

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
Яковлева Елизавета Андреевна.

« »_____2023г.

Подпись студента_____

Работа защищена « »_____20__г.

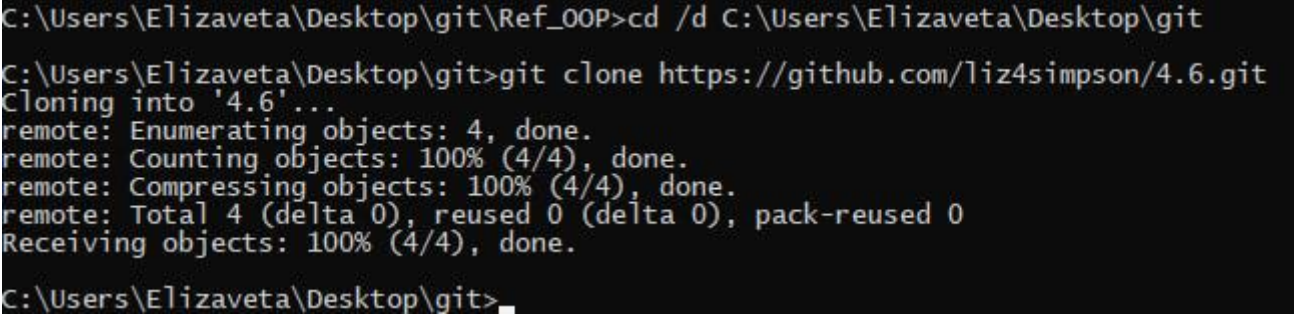
Проверил Воронкин Р.А. _____
(подпись)

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:

1. Выполните клонирование созданного репозитория.



```
C:\Users\Elizaveta\Desktop\git\Ref_00P>cd /d C:\Users\Elizaveta\Desktop\git  
C:\Users\Elizaveta\Desktop\git>git clone https://github.com/liz4simpson/4.6.git  
Cloning into '4.6'...  
remote: Enumerating objects: 4, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (4/4), done.  
C:\Users\Elizaveta\Desktop\git>
```

Рисунок 2 - Клонирование репозитория

2. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

```

C:\Users\Elizaveta\Desktop\git\Ref_00P>cd /d C:\Users\Elizaveta\Desktop\git
C:\Users\Elizaveta\Desktop\git>git clone https://github.com/liz4simpson/4.6.git
Cloning into '4.6'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
C:\Users\Elizaveta\Desktop\git>

```

Рисунок 3 - Ветвление по модели git-flow

3. Проработать примеры лабораторной работы.

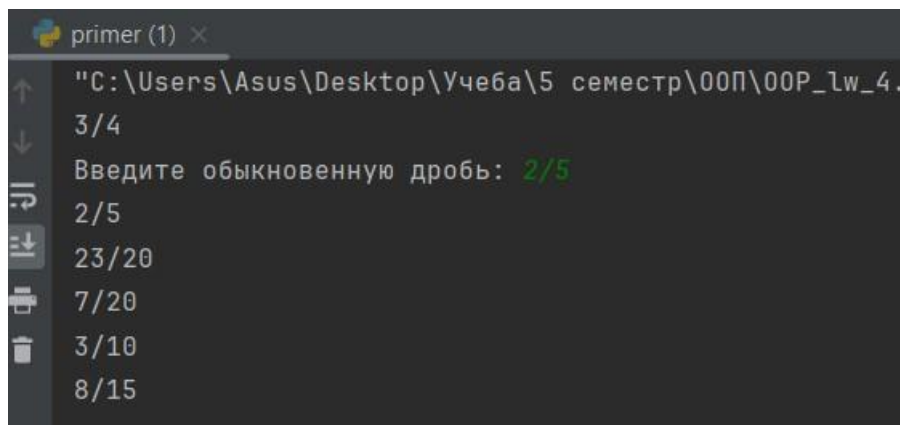


Рисунок 4 - Результат выполнения примера

4. Выполнить индивидуальные задания.

Задание 1. Вариант 22

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`.

```

Enter the first number: 2
Enter the second number: 3
Pair: first = 2.0, second = 3.0, result = 8.0

```

Рисунок 5 - Результат выполнения индивидуального задания 1

Задание 2. Вариант 22(5)

Создать класс `Angle` для работы с углами на плоскости, задаваемыми величиной в градусах и минутах. Обязательно должны быть реализованы: перевод в радианы, приведение к диапазону 0-360, увеличение и уменьшение угла на заданную величину, получение синуса, сравнение углов.

```
Введите угол: 40 0
Введите угол для сравнения: 50 8
40.0 меньше 50.13333333333333
В радианах: 0.6981317007977318
В диапазоне 0-360: 40.0
Синус: 0.6427876096865393
Изменение на определенный угол: 70.0
Сравнение с другим углом: 40.0 меньше 50.13333333333333
```

Рисунок 6 - 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.spec, avocado.spec, ben.spec]
```

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.