**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

# Кафедра инфокоммуникаций

**Отчет**

# по лабораторной работе №8

**«Элементы объектно-ориентированного программирования в языке Python»**

# по дисциплине:

**«Введение в системы искусственного интеллекта»**

Вариант 3

Выполнил: студент группы ИВТ-б-о-18-1

Данченко Максим Игоревич

(подпись)

Проверил:

Воронкин Роман Александрович

(подпись)

Ставрополь, 2022 г.

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

# Задание №1

**3. Поле first — целое положительное число, числитель; поле second — целое положительное число, знаменатель. Реализовать метод ipart() — выделение целой части дроби first/second. Метод должен проверять неравенство знаменателя нулю.**

Был создан класс Neravenstvo с полями firs(числитель) и second(знаменатель), для доступа к переменным были реализованы get и set методы. Для выделение целой части дроби, был создан специальный метод def ipart () (рисунок 1)

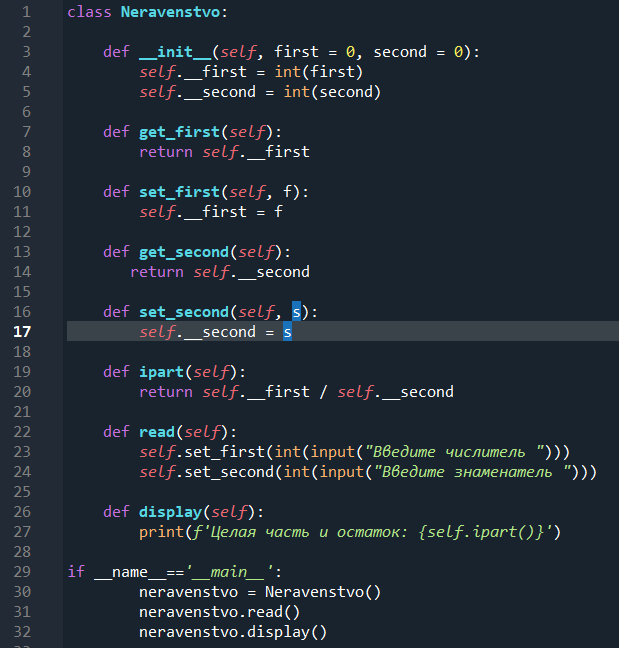


Рисунок 1 – Листинг программы

Результат работы программы изображен на рисунке 2

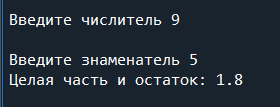


Рисунок 2 – Результат программы

# Задание №2

# Создать класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа int для рублей и копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения.

# Код программы показан на рисунках 1-4



Рисунок 1 – листинг программы

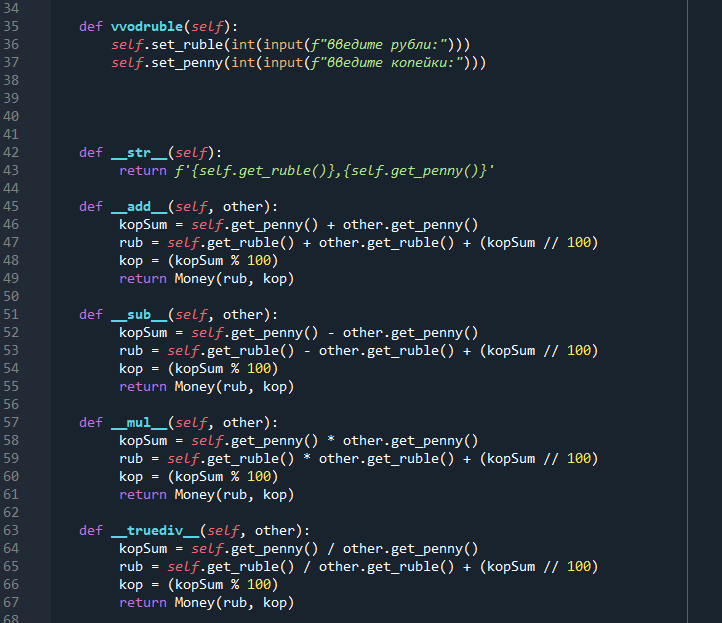


Рисунок 2 – листинг программы

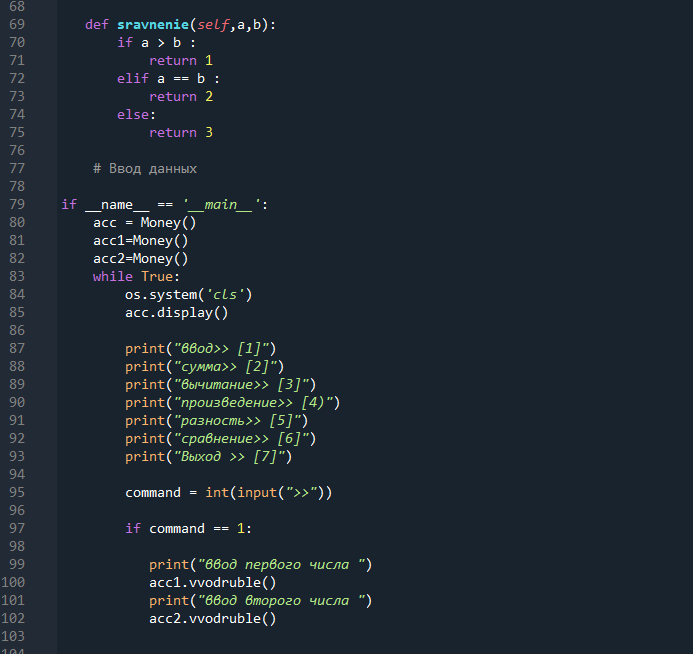


Рисунок 3 – листинг программы

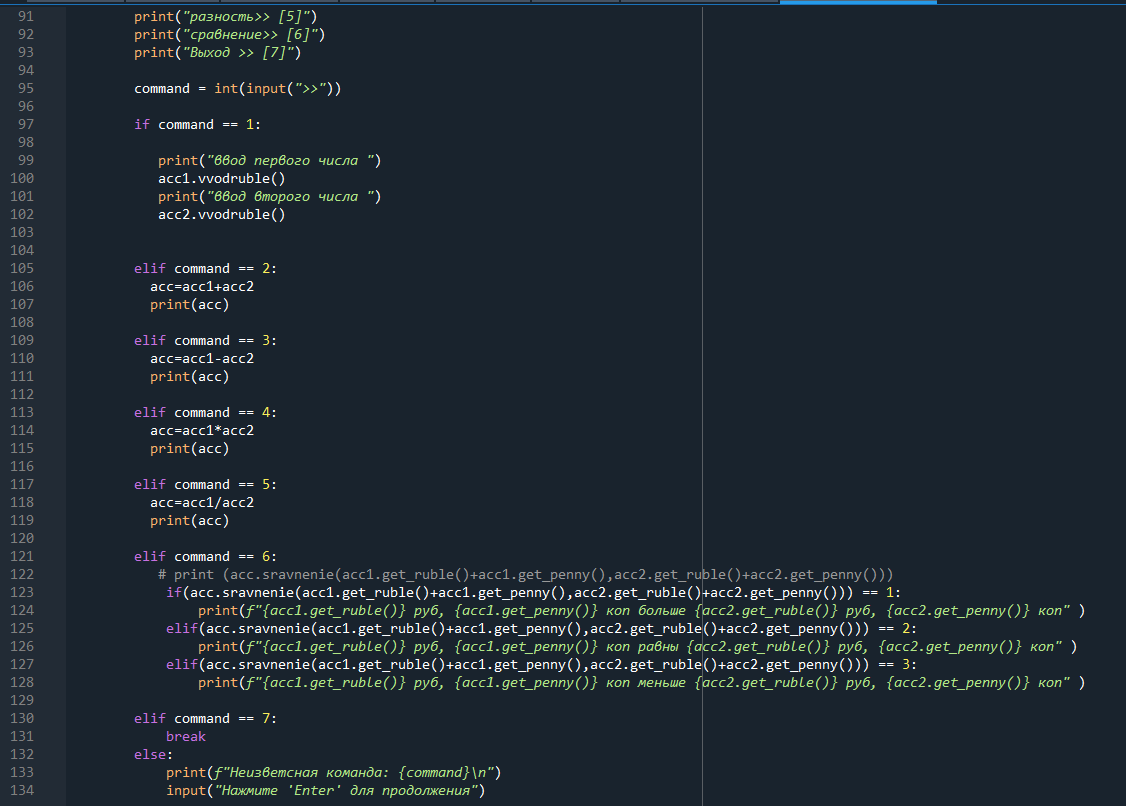


Рисунок 4 – листинг программы

Результат работы программы изображен на рисунках 5-10

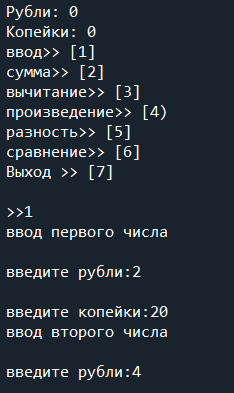


Рисунок 5 – Результат выполнения ввода чисел

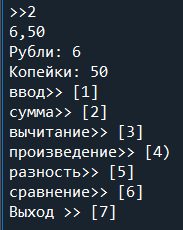


Рисунок 6 – Результат выполнения суммирования

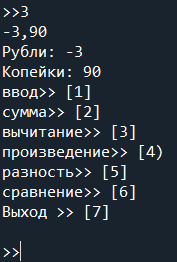


Рисунок 7 – Результат выполнения вычитания

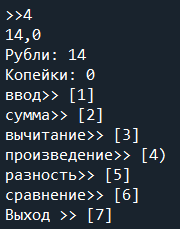


Рисунок 8 – Результат выполнения произведения

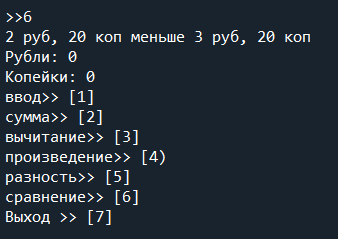


Рисунок 9 – Результат выполнения сравнения в первом случае

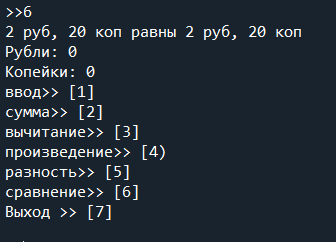


Рисунок 10 – Результат выполнения сравнения во втором случае

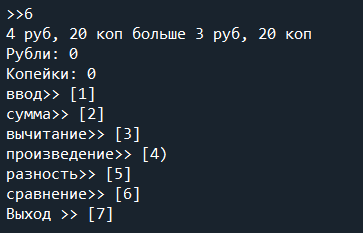


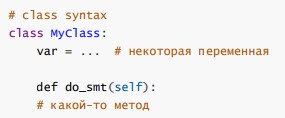
Рисунок 11 – Результат выполнения сравнения в третьем случае

**Вывод:** в процессе выполнения лабораторной работы, были приобретены навыки по работе с классами и объектами при написании программ с помощью языка программирования Python версии 3.x.

# Ответы на вопросы:

1. **Как осуществляется объявление класса в языке Python?**

Классы объявляются с помощью ключевого слова class и имени класса:



# Чем атрибуты класса отличаются от атрибутов экземпляра?

Атрибуты класса являются общими для всех объектов класса, а атрибуты экземпляра специфическими для каждого экземпляра. Более того, атрибуты класса определяются внутри класса, но вне каких-либо методов, а атрибуты экземпляра обычно определяются в методах, чаще всего в init .

# Каково назначение методов класса?

Методы определяют функциональность объектов, принадлежащих конкретному классу.

# Для чего предназначен метод init () класса?

Метод init является конструктором. Конструкторы - это концепция объектноориентированного программирования. Класс может иметь один и

только один конструктор. Если init определен внутри класса, он

автоматически вызывается при создании нового экземпляра класса.

# Каково назначение self ?

Аргумент self представляет конкретный экземпляр класса и позволяет нам получить доступ к его атрибутам и методам. Важно использовать параметр self внутри метода, если мы хотим сохранить значения экземпляра для последующего использования.

В большинстве случаев нам также необходимо использовать параметр self в других методах, потому что при вызове метода первым аргументом, который ему передается, является сам объект. Давайте добавим метод к нашему классу River и посмотрим, как он будет работать.

# Как добавить атрибуты в класс?

Атрибуты созданного экземпляра класса можно добавлять, изменять или удалять в любое время, используя для доступа к ним точечную запись. Если построить инструкцию, в которой присвоить значение атрибуту, то можно изменить значение, содержащееся внутри существующего атрибута, либо создать новый с указанным именем и содержащий присвоенное значение:

# Как осуществляется управление доступом к методам и атрибутам в языке Python?

Если вы знакомы с языками программирования Java, C#, C++ то, наверное, уже задались вопросом: “а как управлять уровнем доступа?”. В перечисленных языка вы можете явно указать для переменной, что доступ к ней снаружи класса запрещен, это делается с помощью ключевых слов (private, protected и т.д.). 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).

# Каково назначение функции isinstance ?

Встроенная функция isinstance(obj, Cls) , используемая при реализации методов арифметических операций и операций отношения, позволяет узнать что некоторый объект obj является либо экземпляром класса Cls либо экземпляром одного из потомков класса Cls.