

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии  
Департамент цифровых, робототехнических систем и электроники

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**  
**дисциплины «Объектно-ориентированное программирование»**  
**Вариант 3**

Выполнил:  
Говоров Егор Юрьевич  
3 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем»,  
очная форма обучения

---

(подпись)

Проверил:  
Воронкин Роман Александрович,  
доцент департамента цифровых,  
робототехнических систем и  
электроники

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

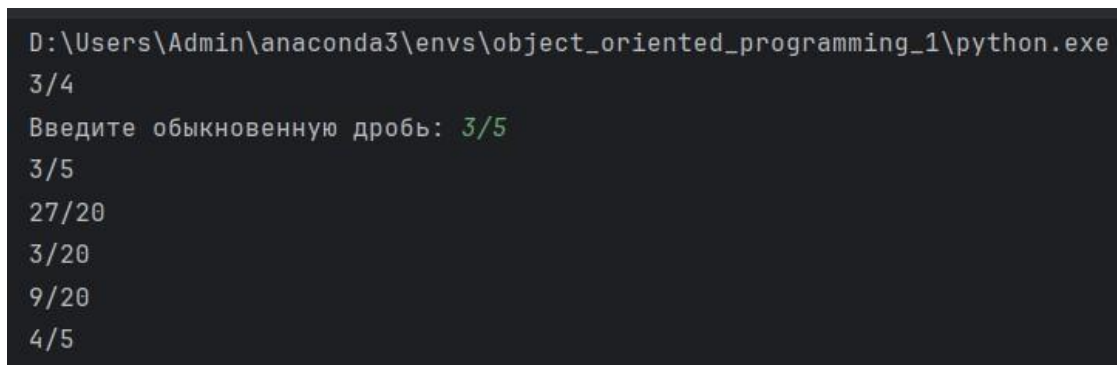
Ставрополь, 2024 г.

Тема: элементы объектно-ориентированного программирования в языке Python.

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

Ход работы:

1. Проработал пример:



```
D:\Users\Admin\anaconda3\envs\object_oriented_programming_1\python.exe
3/4
Введите обыкновенную дробь: 3/5
3/5
27/20
3/20
9/20
4/5
```

Рисунок 1. Результат работы примера

2. Выполнил индивидуальное задание 1 Вариант 3:

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 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.

Поле `first` — целое положительное число, числитель; поле `second` — целое положительное число, знаменатель. Реализовать метод `ipart()` —

выделение целой части дроби first/second. Метод должен проверять равенство знаменателя нулю.

```
D:\Users\Admin\anaconda3\envs\object_oriented_programming_1\python.exe
5/3
Целая часть дроби: 1
Введите дробь в формате числитель/знаменатель: 64/8
64/8
Целая часть дроби: 8
```

Рисунок 2. Результат работы индивидуального задания 1

### 3. Выполнил индивидуальное задание 2 Вариант 3:

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.

```
D:\Users\Admin\anaconda3\envs\object_oriented_programming_1\python.exe
100,50
Введите сумму в формате рубли,копейки: 300,65
300,65
Сумма двух сумм:
401,15
Разность двух сумм:
200,15
Умножение на 1.5:
150,75
Деление на 2:
50,25
Сравнение сумм:
money1 == money2: False
money1 < money2: True
money1 <= money2: True
```

Рисунок 3. Результат работы индивидуального задания 2

## Ответы на контрольные вопросы:

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

Класс объявляется с помощью ключевого слова `class`, за которым следует имя класса.

### 2. Чем атрибуты класса отличаются от атрибутов экземпляра?

Атрибуты класса общие для всех экземпляров класса, а атрибуты экземпляра принадлежат конкретному объекту и могут быть уникальными для каждого экземпляра.

### 3. Каково назначение методов класса?

Методы класса выполняют действия или операции, связанные с объектами класса, и могут изменять атрибуты экземпляра или взаимодействовать с данными.

### 4. Для чего предназначен метод `__init__()` класса?

Метод `__init__()` используется для инициализации объекта при его создании, задавая начальные значения атрибутов экземпляра.

### 5. Каково назначение `self`?

`self` указывает на текущий экземпляр класса, позволяя обращаться к атрибутам и методам этого экземпляра внутри класса.

### 6. Как добавить атрибуты в класс?

Атрибуты экземпляров добавляются в любом методе через `self`: `self.attr = value` или вне класса для определенного объекта: `obj.new_attr = value`

Атрибуты класса создаются напрямую в теле класса:

```
class MyClass:
    class_attr = value
```

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

Атрибуты, начинающиеся с одного подчеркивания (`_attr`), считаются защищенными, но к ним можно получить доступ извне, если же атрибут начинается с двух подчеркиваний (`__attr`), то прямого доступа снаружи к нему

не будет, но через `_ClassName_attr` к нему все еще можно будет получить доступ.

#### 8. Каково назначение функции `isinstance`?

Функция `isinstance()` проверяет, является ли объект экземпляром определенного класса или его подклассов.

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