

Московский государственный технический университет им. Н.Э. Баумана

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

Курс «Базовые компоненты интернет-технологий»
Отчёт по лабораторной работе №2

Выполнил:
студент группы ИУ5-31Б
Смыслов Дмитрий
Олегович

Подпись: _____

Дата: _____

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий
Евгеньевич

Подпись: _____

Дата: _____

Москва, 2021 г.

Описание задания

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.
10. 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.
11. В корневом каталоге проекта создайте файл `main.py` для тестирования Ваших классов. 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.

Текст программы

Файл figure.py

```
from abc import ABC, abstractmethod

class Figure(ABC):

    @abstractmethod
    def square(self):
        pass
```

Файл color.py

```
class FigureColor:

    def __init__(self):
        self._color = None

    @property
    def colorproperty(self):
        return self._color

    @colorproperty.setter
    def colorproperty(self, value):
        self._color = value
```

Файл rectangle.py

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

class Rectangle(Figure):

    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()
        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()
        )

```

Файл circle.py

```

from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
import math

```

```

class Circle(Figure):

    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()
        )

```

Файл 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()
    )

```

Файл main.py

```

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

def main():
    r = Rectangle("синего", 19, 19)
    c = Circle("зеленого", 19)
    s = Square("красного", 19)
    print(r)
    print(c)
    print(s)

    LabTable = PrettyTable()
    LabTable.field_names = ["Номер лабы", "Название лабы", "Выполнена",
"Отчёт"]
    LabTable.add_row([1, 'Основные конструкции языка Python', True, True])
    LabTable.add_row([2, 'Объектно-ориентированные возможности языка Python',
True, False])
    LabTable.add_row([3, 'Функциональные возможности языка Python', True,
False])
    print(LabTable)

if __name__ == "__main__":
    main()

```

Экранные формы с примерами выполнения программы

В качестве внешнего пакета был выбран “prettytable” главной задачей которого является создание таблиц с помощью ASCII символов и печати такой таблицы на консоль.

```
Terminal: Local x + v
(venv) PS G:\PycharmProjects\Laba2> py main.py
Прямоугольник синего цвета шириной 19 и высотой 19 площадью 361.
Круг зеленого цвета радиусом 19 площадью 1134.1149479459152.
Квадрат красного цвета со стороной 19 площадью 361.
+-----+-----+-----+-----+
| Номер лабы |          Название лабы          | Выполнена | Отчёт |
+-----+-----+-----+-----+
|    1    | Основные конструкции языка Python |    True  |   True |
|    2    | Объектно-ориентированные возможности языка Python |    True  |   False |
|    3    | Функциональные возможности языка Python |    True  |   False |
+-----+-----+-----+-----+
(venv) PS G:\PycharmProjects\Laba2> 
```