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

Лабораторна робота №8

з дисципліни «Програмування-1. Основи програмування»

Тема “Основи ООП”

Виконали
студенти групи ІА-23:
Тюска А.Ю.
Хохол М. В.
Пожар Д.Ю.

Перевірив:
Колеснік В. М.

Київ 2022

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

Головний клас Main

```
public class Main {
    public static void main(String[] args) {
        try {
            CoordinatesOfPixel firstPixel = new CoordinatesOfPixel(1.8, 2.4, -
3.2, 2, 2);
            System.out.println(firstPixel.toString());

            System.out.println("--------------");

            ColorfulPixel firstColorfulPixel = new ColorfulPixel(1.8, 2.4, -3.2,
2, 2, "blue", "square", 0.8);
            System.out.println(firstColorfulPixel.toString());

            System.out.println("--------------");

            firstPixel.setHeight(3);
            firstPixel.setWidth(1);
            firstPixel.setCoordinateY(4);
            System.out.println(firstPixel.toString());

            System.out.println("--------------");

            CoordinatesOfPixel secondPixel = new CoordinatesOfPixel(1.8, 2.4, -
3.2, -3, 2);
            System.out.println(secondPixel.getArea());
        } catch (IllegalArgumentException e) {
            System.out.println("Exception! " + e.getMessage());
        }
    }
}
```

Клас ColorfulPixel – відповідає за зовнішній вигляд пікселя (колір, форма, прозорість, розміри)

```
public class ColorfulPixel extends CoordinatesOfPixel {
    private String color;
    private double transparency;

    private String shape;

    ColorfulPixel() {
        this.color = "white";
        this.transparency = 1.0;
    }

    ColorfulPixel(double x, double y, double z, int width, int height, String
color, String shape, double transparency) {
        super(x, y, z, width, height);
        this.color = color;
        this.transparency = transparency;
        this.shape = shape;
    }

    public String getColor() {
        return color;
    }

    public double getTransparency() {
        return transparency;
    }

    public String getShape() {
        return shape;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public void setTransparency(double transparency) {
        this.transparency = transparency;
    }

    public void setShape(String shape) {
        this.shape = shape;
    }

    @Override
    public String toString() {
        return "Pixel's characteristics: " + "\n" +
                "color: " + getColor() + "\n" +
                "transparency: " + getTransparency() + "\n" +
                "shape: " + getShape();
    }
}
```

Клас CoordinateOfPixel – відповідає за координати пікселя

```java
public class CoordinatesOfPixel {
    private double x;
    private double y;
    private double z;
    private int width;
    private int height;

    CoordinatesOfPixel() {
        this.x = 0;
        this.y = 0;
        this.z = 0;
    }

    CoordinatesOfPixel(double x, double y, double z, int width, int height) {
        this.x = x;
        this.y = y;
        this.z = z;
        this.width = width;
        this.height = height;
    }

    public double getCoordinateX() {
        return x;
    }

    public double getCoordinateY() {
        return y;
    }

    public double getCoordinateZ() {
        return z;
    }

    public int getWidth() {
        return width;
    }

    public int getHeight() {
        return height;
    }

    public void setCoordinateX(double x) {
        this.x = x;
    }

    public void setCoordinateY(double y) {
        this.y = y;
    }

    public void setCoordinateZ(double z) {
        this.z = z;
    }

    public void setWidth(int width) {
        this.width = width;
    }

    public void setHeight(int height) {
        this.height = height;
    }

    @Override
    public String toString() {
        return "Pixel's coordinates: (" + getCoordinateX() + ", " +
getCoordinateY() + ", " + getCoordinateZ() + ")" + "\n"
```

```
                + "Pixel's area: " + getArea();
    }

    public int getArea() {
        if (this.width <= 0 || this.height <= 0) {
            throw new IllegalArgumentException("Width and height can't be null or
negative");
        } else {
            return width * height;
        }
    }
}
```

РЕЗУЛЬТАТ:

```
Pixel's coordinates: (1.8, 2.4, -3.2)
Pixel's area: 4
--------------
Pixel's characteristics:
color: blue
transparency: 0.8
shape: square
--------------
Pixel's coordinates: (1.8, 4.0, -3.2)
Pixel's area: 3
--------------
Exception! Width and height can't be null or negative
```

Висновок

Виконавши лабораторну роботу, ми на практиці використали наслідування (ключові слова this, super), перевантаження (overloading) та заміщення(overriding) методів; перевантаження (overloading) конструкторів. Розробили базовий (Main) а також допоміжні (ColorfulPixel, CoordinateOfPixel) класи. Передбачили відповідні методи бізнес-логіки