

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №4.2  
Перегрузка операторов в языке Python  
по дисциплине «Объектно-ориентированное программирование»**

Выполнил студент группы ИВТ-б-о-20-1

Ищенко Т.С. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверила Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2022

Цель работы: приобретение навыков по работе с классами и объектами при написании программ с помощью языка программирования Python версии 3.x.

Ход работы:

1. Произвёл отработку примера, согласно методическим рекомендациям



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  class Rational:
5
6      def __init__(self, a=0, b=1):
7          a = int(a)
8          b = int(b)
9
10         if b == 0:
11             raise ValueError("Illegal value of the denominator")
12
13         self.__numerator = a
14         self.__denominator = b
15
16         self.__reduce()
17
18         # Сокращение дроби.
19         def __reduce(self):
20             # Функция для нахождения наибольшего общего делителя
21             def gcd(a, b):
22                 if a == 0:
23                     return b
24                 elif b == 0:
25                     return a
26                 elif a >= b:
27                     return gcd(a % b, b)
28                 else:
29                     return gcd(a, b % a)
30
31             sign = 1
```

Рисунок 1 – Результат выполнения примера

2. Произвёл выполнение индивидуального задания

```

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

'''
Поле first – дробное число, левая граница диапазона; поле second – дробное число,
правая граница диапазона. Реализовать метод rangecheck() – проверку заданного числа
на принадлежность диапазону.
'''

class MyClass:

    def __init__(self, chislo=0):
        self.chislo = chislo

    def __lt__(self, other):
        if self.chislo < other.chislo:
            return "done"

if __name__ == '__main__':
    first = MyClass(2)
    second = MyClass(7)
    third = MyClass(5)
    if first < second and second < third:
        print(f'Число {second} внутри диапазона от {first} до {third}')

```

Рисунок 2 – Результат выполнения индивидуального задания

3. Выполнил задание повышенной сложности, согласно методическим рекомендациям

```

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

'''
Реализовать класс Rational, используя два списка из 100 элементов типа int для
представления числителя и знаменателя. Каждый элемент является десятичной цифрой.
Младшая цифра имеет меньший индекс (единицы – в нулевом элементе списка). Реальный
размер списка задается как аргумент конструктора инициализации.
'''

class Rational:

    def __init__(self, chisl, znam):
        self.list_chisl = []
        self.list_znam = []
        self.CONST_MAX_SIZE = 100
        self.num_chisl = str(chisl)
        self.num_znam = str(znam)
        if len(self.num_znam) and len(self.num_chisl) < self.CONST_MAX_SIZE:
            if chisl and znam is not None:
                for i in self.num_chisl:
                    self.list_chisl.append(i)
                for i in self.num_znam:
                    self.list_znam.append(i)
            else:
                raise ValueError()

    def size(self):
        return f'Размер числителя {len(self.list_chisl)}, ' \
            f'Размер знаменателя: {len(self.list_znam)}'

```

Рисунок 3 – Результат выполнения задания повышенной сложности

Контрольные вопросы:

1. Как осуществляется объявление класса в языке Python? Классы объявляются с помощью ключевого слова `class` и имени класса.
2. Чем атрибуты класса отличаются от атрибутов экземпляра? 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. Что касается переменных экземпляра, они хранят данные, уникальные для каждого объекта класса.

3. Каково назначение методов класса? Методы определяют функциональность объектов, принадлежащих конкретному классу.

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.

5. Каково назначение `self`? Аргумент `self` это обращение к самому экземпляру класса.

6. Как добавить атрибуты в класс? “Имя объекта”. “Название атрибута” = “Значение атрибута”

7. Как осуществляется управление доступом к методам и атрибутам в языке Python? В Python таких возможностей нет, и любой может обратиться к атрибутам и методам вашего класса, если возникнет такая необходимость. Это существенный недостаток этого языка, т.к. нарушается один из ключевых принципов ООП – инкапсуляция. Хорошим тоном считается, что для чтения/изменения какого-то атрибута должны использоваться специальные методы, которые называются `getter/setter`, их можно реализовать, но ничего не помешает 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`? В Python есть встроенная функция `instance ()`, которая сравнивает значение с указанным типом. Если данное значение и тип соответствуют, он вернет `true`, иначе `false`. Используя `isinstance ()`, вы можете проверить строку, число с плавающей точкой, `int`, список, кортеж, `dict`, `set`, `class` и т.д.

Вывод: в ходе выполнения лабораторной работы приобрел навыки по работе с классами и объектами при написании программ с помощью языка программирования Python версии 3.x.