



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні систем та технологій

Лабораторна робота №8
із дисципліни «Основи програмування»

Тема: Основи ООП

Виконали:
Студенти групи ІА-24
Призвіще:
Шкарніков Антон,
Кармазіна Анастасія,
Сотніков Олексій.

Перевірив:
Колеснік Валерій Миколайович

Хід роботи:

1. Повторити теоретичні відомості
2. Проаналізувати предметну область завдання свого варіанту (табл.1)
3. Розробити базовий клас (відповідно до завдання можливо абстрактний клас або інтерфейс), клас-нащадок, а також допоміжні класи та/або інтерфейси за необхідністю. Відповідно до предметної області завдання передбачити відповідні методи бізнес-логіки, а також конструктори, сетери та/або гетери, методи equals() та toString(). Продемонструвати використання:
 - this;
 - super;
 - перевантаження (overloading) та заміщення(overriding) методів;
 - перевантаження (overloading) конструкторів.
4. Відповісти на контрольні запитання

Таблиця 1. Варіанти завдань

№	Базовий клас	Класи-нащадки	Мутабельність об'єктів
4	Довільна фігура	Прямокутник, Коло	mutable

Клас Figure

```
public abstract class Figure {  
    private String name;  
  
    public abstract double getArea();  
    public abstract double getPerimeter();  
  
    public Figure(String name) {  
        this.name = name;  
    }  
  
    @Override  
    public String toString() {  
        return "Figure{" +  
            "name='" + name + '\'' +  
            '}';  
    }  
}
```

Клас Circle

```
import static java.lang.Math.PI;
import static java.lang.Math.pow;
public class Circle extends Figure{
    private double radius;

    public Circle(String name, double radius) {
        super(name);
        this.radius = radius;
    }

    public Circle(double radius) {
        super("No name");
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    @Override
    public double getArea() {
        return PI * pow(radius, 2);
    }

    @Override
    public double getPerimeter() {
        return 2 * PI * radius;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || o instanceof Circle) return false;
        Circle circle = (Circle) o;
        return circle.radius == radius;
    }

    @Override
    public String toString() {
        return "Circle{" +
            super.toString() +
            ", radius=" + radius +
            "} ";
    }
}
```

Клас Rectangle

```
public class Rectangle extends Figure{
    private double length;
    private double width;

    public Rectangle(String name, double length, double width) {
        super(name);
        this.length = length;
        this.width = width;
    }

    public Rectangle(double length, double width) {
        super("No name");
        this.length = length;
        this.width = width;
    }

    public Rectangle(String name, double a) {
        super(name);
        this.length = a;
        this.width = a;
    }

    public Rectangle(double a) {
        super("No name");
        this.length = a;
        this.width = a;
    }

    @Override
    public double getArea() {
        return 0.5 * (length + width);
    }

    @Override
    public double getPerimeter() {
        return 2 * (length + width);
    }

    public double getLength() {
        return length;
    }

    public double getWidth() {
        return width;
    }

    public void setSides(double length, double width){
        this.length = length;
        this.width = width;
    }

    public void setSides(double a){
```

```

        this.length = a;
        this.width = a;
    }

    @Override
    public String toString() {
        return "Rectangle{" +
            super.toString() +
            ", length=" + length +
            ", width=" + width +
            "} ";
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || o instanceof Rectangle) return false;
        Rectangle rectangle = (Rectangle) o;
        return rectangle.length == length && rectangle.width == width;
    }

    public boolean isSquare() {
        if (length == width) return true;
        return false;
    }
}

```

Клас Main

```

public class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle("ABCD", 4, 6);
        System.out.println(rectangle);
        Rectangle unnamedRectangle = new Rectangle(4, 6);
        System.out.println(unnamedRectangle);
        Rectangle square = new Rectangle("A1B1C1D1", 4);
        System.out.println(square);
        Rectangle unnamedSquare = new Rectangle(4);
        System.out.println(unnamedSquare);

        System.out.println("Area of ABCD = " + rectangle.getArea());
        System.out.println("Perimeter of ABCD = " +
rectangle.getPerimeter());
        System.out.println("Area of A1B1C1D1 = " + square.getArea());
        System.out.println("Perimeter of A1B1C1D1 = " +
square.getPerimeter());

        System.out.println("ABCD and unnamed rectangle are equal: " +
rectangle.equals(unnamedRectangle));
        System.out.println("ABCD and A1B1C1D1 are equal: " +
rectangle.equals(square));

        rectangle.setSides(8, 10);
        System.out.println(rectangle);

        square.setSides(5);
        System.out.println(square);

        System.out.println("ABCD is square: " + rectangle.isSquare());
    }
}

```

```

        System.out.println("A1B1C1D1 is square: " + rectangle.isSquare());

System.out.println("\n=====
\n");

        Circle circle = new Circle("O", 5);
        Circle unnamedCircle = new Circle(3);

        System.out.println(circle);
        System.out.println(unnamedCircle);

        System.out.println("Area of O = " + circle.getArea());
        System.out.println("Perimeter of O = " + circle.getPerimeter());
        System.out.println("Area of unnamed circle = " +
unnamedCircle.getArea());
        System.out.println("Perimeter of unnamed circle = " +
unnamedCircle.getPerimeter());

        System.out.println("O and unnamed circle are equal: " +
circle.equals(unnamedCircle));

        circle.setRadius(6);
        System.out.println(circle);

    }
}

```

Результат виконання програми:

```
Rectangle{Figure{name='ABCD'}, length=4.0, width=6.0}
Rectangle{Figure{name='No name'}, length=4.0, width=6.0}
Rectangle{Figure{name='A1B1C1D1'}, length=4.0, width=4.0}
Rectangle{Figure{name='No name'}, length=4.0, width=4.0}
Area of ABCD = 5.0
Perimeter of ABCD = 20.0
Area of A1B1C1D1 = 4.0
Perimeter of A1B1C1D1 = 16.0
ABCD and unnamed rectangle are equal: false
ABCD and A1B1C1D1 are equal: false
Rectangle{Figure{name='ABCD'}, length=8.0, width=10.0}
Rectangle{Figure{name='A1B1C1D1'}, length=5.0, width=5.0}
ABCD is square: false
A1B1C1D1 is square: false
```

=====

```
Circle{Figure{name='0'}, radius=5.0}
Circle{Figure{name='No name'}, radius=3.0}
Area of 0 = 78.53981633974483
Perimeter of 0 = 31.41592653589793
Area of unnamed circle = 28.274333882308138
Perimeter of unnamed circle = 18.84955592153876
0 and unnamed circle are equal: false
Circle{Figure{name='0'}, radius=6.0}
```

Висновок: у даній лабораторній роботі ми попрактикувалися в застосування принципів ООП. Проаналізувавши дану предметну область розробили базовий клас й класи-нащадки, реалізували методи бізнес-логіки, а також конструктори, сетери та гетери, методи equals() та toString().