Московский государственный технический университет им. Н.Э. Баумана.

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2

«**Объектно-ориентированные возможности языка Python.**»

Выполнил:
студент группы ИУ5-32Б
Казицин Алексей

Проверил:
преподаватель кафедры
Гапанюк Ю.Е.

Подпись и дата:

Подпись и дата:

Москва, 2022 г.

Цель лабораторной работы: изучение объектно-ориентированных возможностей языка Python.

Требования к отчету:

Отчет по лабораторной работе должен содержать:

1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

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.
11. **Дополнительное задание.** Протестируйте корректность работы Вашей программы с помощью модульного теста.

Текст программы:

Main.py

```
import numpy

from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square


def main():
    r = Rectangle("синего", 8, 8)
    c = Circle("зеленого", 8)
    s = Square("красного", 8)

    print(r)
    print(c)
    print(s)
    a = numpy.array([[1, 2, 3], [4, 5, 6]])
    print(a)

if __name__ == "__main__":
    main()
```

Circle.py

```
import math

from lab_python_oop.figure import Figure
from lab_python_oop.color import Color


class Circle(Figure):
    """
    Класс "Круг"
    """

    FIGURE_TYPE = "Круг"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, r_param):
        self.fc = Color()
        self.fc.color_property = color_param
        self.r = r_param

    def square(self):
        return 2 * math.pi * self.r ** 2

    def __repr__(self):
        return '{} {} цвета, радиуса {}, площадью {:.3f}.'.format(
            self.get_figure_type(),
            self.fc.color_property,
            self.r,
            self.square()
        )
```

Color.py

```
class Color:
    """
    Класс "Цвет фигуры"
    """

    def __init__(self):
        self._color = None

    @property
    def color_property(self):
        """
        Get-аксессор
        """
        return self._color

    @color_property.setter
    def color_property(self, value):
        """
        Set-аксессор
        """
        self._color = value
```

Figure.py

```
from abc import ABC, abstractmethod


class Figure:
    """
    Абстрактный класс "Геометрическая фигура"
    """

    @abstractmethod
    def square(self):
        """
        Виртуальный метод вычисления площади фигуры
        """
        pass
```

Rectangle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import Color


class Rectangle(Figure):
    """
    Класс "Прямоугольник"
    """
    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 = Color()
        self.fc.color_property = color_param

    def square(self):
        """
        Вычисление площади фигуры
        """
        return self.width * self.height

    def __repr__(self):
        return '{} {} цвета, шириной {} и высотой {}, площадью {:.3f}.'.format(
            Rectangle.get_figure_type(),
            self.fc.color_property,
            self.width,
            self.height,
            self.square()
        )
```

Square.py

```
from lab_python_oop.rectangle import Rectangle


class Square(Rectangle):
    """
    Класс "Квадрат"
    """
    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 "цвет" и "сторона"
        """
        self.side = side_param
        super().__init__(color_param, self.side, self.side)

    def __repr__(self):
        return '{} {} цвета со стороной {}, площадью {:.3f}.'.format(
            self.get_figure_type(),
            self.fc.color_property,
            self.side,
            self.square()
        )
```