**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: |
| студент группы ИУ5-53 |  | Юрий Евгеньевич Гапанюк |
| Миронова Александра |  |  |
| Подпись и дата:  18.01.22 |  | Подпись и дата: |

Москва, 2022 г.

**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. Подробнее про абстрактные классы и методы Вы можете прочитать здесь.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь.
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 - https://pyformat.info/
* 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 (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.

**Ход выполнения задания:**

1) создано виртуальное окружение



2) создан класс **GeometricFigure**(ABC), с виртуальным методом для вычисления площади фигуры def square(self)

3) создан класс class **FigureColor** с Get-аксессором def colorproperty(self) для получения цвета фигуры и с Set-аксессором def colorproperty(self, value) для установки значения цвета фигуры.

4) создан класс **Rectangle**(GeometricFigure), который наследуется от класса **GeometricFigure**. Класс содержит конструктор по параметрам «ширина», «высота» и «цвет» и переопределяет метод, вычисляющий площадь фигуры.

5) создан класс **Circle(GeometricFigure)**, который наследуется от класса **GeometricFigure.** Класс содержит конструктор по параметрам «радиус» и «цвет» и переопределяет метод, вычисляющий площадь фигуры.

6) создан класс **Square(Rectangle)** который наследуется от класса **Rectangle.** Класс содержит конструктор по параметрам «сторона» и «цвет».

7) создайте файл main.py для тестирования классов.

**Текст программы:**

**geometric\_figure.py**

from abc import ABC, abstractmethod

class GeometricFigure(ABC):

@abstractmethod

def square(self): # виртуальный метод для вычисления площади фигуры

pass

**color.py**

class FigureColor:

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

**rectangle.py**

from lab\_python\_oop.geometric\_figure import GeometricFigure

from lab\_python\_oop.color import FigureColor

class Rectangle(GeometricFigure):

FIGURE\_TYPE = "Прямоугольник"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, color\_param, width\_param, height\_param):

self.width = width\_param

self.height = height\_param

self.fc = FigureColor() # В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.

self.fc.colorproperty = color\_param

def square(self):

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()

)

**circle.py**

from lab\_python\_oop.geometric\_figure import GeometricFigure

from lab\_python\_oop.color import FigureColor

import math

class Circle(GeometricFigure):

FIGURE\_TYPE = "Круг"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, color\_param, r\_param):

self.r = r\_param

self.fc = FigureColor()

self.fc.colorproperty = color\_param

def square(self):

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()

)

**square.py**

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

FIGURE\_TYPE = "Квадрат"

@classmethod

def get\_figure\_type(cls):

return cls.FIGURE\_TYPE

def \_\_init\_\_(self, color\_param, side\_param):

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

def main():

rectangle = Rectangle("синего", 10, 5)

circle = Circle("зеленого", 15)

square = Square("красного", 15)

print(rectangle)

print(circle)

print(square)

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

main()

**Экранные формы с примерами выполнения программы:**

Выполнение программы файла main.py

