

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium: 1

Data: 23.01.2022

Temat: " Przekształcenia 2D w bibliotece Java 2D";

Wariant: 13/1

Przemysław Holisz

Informatyka I stopień, stacjonarne,

4 semestr, Gr.4/2b

Zadanie 1

1. Polecenie:

Program Transform2D.java rysuje obraz shuttle.jpg w panelu. Narysować zamiast obrazu wielokat według wariantu (liczba n) w panelu wyświetlania. Panel ma wymiary 600 na 600 pikseli, a wielokat ma promień 150 pikseli. Okno zawiera również wyskakujące menu z etykietą "Transform:". Opcje w menu to "None" i cyfry od 1 do 9. W tym programie menu wyskakujące nie działa. Zadanie polega na dodaniu kodu do metody paintComponent(). (Miejsce jest oznaczone TODO.) Kiedy wybór ma wartość 0, strona powinna wyświetlić obraz nietransformowany. W przypadku innych możliwych wartości musisz zastosować przekształcenie (lub będziesz potrzebował kombinacji przekształceń) dla każdej z wartości od 1 do 9 (patrz Fig. 1).

2. Wprowadzane dane:

```
int[] xpoints = new int[13];
int[] ypoints = new int[13];

for(int i=1; i<=13; i++)
{
    xpoints[i-1] = (int) (150*Math.cos((2*Math.PI/13)*i));
}
for(int i=1; i<=13; i++)
{
    ypoints[i-1] = (int) (150*Math.sin((2*Math.PI/13)*i));
}

Polygon pentagon = new Polygon(xpoints, ypoints, 13);
g2.fillPolygon(pentagon);}}
```

3. Wykorzystane komendy:

a) kod źródłowy

```
switch(whichTransform)
{
    case 0: break;
    case 1:
        g2.scale(0.5,0.5);
        break;

    case 2:

        g2.rotate(0.5);
        break;

    case 3:
        g2.scale(0.5,0.8);
        g2.rotate(Math.toRadians(180));
        break;

    case 4:
        g2.shear(0.35,0);
        break;
```

```
case 5:
    g2.scale(1,0.3);
    g2.translate(0,-900);
    break;

case 6:
    g2.shear(0,-0.5);
    g2.rotate(Math.PI / 2);
    break;

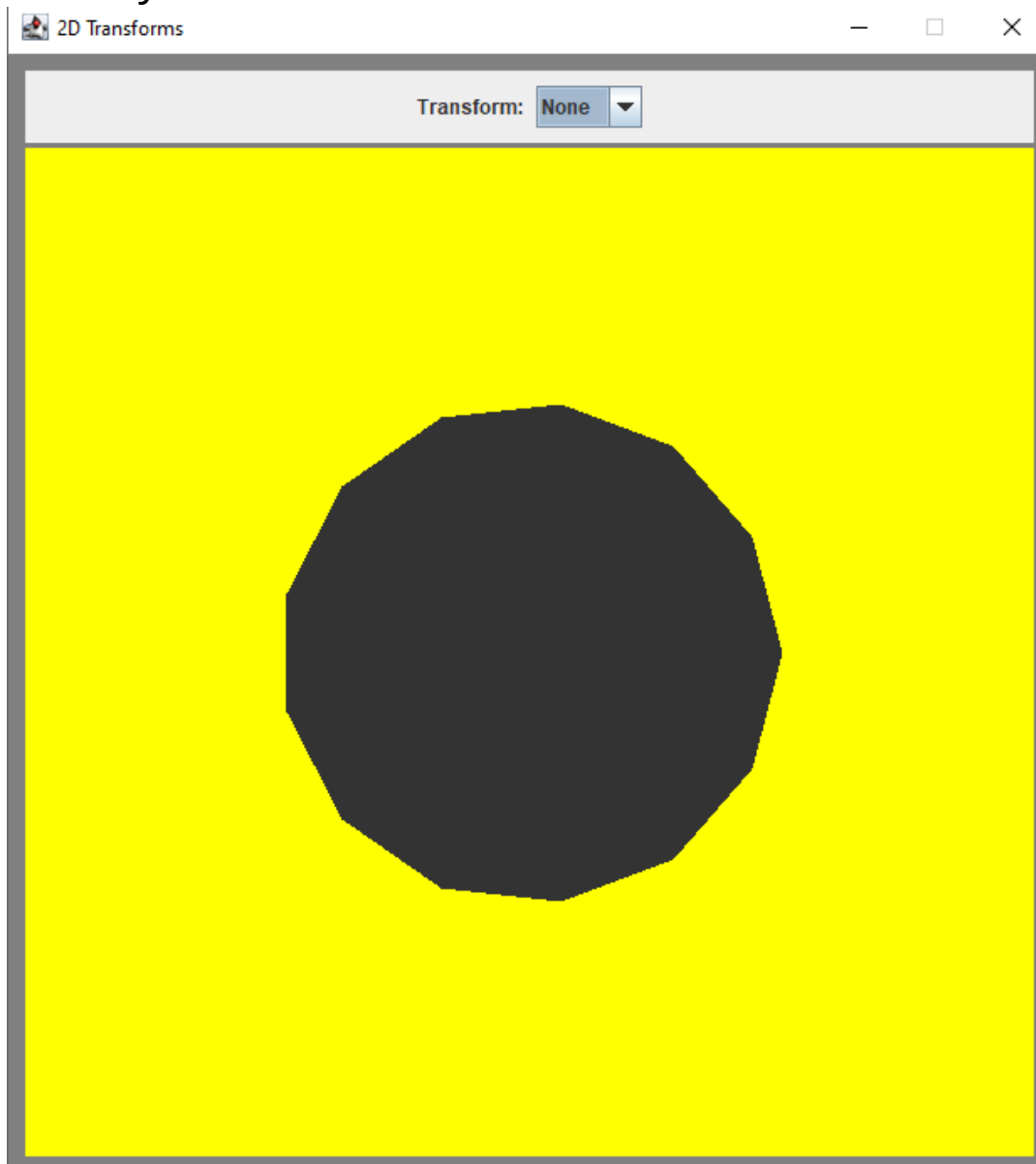
case 7:
    g2.scale(0.5, 1);
    g2.rotate(Math.PI);
    break;

case 8:
    g2.rotate(Math.toRadians(30));
    g2.scale(1,0.3);
    g2.translate(0,200);
    break;

case 9:
    g2.translate(100, 0);
    g2.shear(0, 0.25);
    g2.rotate(Math.PI);
    break;

}
```

4. Wynik działania:



AD.1. Wstawianie obrazu do kodu nie chce funkcjonować.

5. Wnioski:

Na podstawie otrzymanego wyniku można stwierdzić, że przy użyciu podstawowym metod z klasy Graphics biblioteki JAVA 2D, można stworzyć proste rysunki.

6. Polecenie:

Narysować figurę określoną wariantem (patrz Fig. 2). Dostępne są trzy podstawowe kształty: `circle()`, `square()` i `triangle()`. Zacznij od programu `TransformedShapes.java`. TODO. Możesz użyć poleceń do rysowania, takich jak `g.fillRect()` itp.

7. Wprowadzane dane:

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    g2 = (Graphics2D)g.create();
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
    g2.translate(300,300);
    g2.scale(2,2);
    g2.setColor(Color.BLACK);
    circle();

    g2 = (Graphics2D)g.create();
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
    g2.translate(300,300);
    g2.scale(1,1);
    g2.setColor(Color.YELLOW);
    square();

    resetTransform();
}
```

8. Wykorzystane komendy:

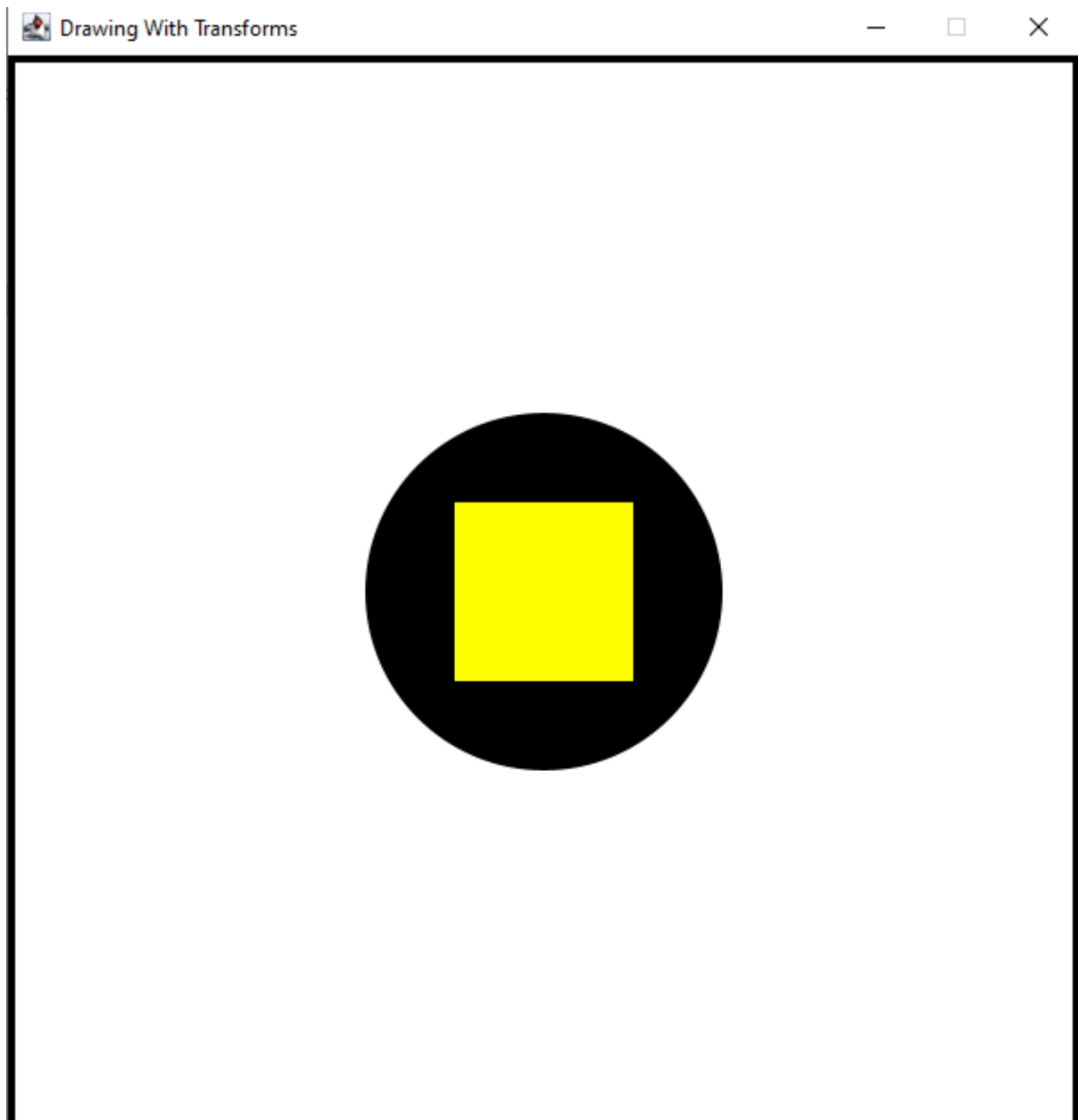
```
public class TransformedShapes extends JPanel {
    private Graphics2D g2;

    private void resetTransform() {
        g2.setTransform(new AffineTransform());
    }

    private void circle() {
        g2.fillOval(-50,-50,100,100);
    }

    private void square() {
        g2.fillRect(-50,-50,100,100);}
}
```

9. Wynik działania:



10. Wnioski:

Na podstawie otrzymanego wyniku można stwierdzić, że używając języka java jesteśmy w stanie wygenerować kształty i je dowolnie zmieniać.