|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №2

Выполнила:

студент группы ИУ5-51Б

Павловская А.А.

06.10.2020

Проверил:

|  |
| --- |
| преподаватель каф. ИУ5 |
| Гапанюк Ю.Е. |

Москва, 2020 г.

1. **Задание**
2. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.
3. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
4. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
5. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
6. 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)
7. 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)
8. 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.
9. 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)
10. 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.
11. 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.
12. **Текст программы**

**Figure.py**

from abc import ABC, abstractmethod

class Figure(ABC):

"""

Класс "Геометрическая фигура"

"""

#Абстрактный метод вычисления площади

@abstractmethod

def square(self):

pass

**FigureColor.py**

class FigureColor:

"""

Класс "Цвет фигуры"

"""

def \_\_init\_\_(self):

self.\_color = None

#Свойство для описания цвета фигуры

@property

def colorproperty(self):

return self.\_color

#Set color

@colorproperty.setter

def colorproperty(self, value):

self.\_color = value

**Rectangle.py**

from lab\_python\_oop.Figure import Figure

from lab\_python\_oop.FigureColor import FigureColor

class Rectangle(Figure):

"""

Класс "Прямоугольник"

"""

FIGURE\_TYPE = "Прямоугольник"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

#Конструктор Rectangle

def \_\_init\_\_(self, color\_param, width\_param, height\_param):

self.fc = FigureColor()

self.fc.colorproperty = color\_param

self.width = width\_param

self.height = height\_param

#Переопределение метода вычисления площади

def square(self):

return self.width\*self.height

def \_\_repr\_\_(self):

return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(

Rectangle.get\_figure\_type(),

self.fc.colorproperty,

self.width,

self.height,

self.square()

)

**Square.py**

from lab\_python\_oop.Rectangle import Rectangle

class Square(Rectangle):

"""

Класс "Квадрат"

"""

FIGURE\_TYPE = "Квадрат"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

#Конструктор Square

def \_\_init\_\_(self, color\_param, side\_param):

self.side = side\_param

super().\_\_init\_\_(color\_param, self.side, self.side)

def \_\_repr\_\_(self):

return '{} {} цвета со стороной {} площадью {}.'.format(

Square.get\_figure\_type(),

self.fc.colorproperty,

self.side,

self.square()

)

**Circle.py**

from lab\_python\_oop.Figure import Figure

from lab\_python\_oop.FigureColor import FigureColor

import math

class Circle(Figure):

"""

Класс "Круг"

"""

FIGURE\_TYPE = "Круг"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, color\_param, r\_param):

self.fc = FigureColor()

self.fc.colorproperty = color\_param

self.r = r\_param

def square(self):

return math.pi\*(self.r\*\*2)

def \_\_repr\_\_(self):

return '{} {} цвета радиусом {} площадью {}.'.format(

Circle.get\_figure\_type(),

self.fc.colorproperty,

self.r,

self.square()

)

**Main.py**

import colorama

from lab\_python\_oop.Rectangle import Rectangle

from lab\_python\_oop.Circle import Circle

from lab\_python\_oop.Square import Square

from colorama import Fore, Style

colorama.init()

def main():

print(Fore.GREEN + 'Павловская А.А. ИУ5-51Б')

print(Style.RESET\_ALL)

figure1 = Rectangle("синего", 15, 15)

figure2 = Circle("зеленого", 15)

figure3 = Square("красного", 15)

print(figure1)

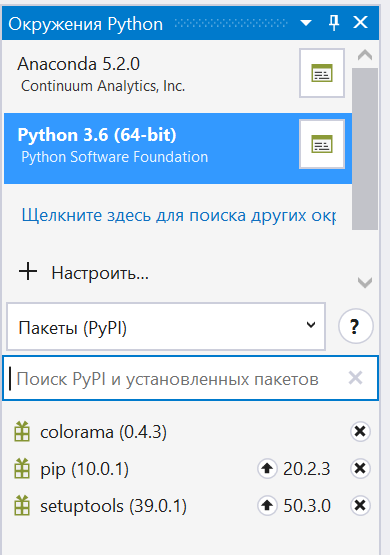
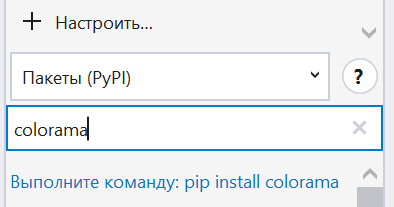
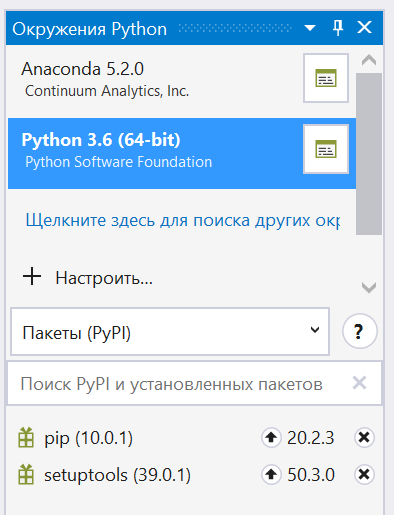
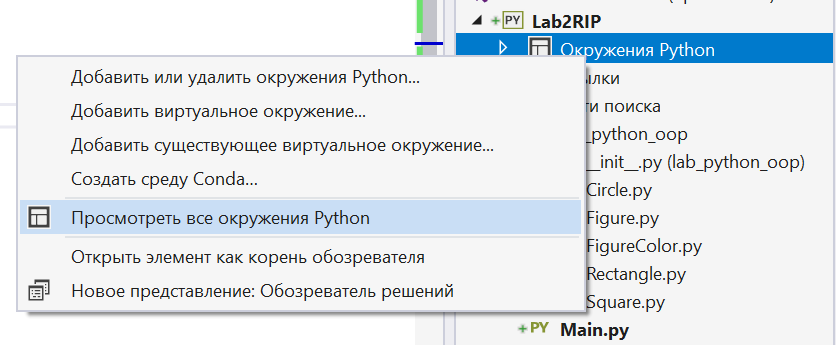
print(figure2)

print(figure3)

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. **Установка в виртуальное окружение внешнего пакета colorama с использованием pip (в Visual Studio 2017)**

****

1. **Экранные формы с примерами выполнения программы**

