

**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 ИУ5-33Б
Левкович Леонид

Проверил:
Преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2025 г.

Задание:

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.

Листинг программы:

main.py

```
"""
Главный модуль для тестирования классов геометрических фигур
"""

from lab_python_oop import Rectangle, Circle, Square
from colorama import init, Fore, Style

# Инициализация colorama для цветного вывода
init(autoreset=True)

def main():
    """
    Основная функция программы
    """
```

```

# Номер варианта (замените на свой номер по списку)
N = 5

print(Fore.CYAN + "=" * 60)
print(Fore.CYAN + "Лабораторная работа №2: ООП в Python")
print(Fore.CYAN + "=" * 60 + "\n")

# Создание прямоугольника
rectangle = Rectangle(N, N, "синий")
print(Fore.BLUE + str(rectangle))

# Создание круга
circle = Circle(N, "зеленый")
print(Fore.GREEN + str(circle))

# Создание квадрата
square = Square(N, "красный")
print(Fore.RED + str(square))

print("\n" + Fore.CYAN + "=" * 60)
print(Fore.YELLOW + "Демонстрация работы внешнего пакета colorama:")
print(Fore.MAGENTA + "Этот текст выведен разными цветами!" +
Style.RESET_ALL)
print(Fore.CYAN + "=" * 60)

if __name__ == "__main__":
    main()

```

init.py

```

"""
Пакет для работы с геометрическими фигурами
"""

from .geometric_figure import GeometricFigure
from .color import Color
from .rectangle import Rectangle
from .circle import Circle
from .square import Square

__all__ = ['GeometricFigure', 'Color', 'Rectangle', 'Circle', 'Square']

```

circle.py

```

"""
Модуль содержит класс Circle (Круг)
"""

import math
from .geometric_figure import GeometricFigure
from .color import Color

class Circle(GeometricFigure):
    """
    Класс Круг, наследуется от GeometricFigure
    """

    FIGURE_NAME = "Круг"

    def __init__(self, radius, color):
        """
        Конструктор круга

        :param radius: радиус
        :param color: цвет (строка)
        """

```

```

        self.radius = radius
        self.figure_color = Color()
        self.figure_color.color = color

    def area(self):
        """
        Вычисление площади круга
        """
        return math.pi * self.radius ** 2

    @classmethod
    def get_figure_name(cls):
        """
        Возвращает название фигуры
        """
        return cls.FIGURE_NAME

    def __repr__(self):
        """
        Строковое представление круга
        """
        return "{} {} цвета с радиусом {}, площадь: {:.2f}".format(
            self.get_figure_name(),
            self.figure_color.color,
            self.radius,
            self.area()
        )

```

color.py

```

"""
Модуль содержит класс Color для работы с цветом фигуры
"""

class Color:
    """
    Класс для описания цвета геометрической фигуры
    """

    def __init__(self):
        self._color = None

    @property
    def color(self):
        """
        Свойство для получения цвета
        """
        return self._color

    @color.setter
    def color(self, value):
        """
        Свойство для установки цвета
        """
        self._color = value

```

geometric_figure.py

```

"""
Модуль содержит абстрактный класс GeometricFigure
"""

from abc import ABC, abstractmethod

class GeometricFigure(ABC):
    """
    Абстрактный класс для геометрических фигур
    """

```

```

@abstractmethod
def area(self):
    """
    Абстрактный метод для вычисления площади фигуры
    """
    pass

```

rectangle.py

```

"""
Модуль содержит класс Rectangle (Прямоугольник)
"""

from .geometric_figure import GeometricFigure
from .color import Color

class Rectangle(GeometricFigure):
    """
    Класс Прямоугольник, наследуется от GeometricFigure
    """

    FIGURE_NAME = "Прямоугольник"

    def __init__(self, width, height, color):
        """
        Конструктор прямоугольника

        :param width: ширина
        :param height: высота
        :param color: цвет (строка)
        """
        self.width = width
        self.height = height
        self.figure_color = Color()
        self.figure_color.color = color

    def area(self):
        """
        Вычисление площади прямоугольника
        """
        return self.width * self.height

    @classmethod
    def get_figure_name(cls):
        """
        Возвращает название фигуры
        """
        return cls.FIGURE_NAME

    def __repr__(self):
        """
        Строковое представление прямоугольника
        """
        return "{} {} цвета со сторонами {} и {}, площадь: {:.2f}".format(
            self.get_figure_name(),
            self.figure_color.color,
            self.width,
            self.height,
            self.area()
        )

```

square.py

```

"""
Модуль содержит класс Square (Квадрат)
"""

from .rectangle import Rectangle

```

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

    FIGURE_NAME = "Квадрат"

    def __init__(self, side, color):
        """
        Конструктор квадрата

        :param side: длина стороны
        :param color: цвет (строка)
        """
        super().__init__(side, side, color)
        self.side = side

    def __repr__(self):
        """
        Строковое представление квадрата
        """
        return "{} {} цвета со стороной {}, площадь: {:.2f}".format(
            self.get_figure_name(),
            self.figure_color.color,
            self.side,
            self.area()
        )
```

Результат выполнения:

```
=====
Лабораторная работа №2: ООП в Python
=====

Прямоугольник синий цвета со сторонами 5 и 5, площадь: 25.00
Круг зеленый цвета с радиусом 5, площадь: 78.54
Квадрат красный цвета со стороной 5, площадь: 25.00

=====
Демонстрация работы внешнего пакета colorama:
Этот текст выведен разными цветами!
=====

Process finished with exit code 0
```

