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 № 4.1 «Элементы объектно-ориентированного программирования в языке Python»

по дисциплине «Основы программной инженерии»

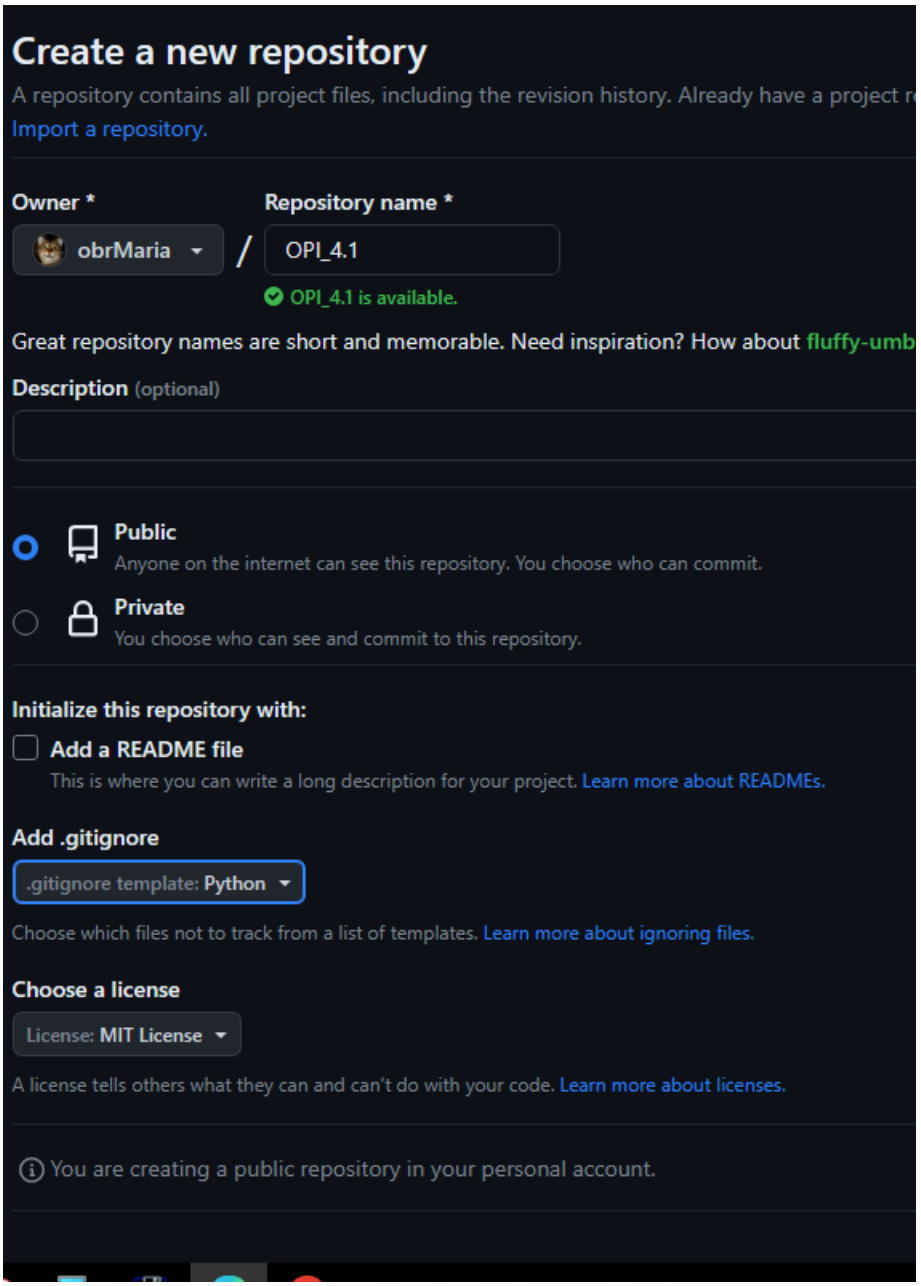
Выполнила:
Образцова Мария Дмитриевна,
2 курс, группа ПИЖ-б-о-21-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2023 г.

Цель работы: приобретение навыков по работе с классами и объектами 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.



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

```
M@DESKTOP-UVM9NOL MINGW64 ~/Desktop (master)
$ git clone https://github.com/obrMaria/OPI_4.1.git
Cloning into 'OPI_4.1'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.

M@DESKTOP-UVM9NOL MINGW64 ~/Desktop (master)
$ cd OPI_4.1
```

Рисунок – клонирование созданного репозитория

4. Проработать примеры лабораторной работы.

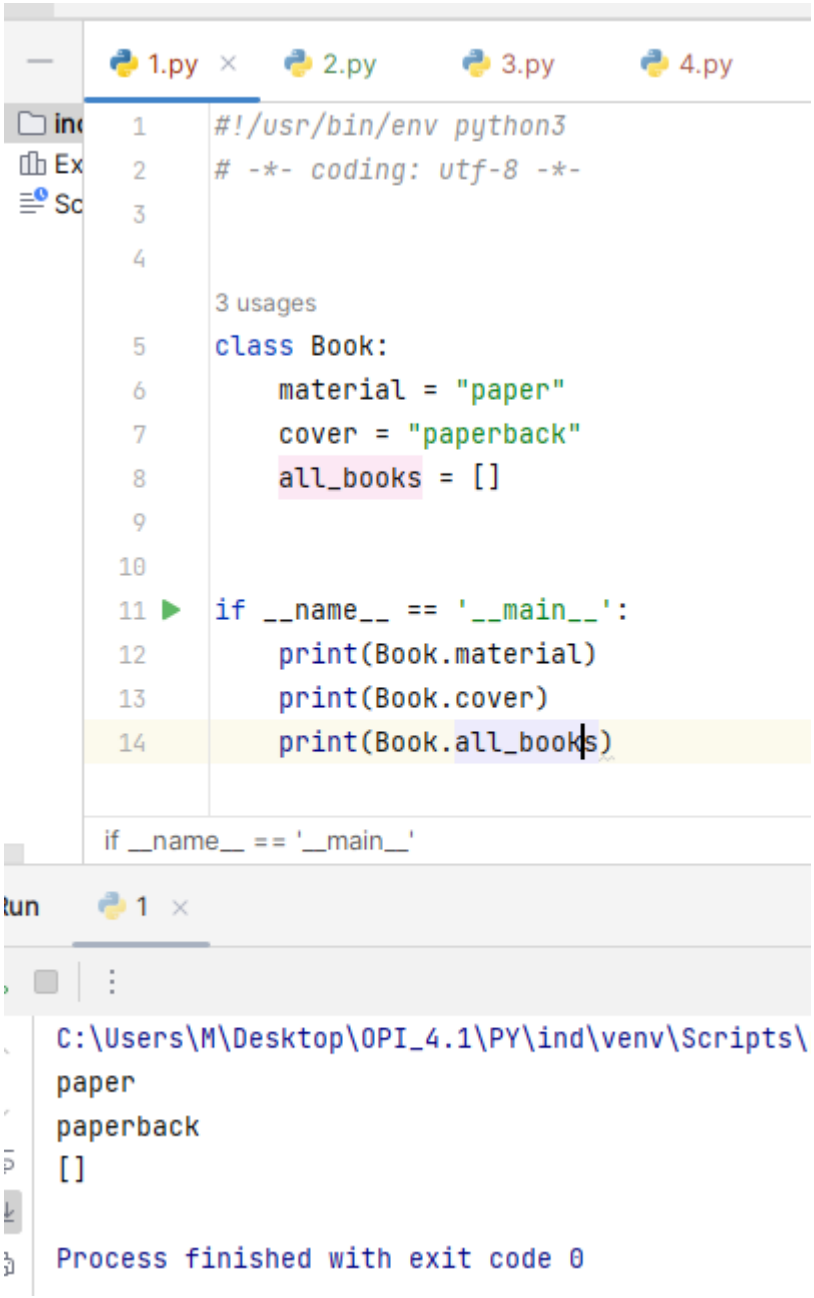


Рисунок – Атрибут класса

Рисунок – Атрибуты и эементы класса

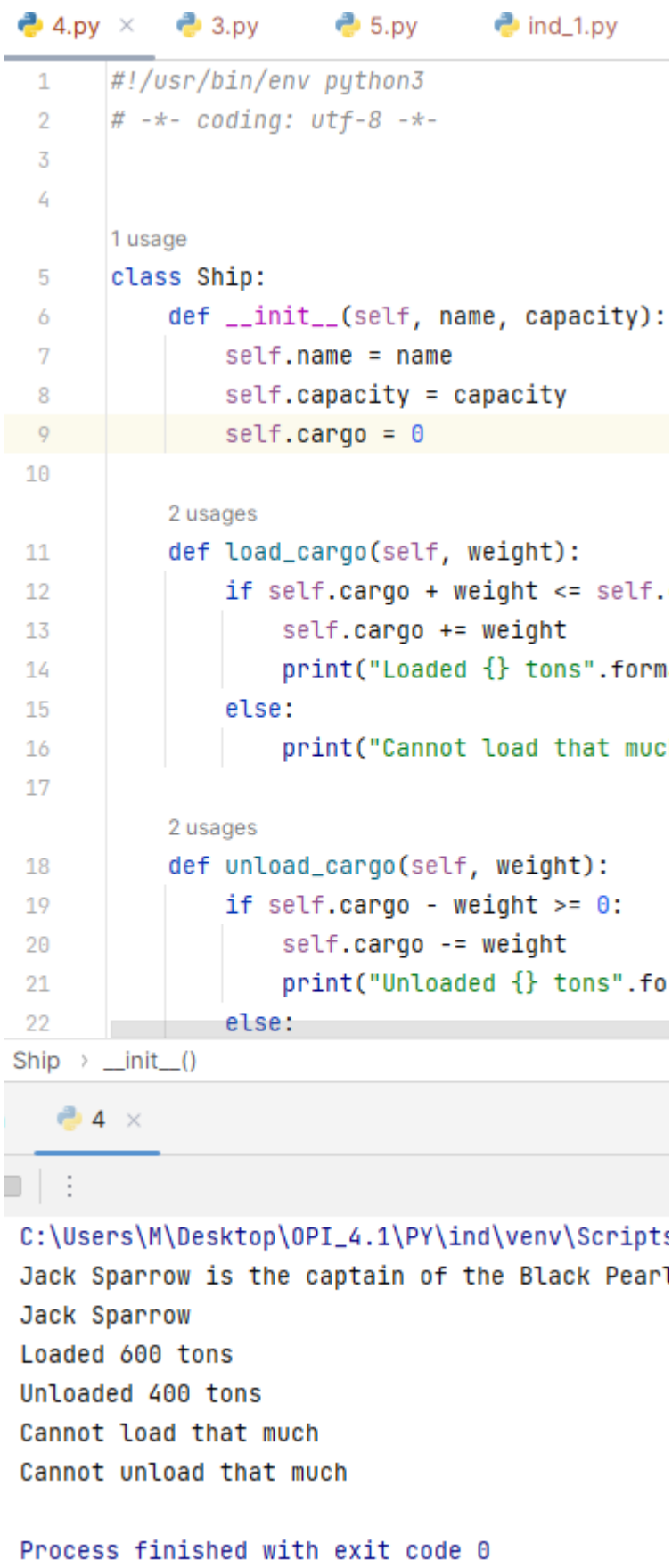


Рисунок – Изменение атрибутов с помощью методов

Рисунок – Свойства

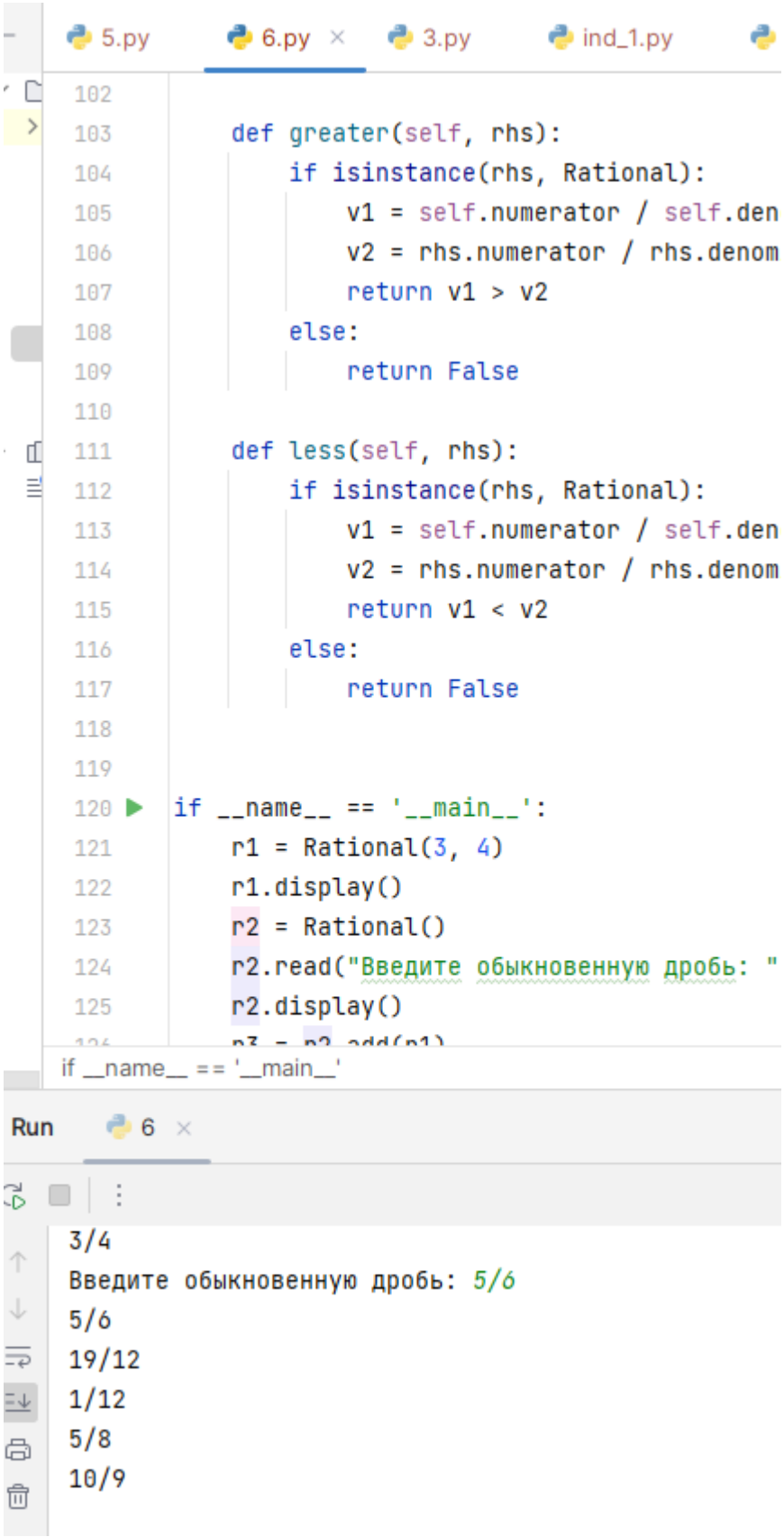


Рисунок – пример

5. Выполнить индивидуальные задания.

Задание 1

Парой называется класс с двумя полями, которые обычно имеют имена first и second. Требуется реализовать тип данных с помощью такого класса. Во всех заданиях обязательно должны присутствовать:

- метод инициализации __init__ ;
- метод должен контролировать значения аргументов на корректность; ввод с клавиатуры read ;
- вывод на экран display .

Реализовать внешнюю функцию с именем make_тип() , где тип — тип реализуемой структуры. Функция должна получать в качестве аргументов значения для полей структуры и возвращать структуру требуемого типа. При передаче ошибочных параметров следует выводить сообщение и заканчивать работу.

В раздел программы, начинающийся после инструкции if __name__ = '__main__': добавить код, демонстрирующий возможности разработанного класса.

1.(21) Поле first — дробное число; поле second — целое число, показатель степени. Реализовать метод power() — возведение числа first в степень second. Метод должен правильно работать при любых допустимых значениях first и second.

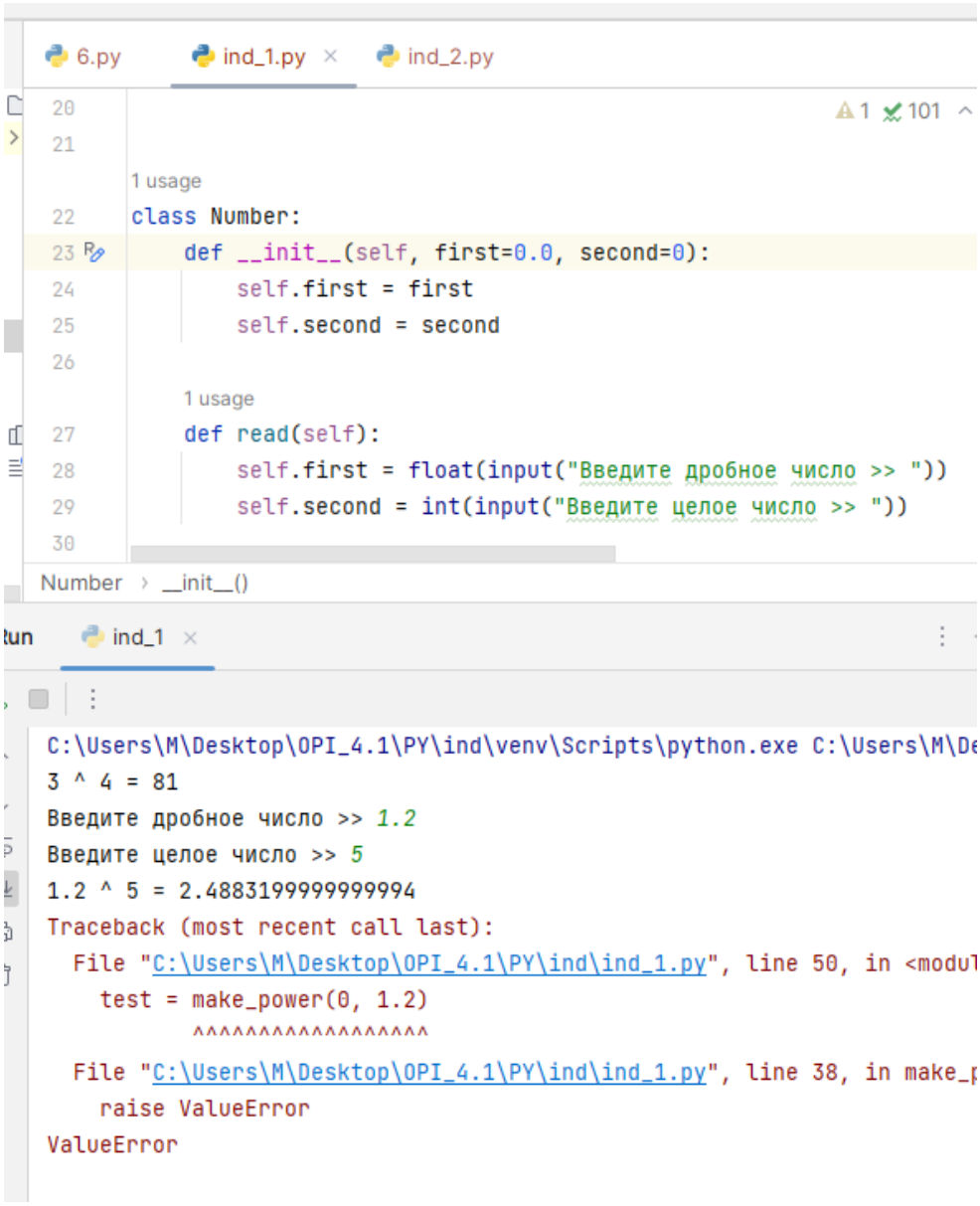


Рисунок – Индивидуальное задание №1

Задание 2

Составить программу с использованием классов и объектов для решения задачи. Во всех заданиях, помимо указанных в задании операций, обязательно должны быть реализованы следующие методы:

- метод инициализации __init__ ;
- ввод с клавиатуры read ;
- вывод на экран display .

В раздел программы, начинающийся после инструкции if __name__ =

'__main__': добавить код, демонстрирующий возможности разработанного класса.

6.(21) Создать класс Point для работы с точками на плоскости. Координаты точки — декартовы. Обязательно должны быть реализованы: перемещение точки по оси , перемещение по оси , определение расстояния до начала координат, расстояния между двумя точками, преобразование в полярные координаты, сравнение на совпадение и несовпадение.

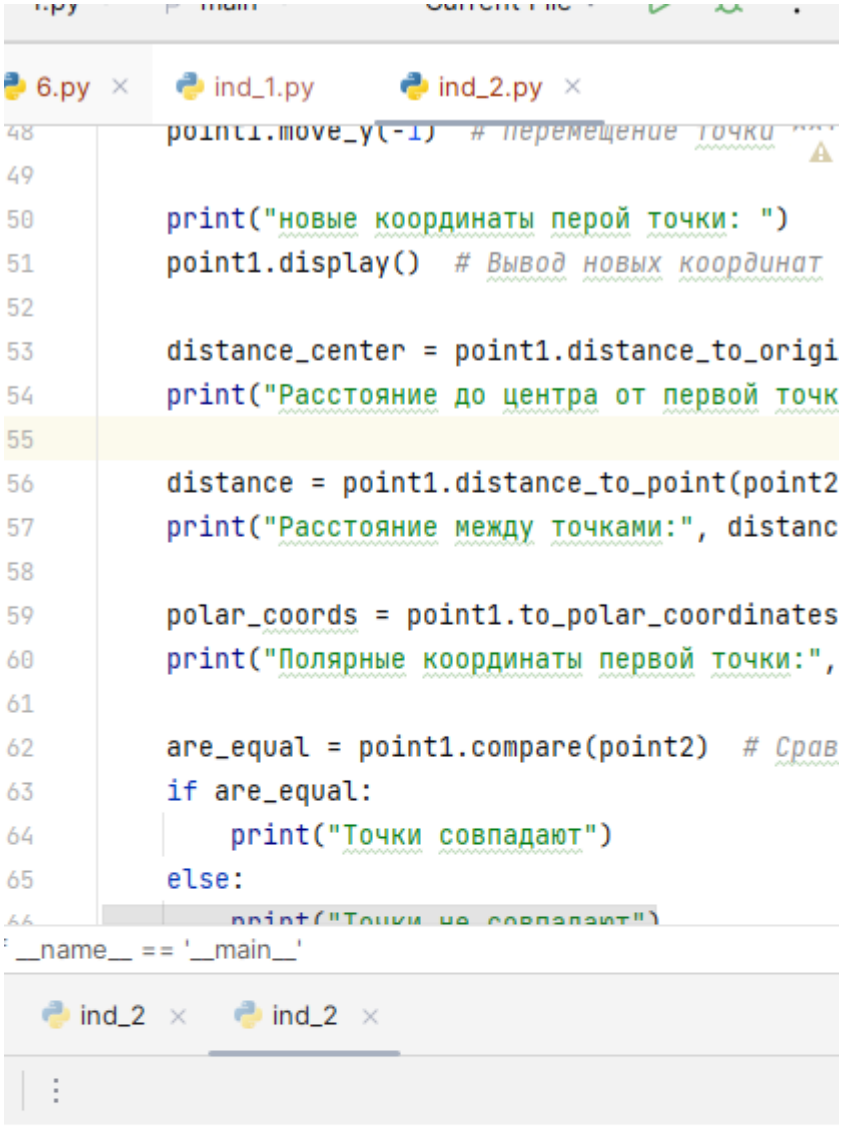


Рисунок – Индивидуальное задание №2

ВОПРОСЫ

1. Как осуществляется объявление класса в языке Python?

Классы объявляются с помощью ключевого слова class и имени класса: class MyClass:

var = ... # некоторая переменная def do_smt(self):

какой-то метод

2. Чем атрибуты класса отличаются от атрибутов экземпляра? Атрибут класса - это атрибут, общий для всех экземпляров класса.

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.

Атрибуты экземпляра определяются в методах и хранят информацию, специфичную для экземпляра.

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. Метод

_____init__указывает, какие атрибуты будут у экземпляров нашего класса.

5. Каково назначение self ?

Аргумент self представляет конкретный экземпляр класса и позволяет нам получить доступ к его атрибутам и методам. В примере с____init__мы создаем атрибуты для конкретного экземпляра и присваиваем им значения

аргументов метода. 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.

6. Как добавить атрибуты в класс?

Новый атрибут класса указывается через точку после названия класса, затем ему присваивается определенное значение.

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

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).

Если же атрибут или метод начинается с двух подчеркиваний, то тут напрямую вы к нему уже не обратитесь (простым образом).

8. 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.