**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5.

Курс «Программирование на основе классов и шаблонов»

Отчет по лабораторной работе №1

« Основные конструкции языка Python »

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33б |  | преподаватель каф. ИУ5 |
| Соколов И.В. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

**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.](https://docs.python.org/3/library/abc.html)
6. 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)
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.](https://docs.python.org/3/library/math.html)
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.
11. **Дополнительное задание.** Протестируйте корректность работы Вашей программы с помощью модульного теста.

**Разработка интерфейса класса:**

**Классы квадрат, прямоугольник, круг, цвет**

**Текст программы:**

**Lab2.py**

from lab\_python\_oop.Figure import shape

from lab\_python\_oop.Circle import circle\_figure

from lab\_python\_oop.Rectangle import rectangle\_figure

from lab\_python\_oop.Square import square\_figure

from lab\_python\_oop.Figure import shape

def main():

rect=rectangle\_figure(input('введите цвет прямоугольника: '),int(input('введите сторону прямоугольника: ')),int(input('введите сторону прямоугольника: ')))

print(rect)

sq=square\_figure(input('введите цвет квадрата: '),int(input('введите сторону квадрата: ')))

print(sq)

circ=circle\_figure(input('введите цвет квадрата: '),int(input('введите сторону круга: ')))

print(circ)

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

main()

Square.py

from lab\_python\_oop.Rectangle import rectangle\_figure

class square\_figure(rectangle\_figure):

Figure\_type='Квадрат'

def \_\_init\_\_(self,cl,length):

self.length=length

super().\_\_init\_\_(cl,self.length,self.length)

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

return ("{} {} цвета со стороной {}, площадью {}"

.format(self.Figure\_type, self.fc.get\_color(), self.length,self.Square()))

**Rectangle.py**

from lab\_python\_oop.Figure import shape

from lab\_python\_oop.Color import shape\_color

class rectangle\_figure(shape\_color,shape):

Figure\_type='Прямоугольник'

def \_\_init\_\_(self,cl,length,width):

self.length=length

self.width=width

self.fc=shape\_color(cl)

def Square(self):

return self.length\*self.width

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

return ('{} {} цвета со сторонами {} и {}, и площадью {}'.format(self.Figure\_type,self.fc.get\_color(),self.length,self.width,self.Square()))

**Figure.py** from abc import abstractmethod, ABC

class shape(ABC):

'''Абстрактный класс геометрических фигур'''

@abstractmethod

def Square(self):

pass

Color.py

class shape\_color:

def \_\_init\_\_(self,color) -> None:

'''Конструктор Класса "Цвет"

На вход подается переменная color,которая принимает цвет

'''

self.\_\_color=color

def get\_color(self):

return self.\_\_color

**Circle.py**

from math import pi

from lab\_python\_oop.Color import shape\_color

from lab\_python\_oop.Figure import shape

class circle\_figure(shape\_color,shape):

Figure\_type='Круг'

def \_\_init\_\_(self,cl,radius):

self.\_\_radius=radius

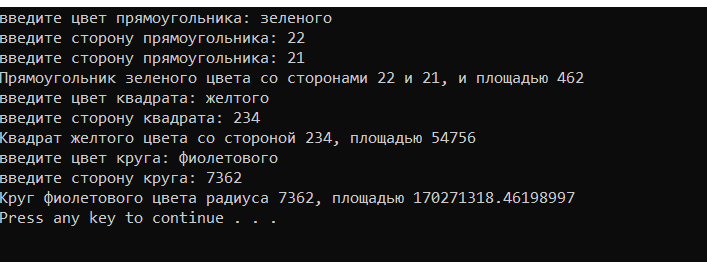
self.fc=shape\_color(cl)

def Square(self):

return (self.\_\_radius\*\*2)\*pi

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

return ("{} {} цвета радиуса {}, площадью {}".format(self.Figure\_type,self.fc.get\_color(),self.\_\_radius,self.Square()))

**Анализ результатов: **