

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №4.3

Наследование и полиморфизм в языке Python

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2022

Цель работы: приобретение навыков по иерархии классов при написании программ с помощью языка программирования Python версии 3.x.

Ход работы: Примеры

<https://github.com/Mirror-Shard/L4.3>

1. Изучил теоретический материал и приступил к выполнению примеров:

```
3/4
Введите обыкновенную дробь: 5/6
5/6|
19/12
1/12
5/8
10/9
```

Рисунок 1 – Результат сложения, вычитания, умножения обыкновенных дробей

```
I have 3 sides
I have 4 sides
I have 5 sides
I have 6 sides
```

Рисунок 2 – Работа абстрактных классов

```
I can walk and run
I can crawl
I can bark
I can roar
```

Рисунок 3 – Работа примера 3

Задание 1

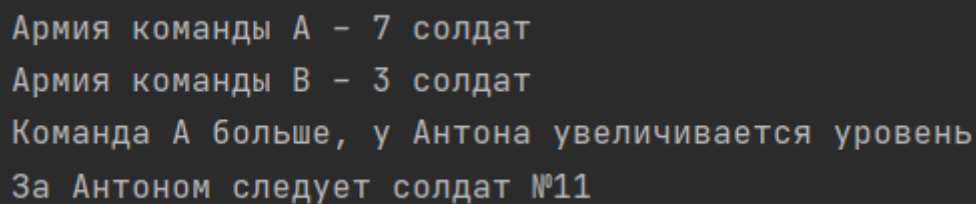
В некой игре-стратегии есть солдаты и герои. У всех есть свойство, содержащее уникальный номер объекта, и свойство, в котором хранится принадлежность команде. У солдат есть метод "иду за героем", который в качестве аргумента принимает объект типа "герой". У героев есть метод увеличения собственного уровня.

В основной ветке программы создается по одному герою для каждой команды. В цикле генерируются объекты-солдаты. Их принадлежность команде определяется случайно. Солдаты разных команд добавляются в разные списки.

Измеряется длина списков солдат противоборствующих команд и выводится на экран. У героя, принадлежащего команде с более длинным списком, увеличивается уровень.

Отправьте одного из солдат первого героя следовать за ним. Выведите на экран идентификационные номера этих двух юнитов.

1. Программа создаёт два героя, армия солдат, которые случайно распределяются по героям. У героя армия которого больше, увеличивается уровень:



```
Армия команды А - 7 солдат  
Армия команды В - 3 солдат  
Команда А больше, у Антона увеличивается уровень  
За Антоном следует солдат №11
```

Рисунок 4 – Обработка ошибки

Индивидуальное задание 1

Создать класс Triangle с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс RightAngled (прямоугольный), имеющий поле площади. Определить метод вычисления площади.

2. Реализовал функцию вычисления площади треугольника в производном классе которой не было в базовом:

```
Введите сторону A:3
Введите сторону B:3
Введите сторону C:5
Периметр треугольника 11
Углы треугольника (0.586, 0.586, 1.97)
Площадь треугольника 4.14578098794425
```

Рисунок 5 – Вычисление углов, периметра и площади

Индивидуальное задание 2

Создать абстрактный базовый класс Function (функция) с виртуальными методами вычисления значения функции в заданной точке и вывода результата на экран. Определить производные классы Ellipse (эллипс), Hyperbola (гипербола) с собственными функциями вычисления y в зависимости от входного параметра.

1. Создал в программе два производных класса из абстрактного базового класса Function, где переопределил функцию calculation:

```
Введите a: 2
Введите b: 3
Введите x: 4
Для эллипса y равен 6.0
Для гиперболы y равен (6.0, -6.0)
```

Рисунок 6 – Работа с абстрактными классами

Контрольные вопросы:

1. Что такое наследование как оно реализовано в языке Python?

Наследование — это возможность расширения (наследования) ранее написанного программного кода класса с целью дополнения, усовершенствования или привязки под новые требования.

Синтаксически создание класса с указанием его родителя выглядит так:

`class имя_класса(имя_родителя1, [имя_родителя2,..., имя_родителя_n])`

2. Что такое полиморфизм и как он реализован в языке Python?

Полиморфизм - это способность выполнять действие над объектом независимо от его типа. Это обычно реализуется путем создания базового класса и наличия двух или более подклассов, которые все реализуют методы с одинаковой сигнатурой.

3. Что такое "утиная" типизация в языке программирования Python?

Эта концепция адаптирована из следующего абдуктивного умозаключения:

Если что-то выглядит как утка, плавает как утка и крикает как утка, это наверняка и есть утка. Концепция утиной типизации в основном принята в языках программирования, поддерживающих динамическую типизацию, таких как Python и JavaScript. Общей особенностью этих языков является возможность объявления переменных без указания их типа.

При использовании пользовательских типов для определённых целей, реализация связанных функций важнее, чем точные типы данных. Утиная типизация подчёркивает реализацию связанных выполняемых функций, а конкретные типы данных менее важны

4. Каково назначение модуля abc языка программирования Python?

Начиная с версии языка 2.6 в стандартную библиотеку включается модуль abc, добавляющий в язык абстрактные базовые классы (далее АБК). АБК позволяют определить класс, указав при этом, какие методы или свойства обязательно переопределить в классах-наследниках.

5. Как сделать некоторый метод класса абстрактным?

Перед методом класса необходимо добавить декоратор модуля abc: `@abstractmethod`.

6. Как сделать некоторое свойство класса абстрактным?

Абстрактные классы включают в себя атрибуты в дополнение к методам, вы можете потребовать атрибуты в конкретных классах, определив их с помощью `@abstractproperty`.

7. Каково назначение функции `isinstance` ?

Функция `isinstance ()` в Python используется для проверки, является ли объект экземпляром указанного класса или нет.

Вывод: в ходе работы приобрёл навыки по работе с иерархией классов с помощью языка программирования Python версии 3.