

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
«Элементы объектно-ориентированного программирования в языке
Python»

Отчет по лабораторной работе № 4.1
по дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-21-1

Стригалов Дмитрий.

«10» октября 2023г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

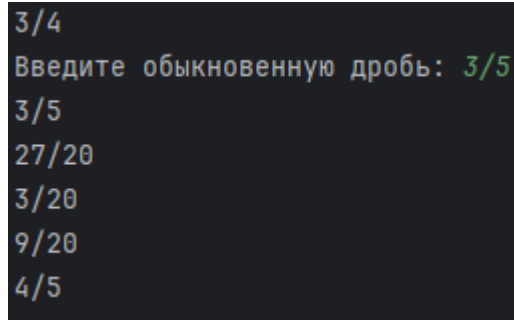
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

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

Порядок выполнения работы:

Проработать примеры лабораторной работы.



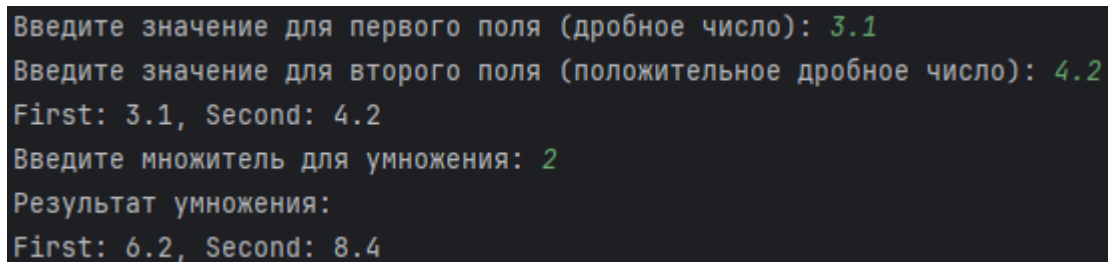
```
3/4
Введите обыкновенную дробь: 3/5
3/5
27/20
3/20
9/20
4/5
```

Рисунок 1 - Результат выполнения примера

Выполнить индивидуальные задания.

Задание 1. Вариант 16

Поле `first` — дробное число, целая часть числа; поле `second` — положительное дробное число, дробная часть числа. Реализовать метод `multiply()` — умножение на произвольное дробное число типа `double`. Метод должен правильно работать при любых допустимых значениях `first` и `second`.



```
Введите значение для первого поля (дробное число): 3.1
Введите значение для второго поля (положительное дробное число): 4.2
First: 3.1, Second: 4.2
Введите множитель для умножения: 2
Результат умножения:
First: 6.2, Second: 8.4
```

Рисунок 2 - Результат выполнения индивидуального задания 1

Задание 2. Вариант 16

Создать класс `Vector3D`, задаваемый тройкой координат. Обязательно должны быть реализованы: сложение и вычитание векторов, скалярное произведение векторов, умножение на скаляр, сравнение векторов, вычисление длины вектора, сравнение длины векторов.

```
Координаты - 1.0, 2.0, 3.0
Введите координаты: 1.0 2.0 3.0
Координаты - 1.0, 2.0, 3.0
Координаты - 2.0, 4.0, 6.0
Координаты - 0.0, 0.0, 0.0
Координаты - 1.0, 4.0, 9.0
Координаты - 2.0, 4.0, 6.0
Векторы совпадают
3.7416573867739413
Длины векторов равны
```

Рисунок 3 - Результат выполнения индивидуального задания 2

Контрольные вопросы:

1. Как осуществляется объявление класса в языке Python?

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

```
# class syntax 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?

Классы позволяют определять данные и поведение похожих объектов. Поведение описывается методами. Метод похож на функцию тем, что это блок кода, который имеет имя и выполняет определенное действие. Методы, однако, не являются независимыми, поскольку они определены внутри класса.

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. 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?

Например, мы хотим видеть информацию о всех видах наших питомцев. Мы могли бы записать ее в самом классе с самого начала или создать переменную следующим образом:

```
Pet.all_specs = [tom.сpec, avocado.сpec, ben.сpec]
```

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 мешает 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` ?

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`.

Вывод: были приобретены навыки по работе с классами и объектами при написании программ с помощью языка программирования Python версии 3.x.