

Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Автоматизированные системы обработки
информации и управления»
Дисциплина «Разработка интернет-приложений»

Отчёт по лабораторной работе №2

Выполнил:
Студент группы ИУ5-53Б
Аникин Ф.А.

Проверил:
Преподаватель
Гапанюк Ю.Е.

Москва, 2020 г.

Постановка задачи

1. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием `pip`.
2. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
3. Все файлы проекта (кроме основного файла `main.py`) должны располагаться в пакете `lab_python_oop`.
4. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета `lab_python_oop`.
5. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь](#).
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).
7. 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.
8. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа `math.pi` из модуля [math](#).
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
 - Определите метод `"repr"`, который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод `format` - <https://pyformat.info/>
 - Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.

10. 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 (N - номер Вашего варианта по списку группы):

- Прямоугольник синего цвета шириной N и высотой N.
- Круг зеленого цвета радиусом N.
- Квадрат красного цвета со стороной N.
- Также вызовите один из методов внешнего пакета, установленного с использованием `pip`.

P.S. `main.py == Lab_2.py`

Текст программы

AF.py

```
import abc
```

```
class Figure(abc.ABC):
    @abc.abstractmethod
    def Area(self):
        pass
```

Rectangle.py

```
from lab_python_oop.AF import Figure
```

```
class Rectangle(Figure):
    def __init__(self, height, width, color):
        self.width = width
        self.height = height
        self.color = color

    def Area(self):
        return self.width*self.height

    def __str__(self):
        return f"{self.color} прямоугольник с высотой {self.height} и шириной {self.width}\nПлощадь = {self.Area()}"
```

Square.py

```
from lab_python_oop.Rectangle import Rectangle
```

```
class Square(Rectangle):
    def __init__(self, side, color):
        self.height = side
        self.width = side
        self.color = color

    def __str__(self):
        return f"{self.color} квадрат со стороной {self.height}\nПлощадь = {self.Area()}"
```

Cirlce.py

```
from lab_python_oop.AF import Figure
from math import pi

class Circle(Figure):
    def __init__(self, radius, color):
        self.radius = radius
        self.color = color

    def Area(self):
        return pi*self.radius*self.radius

    def __str__(self):
        return f"{self.color} круг с радиусом {self.radius}\nПлощадь = {self.Area()}"
```

Color.py

```
class Color():
    def __init__(self, color):
        self.color = color
    def __str__(self):
        return self.color
```

Lab 2.py

```
from lab_python_oop import Circle, Rectangle, Square, Color
from lab_python_oop.AF import Figure
from datetime import datetime

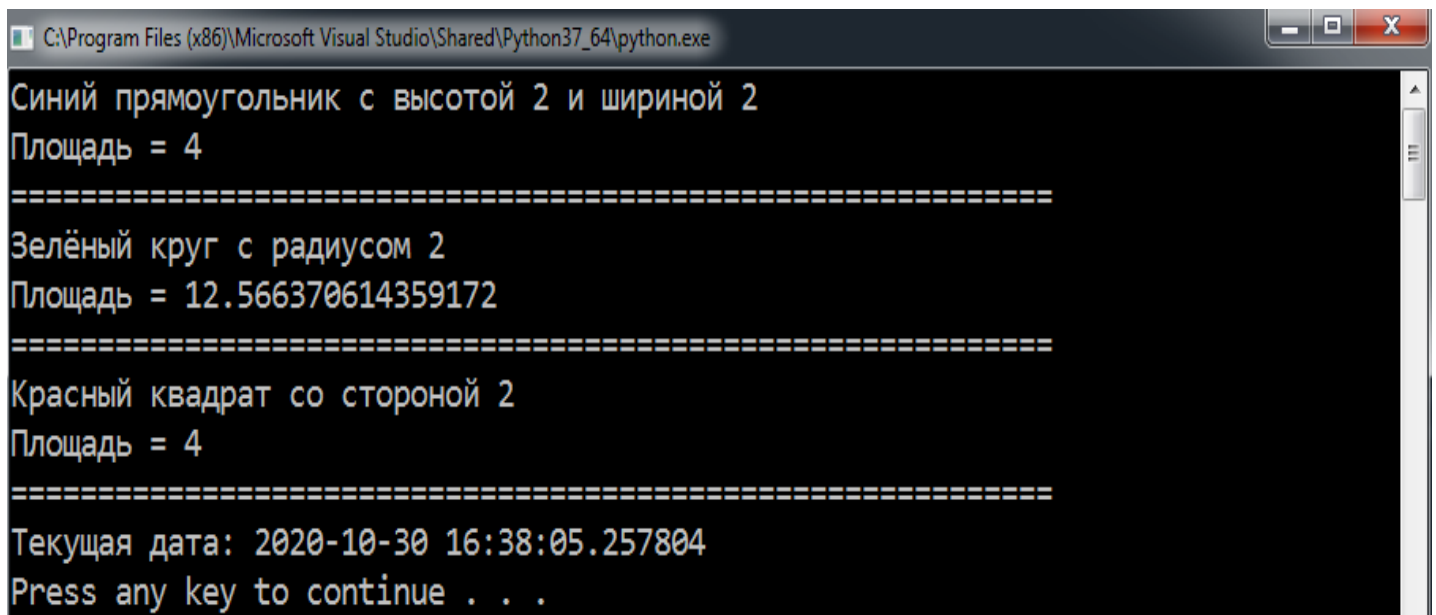
def main():
    BLUE = Color.Color("Синий")
    GREEN = Color.Color("Зелёный")
    RED = Color.Color("Красный")

    r = Rectangle.Rectangle(2,2,BLUE)
    print(f"{r}")
    print("="*60)
    c = Circle.Circle(2, GREEN)
    print(f"{c}")
    print("="*60)
    s = Square.Square(2,RED)
    print(f"{s}")
    print("="*60)

    print(f"Текущая дата: {datetime.now()}") #получение текущего времени

if __name__ == "__main__":
    main()
```

Пример работы программы

A screenshot of a Windows command prompt window running a Python script. The window title is 'C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe'. The output text is as follows:

```
Синий прямоугольник с высотой 2 и шириной 2  
Площадь = 4  
=====  
Зелёный круг с радиусом 2  
Площадь = 12.566370614359172  
=====  
Красный квадрат со стороной 2  
Площадь = 4  
=====  
Текущая дата: 2020-10-30 16:38:05.257804  
Press any key to continue . . .
```