first}\nsecond: {self.second}"

def make_Pair(first, second):
    return Pair(first, second)

if __name__ == "__main__":
    p = make_Pair(3, 0.5)
    print(p)

    p = p * 2.5
    print(p)
```

Рисунок 5 - Результат выполнения индивидуального задания 1

Задание 2. Вариант 16

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.

Рисунок 6 - Результат выполнения индивидуального задания 2

Контрольные вопросы:

1. Какие средства существуют в Python для перегрузки операций?

В Python для перегрузки операций существуют специальные методы, начинающиеся и заканчивающиеся двумя подчеркиваниями (например, `__add__` для перегрузки операции сложения).

2. Какие существуют методы для перегрузки арифметических операций и операций отношения в языке Python?

Для перегрузки арифметических операций в Python существуют следующие методы:

- `__add__` для перегрузки операции сложения
- `__sub__` для перегрузки операции вычитания
- `__mul__` для перегрузки операции умножения
- `__truediv__` для перегрузки операции деления
- и другие аналогичные методы для других арифметических операций.

Для перегрузки операций отношения существуют следующие методы:

- `__eq__` для перегрузки операции равенства
- `__ne__` для перегрузки операции неравенства
- `__lt__` для перегрузки операции "меньше"
- `__gt__` для перегрузки операции "больше"
- и другие аналогичные методы для других операций отношения.

3. В каких случаях будут вызваны следующие методы: `add`, `iadd` и `radd` ?

Методы `add`, `iadd` и `radd` будут вызваны в следующих случаях:

- `add` вызывается при использовании оператора `+` для объектов (например, `obj1 + obj2`)
- `iadd` вызывается при использовании оператора `+=` для объектов (например, `obj1 += obj2`)
- `radd` вызывается при использовании оператора `+` для объектов, если левый операнд не поддерживает операцию сложения, но правый операнд поддерживает

(например, `1 + obj`, где `obj` поддерживает сложение)

4. Для каких целей предназначен метод `new` ? Чем он отличается от метода `init` ?

Метод `new` предназначен для создания нового экземпляра класса. Он отличается от метода `init` тем, что `init` используется для инициализации нового экземпляра, а `new` используется для создания этого экземпляра до его инициализации.

5. Чем отличаются методы `str` и `repr`.

Метод `str` используется для представления объекта в виде строки, который чаще всего используется для вывода информации об объекте пользователю. Метод `repr` также используется для представления объекта в виде строки, но его цель - предоставить однозначное представление объекта, которое может быть использовано для его воссоздания.

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

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