

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Лабораторная работа 1

Элементы объектно-ориентированного
программирования в языке Python

Выполнил студент группы ИВТ-б-о-20-1

Симанский М.Ю « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

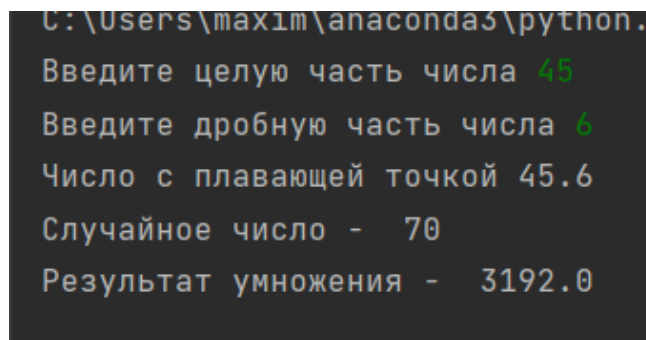
Проверил Воронкин Р.А. _____

(подпись)

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

Задание 1

Поле `first` — целое число, целая часть числа; поле `second` — положительное целое число, дробная часть числа. Реализовать метод `multiply()` — умножение на произвольное целое число типа `int`. Метод должен правильно работать при любых допустимых значениях `first` и `second`. Результат выполнения отображен на рисунке 1.



```
C:\Users\maxim\anaconda3\python.
Введите целую часть числа 45
Введите дробную часть числа 6
Число с плавающей точкой 45.6
Случайное число - 70
Результат умножения - 3192.0
```

Рисунок 1 — Результат выполнения

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. Результат выполнения отображен на рисунке 2.

```
Сумма - 35,50 рублей
Введите количество рублей 45
Введите количество копеек 50
Сумма - 45,50 рублей
81.0 рублей в сумме
10.0 - разница между 2 суммами
Результат деления - 1.28
Введите дробь в виде Число/Число 3/7
Результат умножения на дробь 19.5
Вторая сумма больше
```

Рисунок 2 – Результат выполнения

Ответы на контрольные вопросы

1. Как осуществляется объявление класса в языке Python? - Классы объявляются с помощью ключевого слова `class` и имени класса.
2. Чем атрибуты класса отличаются от атрибутов экземпляра? - 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. Что касается переменных экземпляра, они хранят данные, уникальные для каждого объекта класса. В этой теме мы рассмотрим только атрибуты класса, но не волнуйтесь, у вас будет достаточно времени, чтобы узнать больше и об атрибутах экземпляра.
3. Каково назначение методов класса? - Методы определяют функциональность объектов, принадлежащих конкретному классу.
4. Для чего предназначен метод `__init__()` класса? – Для объявления конструктора класса.
5. Каково назначение `self`? – Указание объекта на самого себя.
6. Как добавить атрибуты в класс? - Атрибуты экземпляра - это как раз те, которые мы определяем в методах, поэтому по определению мы можем создавать новые атрибуты внутри наших пользовательских методов.

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

7. В Python есть встроенная функция `instance ()`, которая сравнивает значение с указанным типом. Если данное значение и тип соответствуют, он вернет `true`, иначе `false`. Используя `isinstance ()`, вы можете проверить строку, число с плавающей точкой, `int`, список, кортеж, `dict`, `set`, `class` и т. Д.