**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)**

Курс «Разработка интернет-приложений»

# Отчет по лабораторной работе №2

Выполнила:

студентка группы ИУ5-53Б

Рыбина А.Д.

Преподаватель:

Гапанюк Ю.Е

# 2021

**Описание задания:**

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.

1. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
2. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
3. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры.
4. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
5. 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.

1. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math.
2. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:

o Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь.

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

1. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов. 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 крсного 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  
**import** arrow  
  
  
**def** main():  
 r = Rectangle(**"синего"**, 17, 17)  
 c = Circle(**"зеленого"**, 17)  
 s = Square(**"красного"**, 17)  
  
 utc\_time = arrow.utcnow()  
 print(**'Время (UTC): '**,utc\_time)  
 print(r)  
 print(c)  
 print(s)  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 main()

**Color**

**class** FigureColor:  
 *"""  
 Класс «Цвет фигуры»  
 """* **def** \_\_init\_\_(self):  
 self.\_color = **None** @property  
 **def** colorproperty(self):  
 *"""  
 Get-аксессор  
 """* **return** self.\_color  
  
 @colorproperty.setter  
 **def** colorproperty(self, value):  
 *"""  
 Set-аксессор  
 """* self.\_color = value

**Figure**

**from** abc **import** ABC, abstractmethod  
  
  
**class** Figure(ABC):  
 *"""  
 Абстрактный класс «Геометрическая фигура»  
 """* @abstractmethod  
 **def** square(self):  
 *"""  
 содержит виртуальный метод для вычисления площади фигуры.  
 """*

**Circle**

**from** lab\_python\_oop.figure **import** Figure  
**from** lab\_python\_oop.color **import** FigureColor  
**import** math  
  
  
**class** Circle(Figure):  
 *"""  
 Класс «Круг» наследуется от класса «Геометрическая фигура».  
 """* FIGURE\_TYPE = **"Круг"** @classmethod  
 **def** get\_figure\_type(cls):  
 **return** cls.FIGURE\_TYPE  
  
 **def** \_\_init\_\_(self, color\_param, r\_param):  
 *"""  
 Класс должен содержать конструктор по параметрам «радиус» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.  
 """* self.r = r\_param  
 self.fc = FigureColor()  
 self.fc.colorproperty = color\_param  
  
 **def** square(self):  
 *"""  
 Класс должен переопределять метод, вычисляющий площадь фигуры.  
 """* **return** math.pi\*(self.r\*\*2)  
  
 **def** \_\_repr\_\_(self):  
 **return '\033[32m{}\033[0m {} цвета радиусом {} площадью {}.'**.format(  
 Circle.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.r,  
 self.square()  
 )

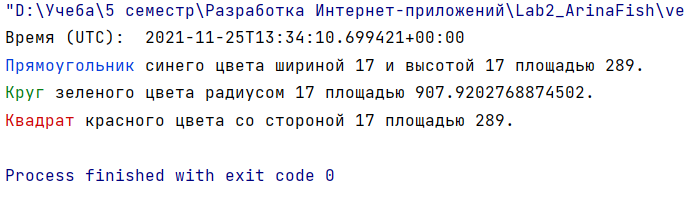
**Rectangle**

**from** lab\_python\_oop.figure **import** Figure  
**from** lab\_python\_oop.color **import** FigureColor  
  
  
**class** Rectangle(Figure):  
 *"""  
 Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».  
 """* FIGURE\_TYPE = **"Прямоугольник"** @classmethod  
 **def** get\_figure\_type(cls):  
 **return** cls.FIGURE\_TYPE  
  
 **def** \_\_init\_\_(self, color\_param, width\_param, height\_param):  
 *"""  
 Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.  
 """* self.width = width\_param  
 self.height = height\_param  
 self.fc = FigureColor()  
 self.fc.colorproperty = color\_param  
  
 **def** square(self):  
 *"""  
 Класс должен переопределять метод, вычисляющий площадь фигуры.  
 """* **return** self.width\*self.height  
  
 **def** \_\_repr\_\_(self):  
 **return '\033[34m{}\033[0m {} цвета шириной {} и высотой {} площадью {}.'**.format(  
 Rectangle.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.width,  
 self.height,  
 self.square()  
 )

**Square**

**from** lab\_python\_oop.rectangle **import** Rectangle  
  
  
**class** Square(Rectangle):  
 *"""  
 Класс «Квадрат» наследуется от класса «Прямоугольник».  
 """* FIGURE\_TYPE = **"Квадрат"** @classmethod  
 **def** get\_figure\_type(cls):  
 **return** cls.FIGURE\_TYPE  
  
 **def** \_\_init\_\_(self, color\_param, side\_param):  
 *"""  
 Класс должен содержать конструктор по параметрам «сторона» и «цвет».  
 """* self.side = side\_param  
 super().\_\_init\_\_(color\_param, self.side, self.side)  
  
 **def** \_\_repr\_\_(self):  
 **return '\033[31m{}\033[0m {} цвета со стороной {} площадью {}.'**.format(  
 Square.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.side,  
 self.square()  
 )

**Пример выполнения программы:**

****