МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

**АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО**

ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

**Отчет по лабораторной работе № 4.2**

**«Перегрузка операторов в языке Python»**

**по дисциплине «Основы программной инженерии»**

Выполнила:

Образцова Мария Дмитриевна,

2 курс, группа ПИЖ-б-о-21-1, Проверил:

Доцент кафедры инфокоммуникаций, Воронкин Р.А.

Ставрополь, 2023 г.

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

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.

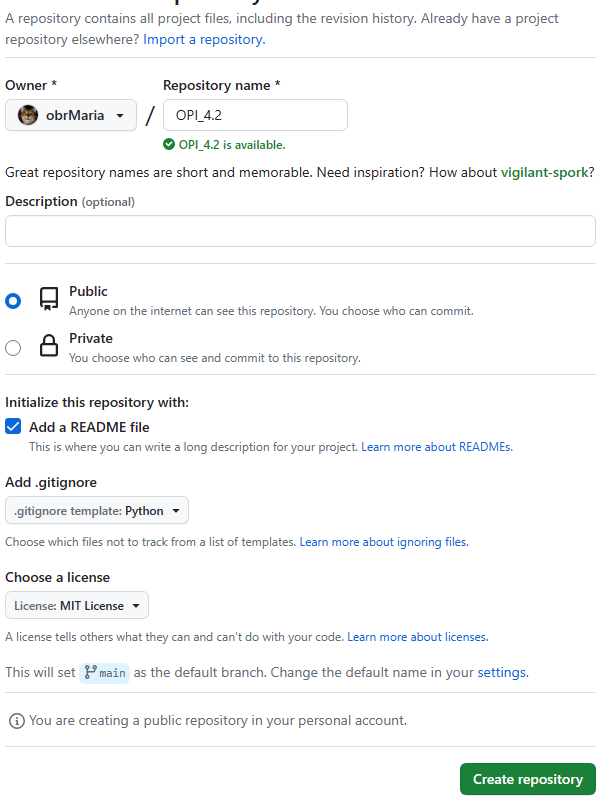


Рисунок –создание репозитория

1. Выполните клонирование созданного репозитория. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

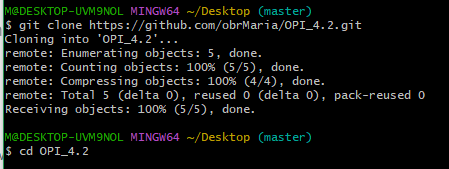
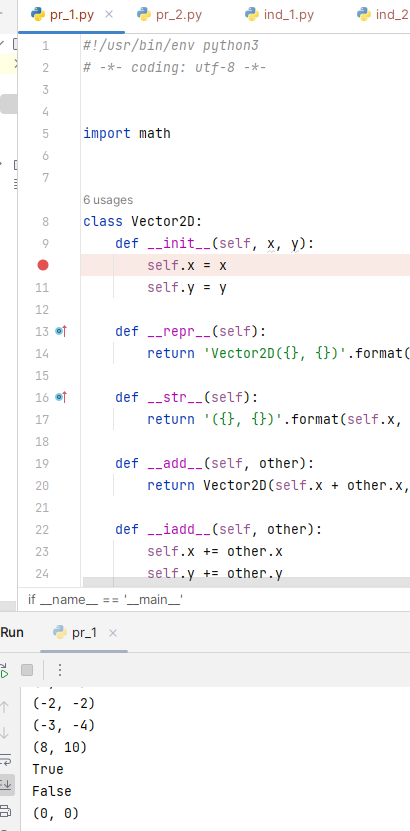


Рисунок – клонирование созданного репозитория

1. Проработать примеры лабораторной работы.

 Рисунок 1 – Перегрузка операторов

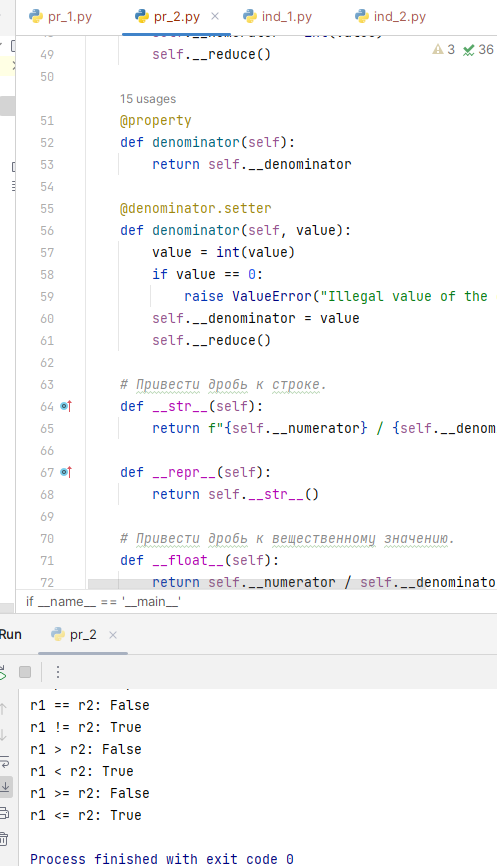


Рисунок 2 – Перегрузка операторов

1. Выполнить индивидуальные задания.

Задание 1

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

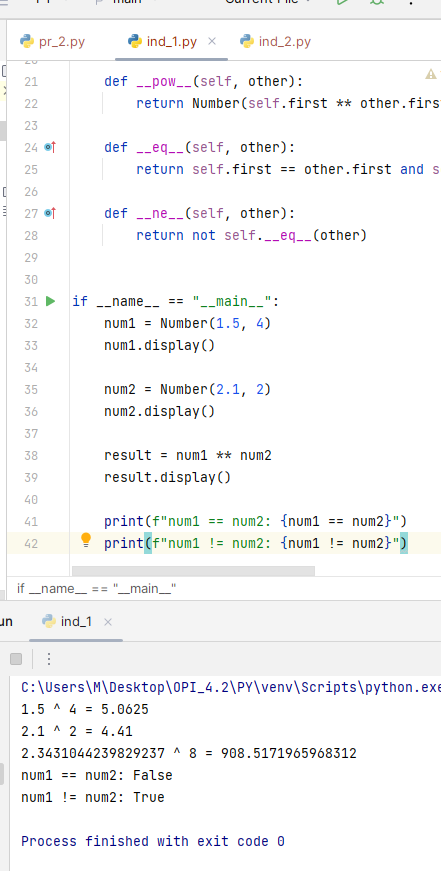


Рисунок – Индивидуальное задание №1

Задание 2

Дополнительно к требуемым в заданиях операциям перегрузить операцию индексирования []. Максимально возможный размер списка задать константой. В отдельном поле size должно храниться максимальное для данного объекта количество элементов списка; реализовать метод size(), возвращающий установленную длину. Если количество элементов списка изменяется во время работы, определить в классе поле count. Первоначальные значения size и count устанавливаются конструктором.

В тех задачах, где возможно, реализовать конструктор инициализации строкой.

Создать класс BitString для работы с битовыми строками не более чем из 100 бит. Битовая строка должна быть представлена списком типа int, каждый элемент которого принимает значение 0 или 1. Реальный размер списка задается как аргумент конструктора инициализации. Должны быть реализованы все традиционные операции для работы с битовыми строками: and, or, xor, not. Реализовать сдвиг влево и сдвиг вправо на заданное количество битов.

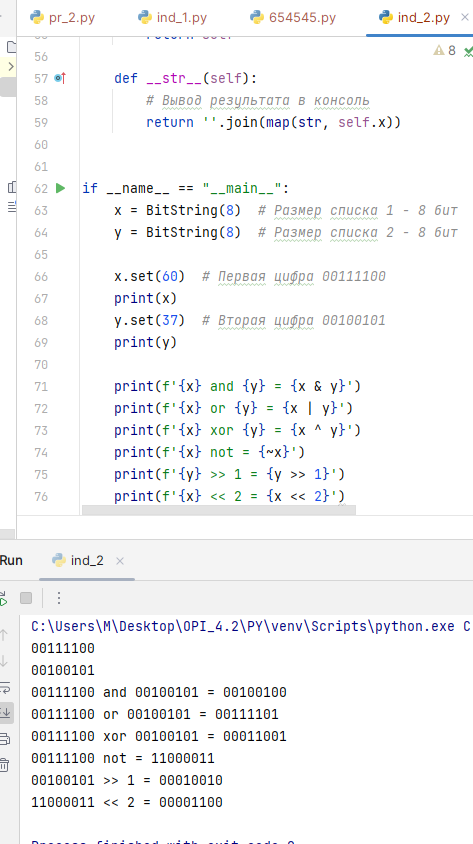


Рисунок – Индивидуальное задание №2

ВОПРОСЫ

1. Какие средства существуют в Python для перегрузки операций? Заключение опреатора в двойное подчёркивание « » с обеих сторон.
2. Какие существуют методы для перегрузки арифметических операций и операций отношения в языке Python?

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).

  or (self, other) - битовое 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 для левого операнда не определён соответствующий  метод.

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

  iadd и radd ?

Например, операция x + y будет сначала пытаться вызвать x. add (y)

, и только в том случае, если это не получилось, будет пытаться вызвать y. radd (x) . Аналогично для остальных методов.

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

Он управляет созданием экземпляра. В качестве обязательного аргумента принимает класс (не путать с экземпляром). Должен возвращать экземпляр класса для его последующей его передачи методу init .

1. Чем отличаются методы str и repr ?

  str (self) - вызывается функциями str, print и format. Возвращает строковое представление объекта.

  repr (self) - вызывается встроенной функцией repr; возвращает "сырые" данные, использующиеся для внутреннего представления в python.