**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 вычислительная техника»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-52Б |  | преподаватель каф. ИУ5 |
| Воронцова А.В. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

# ****Цель лабораторной работы****

Изучить объектно-ориентированные возможности языка Python.

# Описание задания

# Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.

# Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.

# Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.

# Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.

# Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры.

# Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.

# 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.

# Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math.

# 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.

# Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.

# 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.

# Текст программы

**main**

from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
  
  
def main():  
 r = Rectangle(18, 18, "синий")  
 print(r.get\_name())  
 print("---------------", r.\_\_repr\_\_())  
  
 c = Circle(18, "зеленый")  
 print(c.get\_name())  
 print("---------------", c.\_\_repr\_\_())  
  
 s = Square(18, "красный")  
 print(s.get\_name())  
 print("---------------", s.\_\_repr\_\_())  
 return  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**figueClass**

from abc import ABC, abstractmethod  
from lab\_python\_oop.figurColor import ColorShape  
  
  
  
class Figure(ABC):  
 def \_\_init\_\_(self, h, color, name):  
 self.\_color = ColorShape(color).color  
 self.\_h = h  
 self.\_name = name  
 self.\_s = self.schet()  
  
 @abstractmethod  
 def schet(self):  
 pass  
  
 def \_\_repr\_\_(self) -> str:  
 return '{}, его параметры: {}, площадь {}, цвет {}'.format(self.\_name, self.\_strSize, self.\_s, self.\_color)  
  
 def get\_name(self):  
 return self.\_name  
  
 def get\_s(self):  
 return self.\_s

**Circle**

from lab\_python\_oop.figueClass import Figure  
import math  
  
  
class Circle(Figure):  
  
 def \_\_init\_\_(self, h: int, color: str):  
 super(Circle, self).\_\_init\_\_(h, color, "Круг")  
 self.\_strSize = "радиус {}".format(h)  
  
 def schet(self):  
 return self.\_h \*\* 2 \* math.pi

**ColorFigure**

class ColorShape:  
 *"""Цвет фигуры"""* def \_\_init\_\_(self, \_color="синиЙ"):  
 self.set\_color(\_color)  
  
 def get\_color(self):  
 return self.\_\_color  
  
 def set\_color(self, \_color):  
 self.\_\_color = \_color  
  
 color = property(get\_color, set\_color)

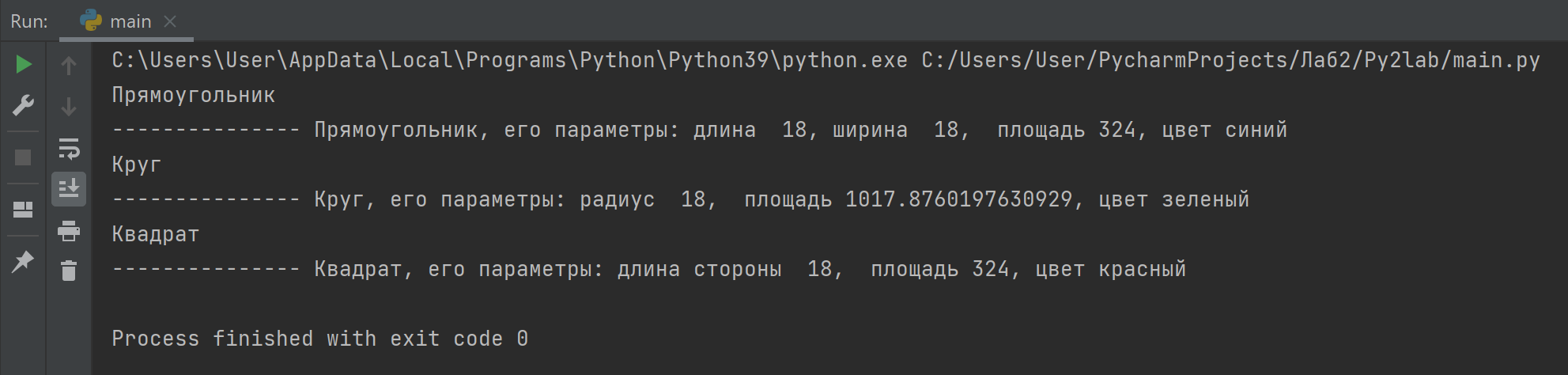
**Rectangle**

from lab\_python\_oop.figueClass import Figure  
  
  
class Rectangle(Figure):  
  
 def \_\_init\_\_(self, h, w, color="Синий"):  
 self.\_\_w = w  
 super(Rectangle, self).\_\_init\_\_(h, color, "Прямоугольник")  
 self.\_strSize = "длина {}, ширина {}".format(h, w)  
  
  
 def schet(self):  
 return self.\_\_w \* self.\_h

**Square**

from lab\_python\_oop.rectangle import Rectangle  
import math  
  
  
class Square(Rectangle):  
  
 \_\_name:str = "Прямоугольник"  
  
 def \_\_init\_\_(self, h, color):  
 super(Rectangle, self).\_\_init\_\_(h, color, "Квадрат")  
 self.\_strSize = "длина стороны {}".format(h)  
  
 def schet(self):  
 return self.\_h \* self.\_h

# Экранные формы с примерами выполнения программы



# Вывод

В ходе выполнения данной лабораторной работы были повторены основные конструкции языка Python. Также были изучены объектно-ориентированные возможности языка Python я.