МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2 Перегрузка операторов в языке Python

по дисциплине «Объектно-ориентированное программирование»

|  |
| --- |
| Выполнил студент группы ИВТ-б-о-20-1 |
| Павленко М.С. « » 20 г. |
| Подпись студента |
| Работа защищена« » 20 г. |
| Проверил Воронкин Р.А.  (подпись) |

Ставрополь 2022

Цель работы: приобретение навыков по перегрузке операторов при написании программ с помощью языка программирования Python версии 3.x.

Ход работы

1. Изучив методические указания, приступил к разбору примера.

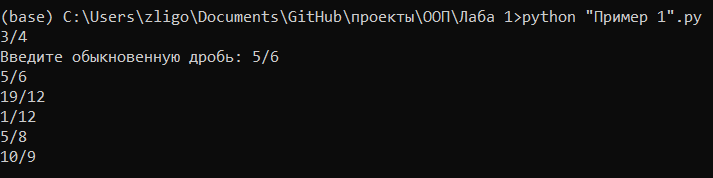


Рисунок 1.1 – Проверка правильности работы кода

1. Затем начал выполнять задания для моего варианта.

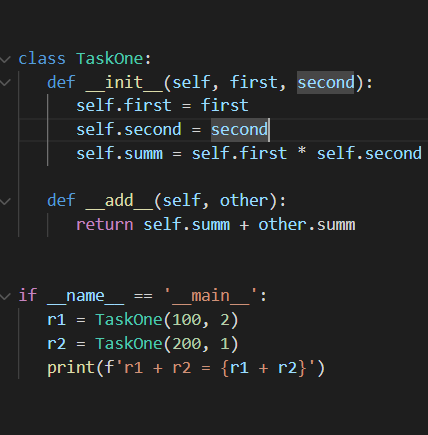


Рисунок 1.2 – Код первого индивидуального задания

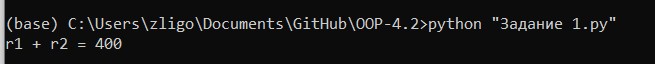


Рисунок 1.3 – Проверка кода первого задания

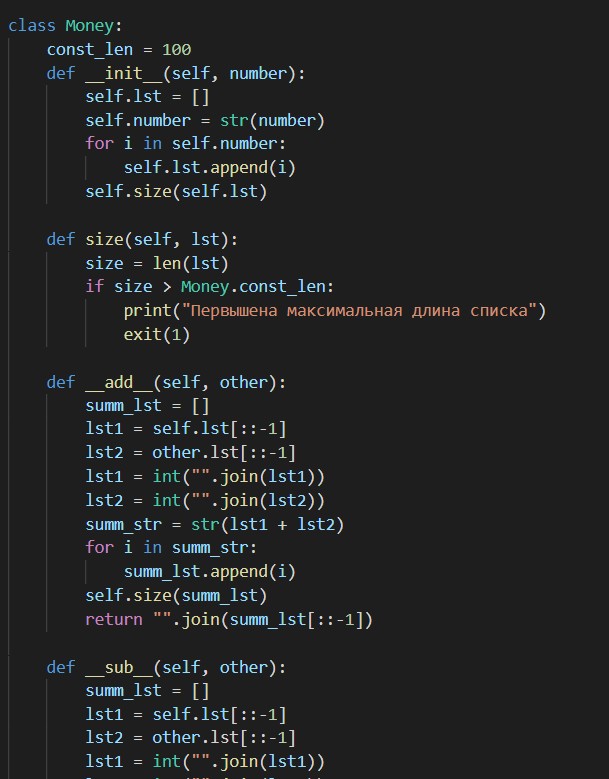


Рисунок 1.4 – Код второго задания

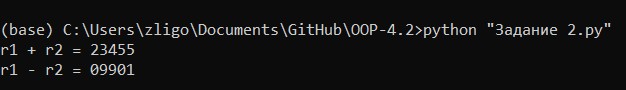


Рисунок 1.5 – Проверка кода второго задания

Контрольные вопросы

1. Какие средства существуют в Python для перегрузки операций? Перегрузка осуществляется при помощи специальных методов. Методы группируются по следующим категориям:

* методы для всех видов операций;
* методы перегрузки операторов работы с коллекциями;
* методы для числовых операций в двоичной форме;
* методы для других операций над числами;
* методы для операций с дескрипторами;
* методы для операций, используемых с диспетчерами контекста.

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

add (self, other) - сложение. x + y вызывает x. add (y) .

sub (self, other) - вычитание (x - y).

mul (self, other) - умножение (x \* y).

truediv (self, other) - деление (x / y).

floordiv (self, other) - целочисленное деление (x // y).

mod (self, other) - остаток от деления (x % y).

divmod (self, other) - частное и остаток (divmod(x, y)).

pow (self, other[, modulo]) - возведение в степень ( x \*\* y , pow(x, y[, modulo]) ).

lshift (self, other) - битовый сдвиг влево (x << y).

rshift (self, other) - битовый сдвиг вправо (x >> y).

and (self, other) - битовое И (x & y).

xor (self, other) - битовое ИСКЛЮЧАЮЩЕЕ ИЛИ (x ^ y).

radd (self, other) ,

rsub (self, other) ,

rmul (self, other) ,

rtruediv (self, other) ,

rfloordiv (self, other) ,

rmod (self, other) ,

rdivmod (self, other) ,

rpow (self, other) ,

rlshift (self, other) ,

rrshift (self, other) ,

rand (self, other) ,

rxor (self, other) ,

ror (self, other) - делают то же самое, что и арифметические операторы, перечисленные выше, но для аргументов, находящихся справа, и только в случае, если длялевого операнда не определён соответствующий метод.

iadd (self, other) - += .

isub (self, other) - -= .

imul (self, other) - \*= .

itruediv (self, other) - /= .

ifloordiv (self, other) - //= .

imod (self, other) - %= .

ipow (self, other[, modulo]) - \*\*= .

ilshift (self, other) - <<= .

irshift (self, other) - >>= .

iand (self, other) - &= .

ixor (self, other) - ^= .

ior (self, other) - |= .

1. В каких случаях будут вызваны следующие методы: add ,

iadd и radd ?

1. add - a + b
2. iadd - a += b
3. radd - Если не получилось вызвать метод add
4. Для каких целей предназначен метод new ? Чем он отличается от метода init ?

Метод new используется, когда нужно управлять процессом

создания нового экземпляра, а инициализация.

\_\_init

– когда контролируется его

1. Чем отличаются методы str и repr ?

str должен возвращать строковый объект, тогда как repr может возвращать любое выражение в Python

Вывод: в ходе выполнения лабораторной работы были приобретены простейшие навыки по работе с методами перегрузки операторов в языке программирования Python.