

PwJJ_01	Romaniak Hubert	Informatyka niestacjonarna III rok	Semestr zimowy 2024/25
---------	-----------------	---------------------------------------	---------------------------

Zadanie 1

Sprawdzenie zainstalowanej wersji Javy

```
C:\Users\huber>java --version
openjdk 23.0.1 2024-10-15
OpenJDK Runtime Environment Temurin-23.0.1+11 (build 23.0.1+11)
OpenJDK 64-Bit Server VM Temurin-23.0.1+11 (build 23.0.1+11, mixed mode, sharing)
```

Zadanie 2

Implementacja klasy Shape, kompilacja i uruchomienie z terminala

```
/**
 * Class representing any shape.
 */
public final class Shape { new *

    /**
     * Application entry point.
     */
    public static void main() { new *
    |     new Shape().print();
    |
    |
    |     private void print() { 1 usage new *
    |     |     System.out.println(getClass().getName());
    |     |
    |     }
    |
    }
}
```

```
>javac Shape.java && java --enable-preview Shape
```

```
Shape
```

Zadanie 3

Implementacja klas Rectangle, Triangle i ShapeDescriber

```
/**
 * Class representing any shape.
 */
public abstract class Shape { 3 usages 2 inheritors new *

    /**
     * Prints class name.
     */
    void print() { 1 usage new *
        System.out.println(getClass().getName());
    }

    /**
     * Calculates area.
     *
     * @return the area of a shape
     */
    public abstract double getArea(); 1 usage 2 implementations new *

    /**
     * Calculates perimeter.
     *
     * @return the perimeter of a shape
     */
    public abstract double getPerimeter(); 2 usages 2 implementations new *
}
```

```
/**
 * Class representing rectangle.
 */
final class Rectangle extends Shape { 1 usage new *

    private final double x; 3 usages
    private final double y; 3 usages

    /**
     * Creates {@code Shape} representing rectangle.
     *
     * @param x first side length
     * @param y second side length
     */
    Rectangle(final double x, final double y) { 1 usage new *
        this.x = x;
        this.y = y;
    }

    @Override 1 usage new *
    public double getArea() {
        return x * y;
    }

    @Override 2 usages new *
    public double getPerimeter() {
        return 2.0 * (x + y);
    }
}
```

```

/**
 * Class representing triangle.
 */
final class Triangle extends Shape { 1 usage new *

    private final double x; 3 usages
    private final double y; 3 usages
    private final double z; 3 usages

    /**
     * Creates {@code Shape} representing triangle.
     *
     * @param x first side length
     * @param y second side length
     * @param z third side length
     */
    Triangle(final double x, final double y, final double z) { 1 usage new *
        this.x = x;
        this.y = y;
        this.z = z;
    }

    @Override 1 usage new *
    public double getArea() {
        final double halfPerimeter = getPerimeter() / 2.0;
        return Math.sqrt(halfPerimeter * (halfPerimeter - x) * (halfPerimeter - y) * (halfPerimeter - z));
    }

    @Override 2 usages new *
    public double getPerimeter() {
        return x + y + z;
    }
}

```

```

/**
 * Class for statically describing objects of type {@code Shape}.
 */
final class ShapeDescriber { 2 usages new *

    private ShapeDescriber() { no usages new *
    }

    /**
     * Prints {@code shape} information.
     *
     * @param shape shape to describe
     */
    static void describe(final Shape shape) { 2 usages new *
        System.out.print("Shape name: ");
        shape.print();
        System.out.println("Area: " + shape.getArea());
        System.out.println("Perimeter: " + shape.getPerimeter() + '\n');
    }
}

```

```
/** new *  
 * Application entry point.  
 */  
public static void main() { new *  
    ShapeDescriber.describe(new Rectangle(x: 1.0, y: 2.0));  
    ShapeDescriber.describe(new Triangle(x: 3.0, y: 4.0, z: 5.0));  
}
```