

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по лабораторной работе № 2
“Объектно-ориентированные возможности языка Python”

Выполнил:
Студент группы ИУБ-36Б
Левочкин В.В.


Преподаватель:
Нардид. А.Н.

Москва 2025

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.

main.py

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
import requests
import sys
import math

N=18
print(f"Используем N = {N}( номер варианта.)")

def main():
    # 1. Прямоугольник синего цвета шириной N и высотой N.
    rect=Rectangle(N, N, "синий")
    print(repr(rect))

    # 2. Круг зеленого цвета радиусом N.
    circ=Circle(N, "зеленый")
    print(repr(circ))

    # 3. Квадрат красного цвета со стороной N.
    sq=Square(N, "красный")
    print(repr(sq))

    print("\n--- Вызов метода внешнего пакета (requests) ---")

    try:

        response=requests.get('https://www.google.com/', timeout=5)

        print(f"Запрос к Google.com успешен. Статус-код: {response.status_code}")
        print(f"Тип контента: {response.headers.get('Content-Type', 'Не указан')}")


    except requests.exceptions.RequestException as e:
        print(f"Ошибка при выполнении запроса: {e}")

    print("-----")
```

```
print(f"Версия Python: {sys.version.split()[0]}")
print(f"Значение math.pi: {math.pi}")

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

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 Figure.
    """

    FIGURE_TYPE="Круг"

    def __init__(self, radius_param, color_param):
        """
        Конструктор круга.
        :param radius_param: Радиус.
        :param color_param: Цвет (строка).
        """
        self.radius=radius_param
        self.fc=FigureColor(color_param)

    def area(self):
        """
        Переопределенный метод для вычисления площади круга.
        """
        return math.pi*(self.radius**2)

    def __repr__(self):
        """
        Метод __repr__, возвращающий основные параметры фигуры, ее цвет и
        площадь.
        """
        return f"{{}, радиус {:.2f}, цвет {}, площадь {:.2f}}".format(
            self.get_figure_type(),
            self.radius,
            self.fc.color,
            self.area()
        )

    @classmethod
    def get_figure_type(cls):
        """
        Метод класса, возвращающий название фигуры.
        """
```

```
returncls.FIGURE_TYPE
```

Color.py

```
classFigureColor:  
    """  
    Класс, содержащий свойство для описания цвета геометрической фигуры.  
    """  
    def __init__(self, color_param):  
        self._color=color_param  
  
    @property  
    defcolor(self):  
        """Свойство для получения цвета."""  
        returnself._color  
  
    @color.setter  
    defcolor(self, value):  
        """Свойство для установки цвета."""  
        self._color=value
```

figure.py

```
importabc  
  
classFigure(abc.ABC):  
    """  
    Абстрактный класс "Геометрическая фигура".  
    Содержит абстрактный метод для вычисления площади.  
    """  
  
    @abc.abstractmethod  
    defarea(self):  
        """  
        Абстрактный метод для вычисления площади фигуры.  
        """  
        pass
```

rectangle.py

```
fromlab_python_oop.figureimportFigure  
fromlab_python_oop.colorimportFigureColor  
  
classRectangle(Figure):  
    """  
    Класс "Прямоугольник", наследуется от Figure.  
    """  
  
    FIGURE_TYPE="Прямоугольник"  
  
    def __init__(self, width_param, height_param, color_param):  
        """  
        Конструктор прямоугольника.  
        """
```

```

:param width_param: Ширина.
:param height_param: Высота.
:param color_param: Цвет (строка).
"""
self.width=width_param
self.height=height_param
self.fc=FigureColor(color_param)

def area(self):
    """
    Переопределенный метод для вычисления площади прямоугольника.
    """
    return self.width*self.height

def __repr__(self):
    """
    Метод repr, возвращающий основные параметры фигуры, ее цвет и
    площадь.
    """
    return '{} , ширина {}, высота {}, цвет {}, площадь {}'.format(
        self.get_figure_type(),
        self.width,
        self.height,
        self.fc.color,
        self.area()
    )

@classmethod
def get_figure_type(cls):
    """
    Метод класса, возвращающий название фигуры.
    """
    return cls.FIGURE_TYPE

```

Square.py

```

from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    """
    Класс "Квадрат", наследуется от Rectangle.
    """

    FIGURE_TYPE="Квадрат"

    def __init__(self, side_param, color_param):
        """
        Конструктор квадрата.
        :param side_param: Длина стороны.
        :param color_param: Цвет (строка).
        """
        super().__init__(side_param, side_param, color_param)
        self.side=side_param

```

```
def __repr__(self):
    """
    Переопределенный метод repr для квадрата.
    """
    return "{} {}, сторона {}, цвет {}, площадь {}".format(
        self.get_figure_type(),
        self.side,
        self.fc.color,
        self.area()
    )
```

Вывод программы

```
● (venv) @vasyteri → /workspaces/Python/LAb_2 (main) $ python3 main.py
Используем N = 18 ( номер варианта).
"Прямоугольник, ширина 18, высота 18, цвет синий, площадь 324"
"Круг, радиус 18.00, цвет зеленый, площадь 1017.88"
"Квадрат, сторона 18, цвет красный, площадь 324"

--- Вызов метода внешнего пакета (requests) ---
Запрос к Google.com успешен. Статус-код: 200
Тип контента: text/html; charset=ISO-8859-1
```