Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

Выполнил:
студент группы ИУ5-34Б:

Шимко Даниил
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.
Подпись и дата:

Москва, 2023 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.

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.

Код:

squaref.py

```
from figure import Geom
from coloroffigure import ColorFig
from prettytable import PrettyTable
class Square(Geom):
    name = "квадрат"
    def __init__(self, height, color):
        self.__height = height
        a = ColorFig()
        a.color = color
        self.__color = a.color
    def get_area(self):
        return self.__height ** 2
    def repr(self):
        table = PrettyTable()
        table.field_names = ["фигура","цвет","площадь"]
        table.add_row([self.name, self.__color, self.get_area()])
        print(table)
```

rectnglef.py

```
from figure import Geom
from coloroffigure import ColorFig
from prettytable import PrettyTable
class Rectangle(Geom):
    name = "прямоугольник"
    def __init__(self, width, height, color):
        self.__width = width
        self.__height = height
        a = ColorFig()
        a.color = color
        self.__color = a.color
    def get_area(self):
        return self.__width * self.__height
    def repr(self):
        table = PrettyTable()
        table.field_names = ["фигура","цвет","площадь"]
        table.add_row([self.name, self.__color, self.get_area()])
        print(table)
```

main.py

```
if __name__ == '__main__':
    from rectanglef import Rectangle
    from circlef import Circle
    from squaref import Square
    r = Rectangle(3, 4, "red")
    c = Circle(12, "green")
```

```
    s = Square(2, "blue")
    r.repr()
    c.repr()
    s.repr()
```

figure.py

```
from abc import ABC, abstractmethod
class Geom(ABC):
    @abstractmethod
    def get_area(self):
        ...
    @abstractmethod
    def repr(self):
        ...
```

coloroffigure.py

```
class ColorFig:
    @property
    def color(self):
        return self.__color
    @color.setter
    def color(self, color):
        self.__color = color
```

circlef.py

```
from figure import Geom
from coloroffigure import ColorFig
from prettytable import PrettyTable
from math import pi
class Circle(Geom):
    name ="круг"
    def __init__(self, radius, color):
        self.__radius = radius
        a = ColorFig()
        a.color = color
        self.__color = a.color
    def get_area(self):
        return 2 * pi * (self.__radius ** 2)
    def repr(self):
        table = PrettyTable()
        table.field_names = ["фигура","цвет","площадь"]
        table.add_row([self.name, self.__color, self.get_area()])
        print(table)
```