МГТУ им. Н.Э. Баумана

Факультет «Информатика и системы управления»

ДИСЦИПЛИНА:

«Разработка интернет-приложений»

Отчёт по лабораторной работе №2

|  |
| --- |
| Выполнил:  Студент 3 курса  Факультет ИУ  Группа  ИУ5-52  Болотин А.С.  Преподаватель  Гапанюк Ю.Е. |

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

**Код программы**

AbstClass.py

from abc import ABC, abstractmethod  
# Абстрактный класс  
class Figure(ABC):  
 # Абстрактный метод  
 @abstractmethod  
 def square(self): pass  
  
class Color:  
 def \_\_init\_\_(self, color):  
 self.\_c = color  
  
 def getc(self):  
 return self.\_c  
  
 def setc(self, value):  
 self.\_c = value  
  
 def delc(self):  
 del self.\_c  
 # Объявление свойства  
 c = property(getc, setc, delc, "Color")

rect.py

from lab\_python\_oop.AbstClass import Figure, Color  
  
#Класс для расчета площади прямоугольника  
class Rectangle(Figure):  
 def \_\_init\_\_(self, length, size, color):  
 self.length = length  
 self.size = size  
 c = Color(color)  
 self.color = c.getc()  
 self.name = "Прямоугольник"  
#Вычисление площади  
 def square(self):  
 return self.length \* self.size  
  
 def \_\_repr\_\_(self):  
 return '{}, {}, {}, {}, {}'.format(self.name, self.length, self.size, self.square(), self.color)

Квадрат.py

from lab\_python\_oop.AbstClass import Figure, Color  
from lab\_python\_oop.rect import Rectangle  
  
class Square1(Rectangle):  
 def \_\_init\_\_(self, length, color):  
 self.length = length  
 c = Color(color)  
 self.color = c.getc()  
 self.name = "Квадрат"  
  
 def square(self):  
 return self.length \*\* 2  
  
 def \_\_repr\_\_(self):  
 print('{},{},{},{}'.format(self.name, self.length, self.square(), self.color))

Круг.py

from lab\_python\_oop.AbstClass import Figure, Color  
from math import pi  
  
class Circle(Figure):  
 def \_\_init\_\_(self, radius, color):  
 self.radius = radius  
 c = Color(color)  
 self.color = c.getc()  
 self.name = "Круг"  
  
 def square(self):  
 return pi \* (self.radius \*\* 2)  
  
 def \_\_repr\_\_(self):  
 print('{},{},{},{}'.format(self.name, self.radius, self.square(), self.color))

\_\_main\_\_.py

from lab\_python\_oop.rect import Rectangle  
from lab\_python\_oop.Квадрат import Square1  
from lab\_python\_oop.Круг import Circle  
  
if \_\_name\_\_ == '\_\_main\_\_':  
  
 rec = Rectangle(2, 3, 'синий')  
 print(rec)  
 #Rectangle.\_\_repr\_\_(rec)  
 #print(abv)  
 #Rectangle.square(rec)  
 #Rectangle.\_\_repr\_\_(rec)  
  
 sq = Square1(5, 'красный')  
 Square1.square(sq)  
 Square1.\_\_repr\_\_(sq)  
  
 crc = Circle(5, 'зеленый')  
 Circle.square(crc)  
 Circle.\_\_repr\_\_(crc)

**Результат работы программы**

