

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
“Основные конструкции языка Python ”

Выполнила:
Студент группы ИУ5-34Б
Глозман В.А.
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 "getr", 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.

main.cpp

```
import sys
```

```
import math
```

```
def read_coefficient(param_index, prompt_message):
```

```
    coefficient = None
```

```
    if len(sys.argv) > param_index:
```

```
        try:
```

```
            coefficient = float(sys.argv[param_index])
```

```
            return coefficient
```

```
        except ValueError:
```

```
            print("Некорректное значение в аргументах. Требуется ввод с клавиатуры.")
```

```
    while True:
```

```
        print(prompt_message)
```

```
        try:
```

```
            coefficient = float(input())
```

```
            return coefficient
```

```
        except ValueError:
```

```
print("Ошибка: введите действительное число.")
```

```
def solve_biquadratic(a, b, c):
```

```
    roots = []
```

```
    discriminant = b ** 2 - 4 * a * c
```

```
    if discriminant >= 0:
```

```
        y1 = (-b + math.sqrt(discriminant)) / (2 * a)
```

```
        y2 = (-b - math.sqrt(discriminant)) / (2 * a)
```

```
    for y in [y1, y2]:
```

```
        if y > 0:
```

```
            x1 = math.sqrt(y)
```

```
            x2 = -math.sqrt(y)
```

```
            roots.extend([x1, x2])
```

```
        elif y == 0:
```

```
            roots.append(0.0)
```

```
    return sorted(roots)
```

```
def main():
```

```
    print("Решение биквадратного уравнения  $A \cdot x^4 + B \cdot x^2 + C = 0$ ")
```

```
    a = read_coefficient(1, "Введите коэффициент A:")
```

```
b = read_coefficient(2, "Введите коэффициент В:")
c = read_coefficient(3, "Введите коэффициент С:")

if a == 0:
    print("Ошибка: коэффициент А не может быть нулем для биквадратного
уравнения.")
    return

roots = solve_biquadratic(a, b, c)

if not roots:
    print("Действительных корней нет")
else:
    print(f"Найдено корней: {len(roots)}")
    for i, root in enumerate(roots, 1):
        print(f"Корень {i}: {root:.4f}")

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