

Отчёт

Лабораторная работа №3

Python. Объектно-ориентированные возможности.

Студент: Белкина Е.В.

Группа ИУ5-51

Преподаватель: Гапанюк Ю.В.

Москва, 2019

Задание:

Необходимо разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке Python 3.
2. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab_python_oop.
3. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab_python_oop.
 1. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь - <https://docs.python.org/3/library/abc.html>
 2. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь - <https://docs.python.org/3/library/functions.html#property>
 3. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры.
 4. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math - <https://docs.python.org/3/library/math.html>
 5. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны.
 6. Для классов «Прямоугольник», «Квадрат», «Круг»:
 1. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
 2. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
4. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов (используйте следующую конструкцию - <https://docs.python.org/3/library/main.html>). Создайте следующие объекты и выведите о них информацию в консоль:
 1. Прямоугольник синего цвета шириной 3 и высотой 2.
 2. Круг зеленого цвета радиусом 5.
 3. Квадрат красного цвета со стороной 5.

Исходный код:

Lab1 main.py

```
from lab_python_oop.Rectangle import Rectangle
from lab_python_oop.Circle import Circle
from lab_python_oop.Square import Square
```

```
if __name__ == "__main__":
    rect = Rectangle(3, 2, 'blue')
    circle = Circle(5, 'green')
    square = Square(5, 'red')
    print(rect)
    print(circle)
    print(square)
```

GeomFigure.py

```
from abc import ABC, abstractmethod
```

```
class GeomFigure(ABC): # абстрактный класс
    __name = ""
```

```
    def return_name(self):
        return 'This figure is a {}'.format(self.__name)
```

```
    # абстрактный метод, который будет необходимо переопределять для каждого подкласса
    @abstractmethod
    def square(self):
        pass
```

Property.py

```
class Colour:
    def __init__(self):
        self._col = None

    @property
    def col(self):
        return self._col

    @col.setter
    def col(self, value):
        self._col = value

    @col.deleter
    def col(self):
        del self.col
```

Rectangle.py

```
from lab_python_oop import GeomFigure
from lab_python_oop import Property
```

```

class Rectangle(GeomFigure):
    def __init__(self, width, height, color):
        self.__width = width
        self.__height = height
        self.__c = Property.Colour()
        self.__c.col = color
        self.__name = 'Rectangle'

    def square(self):
        return self.__height * self.__height

    def __repr__(self):
        return 'Parameters of figure: \n \
Width = {} \n \
Height = {} \n \
Colour = {} \n \
Square = {}'.format(self.__width, self.__height, self.__c, self.square())

```

Circle.py

```

from lab_python_oop import GeomFigure
from lab_python_oop import Property
import math

```

```

class Circle(GeomFigure):
    def __init__(self, radius, color):
        self.__radius = radius
        self.__c = Property.Colour()
        self.__c.col = color
        self.__name = 'Circle'

    def square(self):
        return math.pi * self.__radius ** 2

    def __repr__(self):
        return 'Parameters of figure: \n \
Radius = {} \n \
Colour = {} \n \
Square = {}'.format(self.__radius, self.__c, self.square())

```

Square.py

```

from lab_python_oop import Rectangle
from lab_python_oop import Property

```

```

class Square(Rectangle):
    def __init__(self, side, color):
        self.__side = side
        self.__c = Property.Colour()
        self.__c.col = color
        self.__name = 'Square'

    def square(self):

```

```

return self.__side ** 2

def __repr__(self):
    return 'Parameters of figure: \n \
Side = {} \n \
Colour = {} \n \
Square = {}'.format(self.__side, self.__c, self.square())

```

Результат:

```

/Users/user/PycharmProjects/chat/env/bin/python /Users/user/PycharmProjects/Lab2_1/lab_python_oop/main.py
Rectangle -> Weight: 3, Height: 2, Colour: blue, Area: 6
Circle -> Radius: green, Color: 5, Area: 78.53981633974483
Square -> Weight: 5, Color: red, Area: 25
Process finished with exit code 0

```

Диаграмма классов:

