**Московский государственный технический**

**университет 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

«Объектно-ориентированные возможности языка Python».

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| Аляев Д. В. ИУ5-51 |  | Гапанюк Ю. Е. |
|  |  | Балашов А. М. |

Москва, 2021 г.

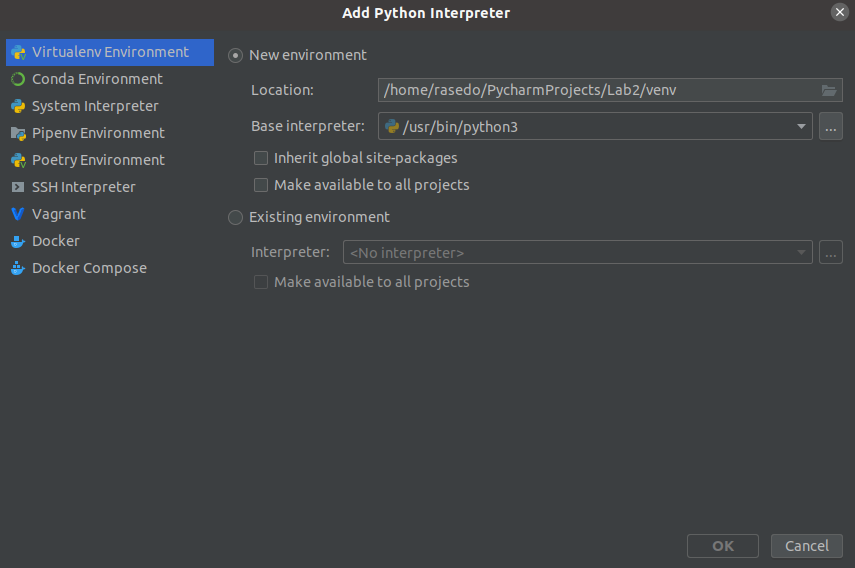
**Цель работы:**

Изучение объектно-ориентированных возможностей языка Python.

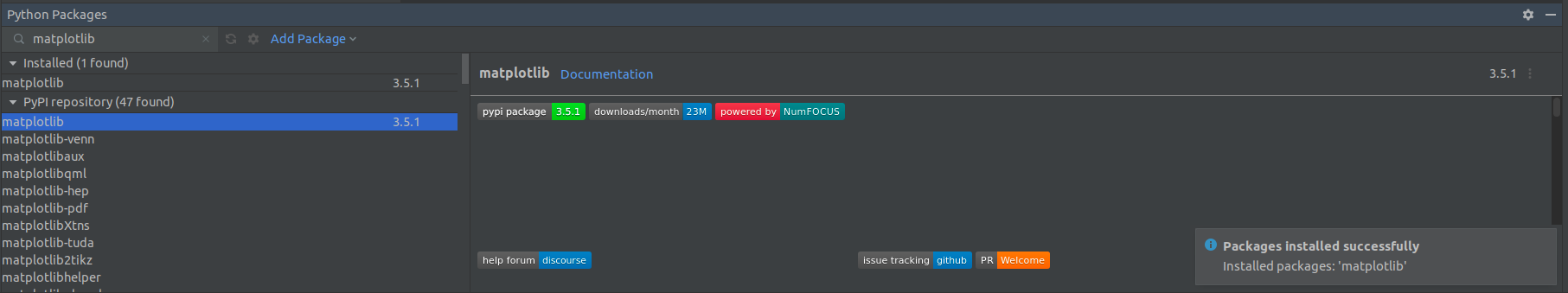
**Задание:**

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.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры.
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.
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.
   * 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 классов. Создайте следующие объекты и выведите о них информацию в консоль:
    * Прямоугольник синего цвета шириной 12 и высотой 12.
    * Круг зеленого цвета радиусом 12.
    * Квадрат красного цвета со стороной 12.
    * Также вызовите один из методов внешнего пакета, установленного с использованием pip.

**Создание виртуального окружения:**



**Установка пакетов:**



**Код:**

figure.py

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

color.py

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

circle.py

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 '{} {} цвета радиусом {} площадью {}.'.format(  
 Circle.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.r,  
 self.square()  
 )

rectangle.py

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 '{} {} цвета шириной {} и высотой {} площадью {}.'.format(  
 Rectangle.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.width,  
 self.height,  
 self.square()  
 )

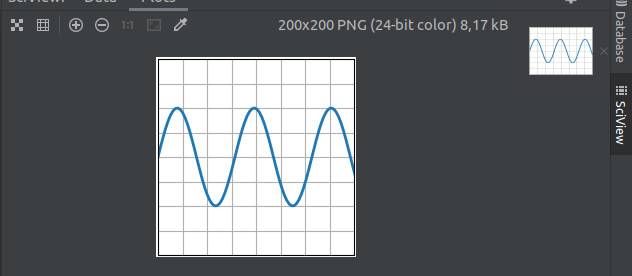
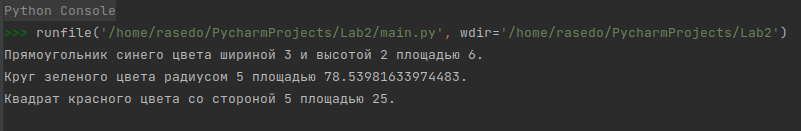
square.py

from lab\_python\_oop.rectangle import Rectangle  
  
  
class Square(Rectangle):  
 *"""*  
 *Класс «Квадрат» 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 '{} {} цвета со стороной {} площадью {}.'.format(  
 Square.get\_figure\_type(),  
 self.fc.colorproperty,  
 self.side,  
 self.square()  
 )

main.py

from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
import matplotlib.pyplot as plt  
import numpy as np  
  
  
def main():  
 r = Rectangle("синего", 3, 2)  
 c = Circle("зеленого", 5)  
 s = Square("красного", 5)  
 print(r)  
 print(c)  
 print(s)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()  
  
plt.style.use('\_mpl-gallery')  
  
# make data  
x = np.linspace(0, 10, 100)  
y = 4 + 2 \* np.sin(2 \* x)  
  
# plot  
fig, ax = plt.subplots()  
  
ax.plot(x, y, linewidth=2.0)  
  
ax.set(xlim=(0, 8), xticks=np.arange(1, 8),  
 ylim=(0, 8), yticks=np.arange(1, 8))  
  
plt.show()

**Пример запуска:**

****