

**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 «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-20-1

Дыбов Д.В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

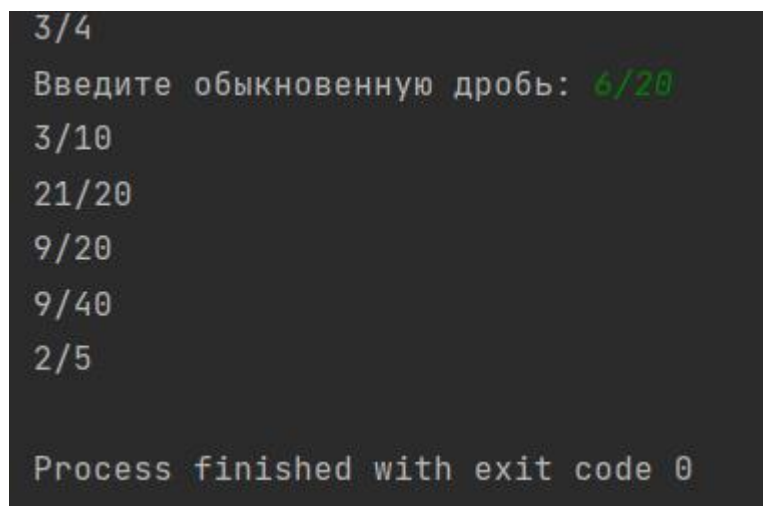
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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

Ход Работы

1. Создал новый репозиторий для лабораторной работы №4.3;
2. Клонировал созданный репозиторий на компьютер;
3. Создал новый PyCharm проект в папке репозитория;
4. Проработал первый пример:

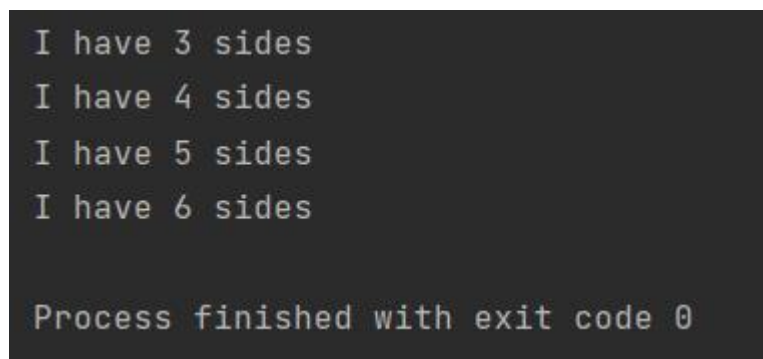


```
3/4
Введите обыкновенную дробь: 6/20
3/10
21/20
9/20
9/40
2/5

Process finished with exit code 0
```

Рисунок 1 – Результат выполнения первого примера

5. Проработал второй пример:



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

Process finished with exit code 0
```

Рисунок 2 – Результат выполнения второго примера

6. Проработал третий пример:

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

Process finished with exit code 0
```

Рисунок 3 – Результат выполнения третьего примера

7. Приступил к выполнению первого индивидуального задания;

8. Проверил задание на работоспособность:

```
Спирт: Метиловый, плотность спирта: 0.812 г/см^3, крепость спирта: 80%
Спирт: Этиловый, плотность спирта: 0.806 г/см^3
Спирт: Пропиловый, плотность спирта: 0.816 г/см^3, крепость спирта: 60%
```

Рисунок 4 – Результат выполнения первого индивидуального задания

9. Выполнил второе индивидуальное задание;

10. Проверил второе задание на работоспособность:

```
14.035668847618199
0.7713892158398701

Process finished with exit code 0
```

Рисунок 5 – Результат выполнения второго индивидуального задания

11. Выполнил задачу;

12. Проверил работоспособность задачи:

```
Солдат с номером 2 добавлен в армию Red
Солдат с номером 3 добавлен в армию Red
Солдат с номером 4 добавлен в армию Red
Солдат с номером 5 добавлен в армию Blue
Солдат с номером 6 добавлен в армию Red
Солдат с номером 7 добавлен в армию Blue
Солдат с номером 8 добавлен в армию Red
Солдат с номером 9 добавлен в армию Blue
Солдат с номером 10 добавлен в армию Red
Солдат с номером 11 добавлен в армию Blue
Армия героя Blue : 4
Армия героя Red : 6
В армии Red больше солдат, увеличиваем его уровень
Уровень героя Red увеличен на 1 и равен 2
Солдат номер 7 следует за героем Red с номером 2

Process finished with exit code 0
```

Рисунок 6 – Результат выполнения задачи

13. Проверил первый пример на наличие ошибок:

100%

[Guido](#)

Lines: 150
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 7 – Проверка первого примера на наличие ошибок

14. Проверил второй пример на наличие ошибок:

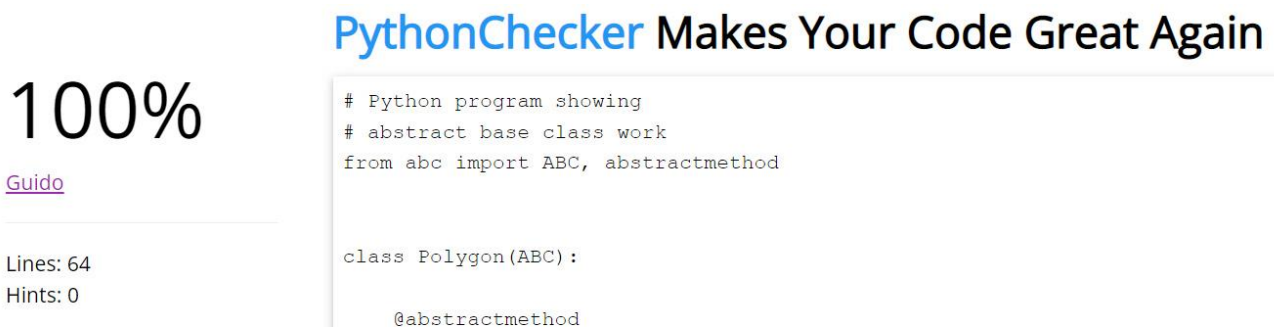


Рисунок 8 – Проверка второго примера на наличие ошибок

15. Проверил третий пример на наличие ошибок:

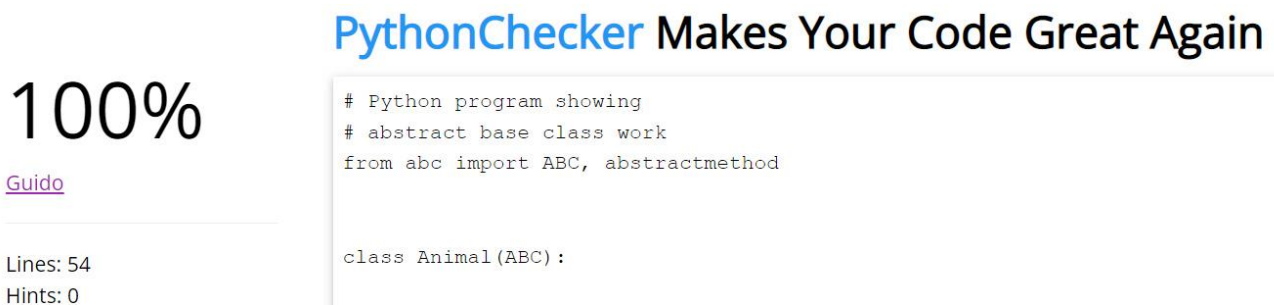


Рисунок 9 – Проверка третьего примера на наличие ошибок

16. Проверил первое индивидуальное задание на наличие ошибок:



Рисунок 10 – Проверка первого индивидуального задания на наличие ошибок

17. Проверил второе индивидуальное задание на наличие ошибок:

PythonChecker Makes Your Code 100% Great Again

[Guido](#)

Lines: 74

Hints: 0

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

"""
Создать абстрактный базовый класс Function (функция) с
виртуальными методами
вычисления значения функции  $y = f(x)$  в заданной точке и
```

Рисунок 11 – Проверка второго индивидуального задания на наличие ошибок

18. Проверил задачу на наличие ошибок:

PythonChecker Makes Your Code 100% Great Again

[Guido](#)

Lines: 63

Hints: 0

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

"""
В некой игре-стратегии есть солдаты и герои. У всех есть
свойство, содержащее уникальный
```

Рисунок 12 – Проверка задачи на наличие ошибок

Ответы на 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 крякает как утка, это наверняка и есть утка. Концепция утиной типизация 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 ABC). ABC 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 выполнения лабораторной работы были приобретены простейшие навыки по работе с наследованием и абстрактными методами классов в языке программирования Python.