Объектно-ориентированные возможности языка Python.

Выполнил: Абдуллаев Гасан
Группа: ИУ5-36Б
Дата: 15.11.24г.

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. 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. 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.

Программный код:

```
import sys
import math

def get_coefficient(prompt):
    while True:
        try:
            return float(input(prompt))
        except ValueError:
            print("Некорректный ввод. Пожалуйста, введите число.")

def get_coefficient_from_args_or_input(arg_index, prompt):
    try:
        return float(sys.argv[arg_index])
    except (IndexError, ValueError):
        return get_coefficient(prompt)

def solve_quadratic_for_t(a, b, c):
    discriminant = b**2 - 4 * a * c
    print(f"Дискриминант: {discriminant}")

    if discriminant > 0:
        t1 = (-b + math.sqrt(discriminant)) / (2 * a)
        t2 = (-b - math.sqrt(discriminant)) / (2 * a)
        return t1, t2
    elif discriminant == 0:
        t = -b / (2 * a)
        return t, None
```

```
    else:
        return None, None

def solve_biquadratic_equation(a, b, c):
    if a == 0:
        print("Это не биквадратное уравнение, так как A = 0.")
        return

    t1, t2 = solve_quadratic_for_t(a, b, c)

    roots = []
    if t1 is not None and t1 >= 0:
        x1 = math.sqrt(t1)
        x2 = -math.sqrt(t1)
        roots.extend([x1, x2])

    if t2 is not None and t2 >= 0:
        x3 = math.sqrt(t2)
        x4 = -math.sqrt(t2)
        roots.extend([x3, x4])

    if roots:
        print("Действительные корни уравнения: ", sorted(set(roots)))
    else:
        print("Действительных корней нет.")

def main():
    a = get_coefficient_from_args_or_input(1, "Введите коэффициент A: ")
    b = get_coefficient_from_args_or_input(2, "Введите коэффициент B: ")
```

```
    c = get_coefficient_from_args_or_input(3, "Введите коэффициент C: ")

    solve_biquadratic_equation(a, b, c)

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

ООП

```
import sys
import math

class BiquadraticEquation:
    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c

    def solve_quadratic_for_t(self):
        discriminant = self.b**2 - 4 * self.a * self.c
        print(f"Дискриминант: {discriminant}")

        if discriminant > 0:
            t1 = (-self.b + math.sqrt(discriminant)) / (2 * self.a)
            t2 = (-self.b - math.sqrt(discriminant)) / (2 * self.a)
            return t1, t2
        elif discriminant == 0:
            t = -self.b / (2 * self.a)
            return t, None
        else:
```

```
            return None, None

    def solve_biquadratic(self):
        if self.a == 0:
            print("Это не биквадратное уравнение, так как A = 0.")
            return

        t1, t2 = self.solve_quadratic_for_t()

        roots = []
        if t1 is not None and t1 >= 0:
            roots.append(math.sqrt(t1))
            roots.append(-math.sqrt(t1))

        if t2 is not None and t2 >= 0:
            roots.append(math.sqrt(t2))
            roots.append(-math.sqrt(t2))

        if roots:
            print("Действительные корни уравнения: ", sorted(set(roots)))
        else:
            print("Действительных корней нет.")

class InputHandler:
    @staticmethod
    def get_coefficient(prompt):
        while True:
            try:
                return float(input(prompt))
```

```
            except ValueError:
                print("Некорректный ввод. Пожалуйста, введите число.")
    @staticmethod
    def get_coefficient_from_args_or_input(arg_index, prompt):
        try:
            return float(sys.argv[arg_index])
        except (IndexError, ValueError):
            return InputHandler.get_coefficient(prompt)


def main():
    a = InputHandler.get_coefficient_from_args_or_input(1, "Введите коэффициент A: ")
    b = InputHandler.get_coefficient_from_args_or_input(2, "Введите коэффициент B: ")
    c = InputHandler.get_coefficient_from_args_or_input(3, "Введите коэффициент C: ")


    equation = BiquadraticEquation(a, b, c)


    equation.solve_biquadratic()


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

Снимки экрана:

```
Введите коэффициент A: 1
Введите коэффициент B: -5
Введите коэффициент C: 4
Дискриминант: 9.0
Действительные корни уравнения:  [-2.0, -1.0, 1.0, 2.0]
```

ООП

```
Введите коэффициент A: 1
Введите коэффициент B: -5
Введите коэффициент C: 4
Дискриминант: 9.0
Действительные корни уравнения:  [-2.0, -1.0, 1.0, 2.0]
```