

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

**Отчет по лабораторной работе № 4.1
«Элементы объектно-ориентированного программирования в языке
Python»**

по дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-20-1

Дыбов Д.В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

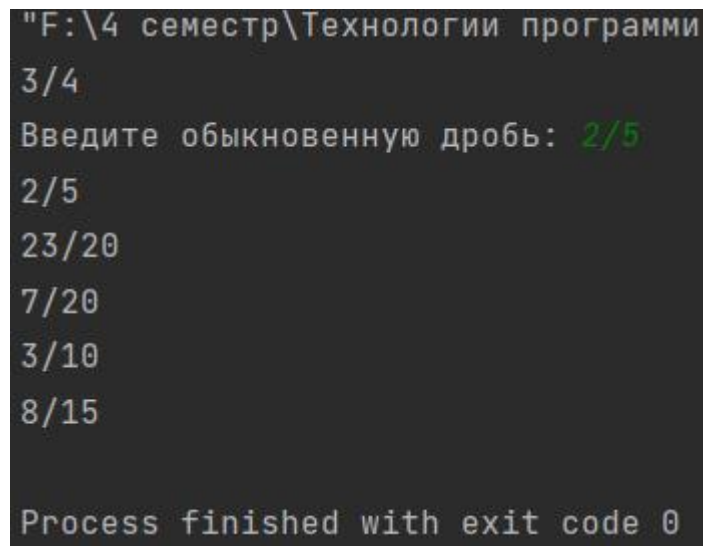
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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

Ход Работы

1. Создал новый репозиторий для лабораторной работы №4.1;
2. Клонировал созданный репозиторий на компьютер;
3. Создал новый PyCharm проект в папке репозитория;
4. Проработал пример:



```
"F:\4 семестр\Технологии программы
3/4
Введите обыкновенную дробь: 2/5
2/5
23/20
7/20
3/10
8/15

Process finished with exit code 0
```

Рисунок 1 – Результат выполнения примера

5. Выполнил первое индивидуальное задание;
6. Проверил индивидуальное задание на работоспособность:

```
"F:\4 семестр\Технологии программирования и алгоритмиз
Введите целое, положительное число, числитель: 26
Введите целое, положительное число, знаменатель: 5
Целая часть дроби 26 / 5:
5
Process finished with exit code 0
```

Рисунок 2 – Результат выполнения первого индивидуального задания

7. Выполнил второе индивидуальное задание;

8. Проверил индивидуальное задание на работоспособность:

```
"F:\4 семестр\Технологии программирования и алгоритмизации\lab
Денежная сумма по умолчанию: 40, 20
Введите рубли: 25
Введите копейки: 60
Вы ввели денежную сумму: 25, 60
Сумма: 65, 80
Вычитание денежных сумм: 14, 60
Умножение сумм: 1012, 0
Деление сумм: 1, 0
Деление суммы на дробное число: 1, 3
Умножение суммы на дробное число: 514, 6
Сумма 40 больше 25
```

Рисунок 3 – Результат выполнения второго индивидуального задания

9. С помощью сайта проверил пример на наличие ошибок;

10. Результат не выдал ошибок;

100%

[Guido](#)

Lines: 149
Hints: 0

PythonChecker Makes Your Code Great Again

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

class Rational:

    def __init__(self, a=0, b=1):
        a = int(a)
        b = int(b)
```

Рисунок 4 – Проверка кода на наличие ошибок

11. Проверил первое индивидуальное задание на наличие ошибок;

100%

[Guido](#)

Lines: 44
Hints: 0

PythonChecker Makes Your Code Great Again

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
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.
"""
```

Рисунок 5 – Проверка первого индивидуального задания на наличие ошибок

11. Проверил первое индивидуальное задание на наличие ошибок;

100%

[Guido](#)

Lines: 149
Hints: 0

PythonChecker Makes Your Code Great Again

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
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.
"""
```

Рисунок 6 – Проверка индивидуального задания на наличие ошибок

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.

3. Каково назначение методов класса?

Методы определяют функциональность объектов, принадлежащих данному классу. Методы 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.

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. На атрибуты данных класса могут ссылаться как методы, так и обычные пользователи - "клиенты" объекта.

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

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.

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