

**Міністерство освіти і науки України**  
**Львівський національний університет імені Івана Франка**  
Факультет прикладної математики та інформатики

Звіт

Лабораторна робота №4

Тема: «Форми»

з дисципліни "Паралельні та розподільні обчислення"

Виконав студент групи ПМі-31  
Яцуляк Андрій

Львів 2023 р.

**Мета:** Написати програму, яка в кожному окремому потоці представляє певне графічне представлення.

### Хід роботи

Завдання виконав мовою програмування Java у середовищі IntelliJ IDEA. Я написав Завдання 1. Для початку, створив окремі 4 класи, де кожен клас має своє певне графічне представлення:

1. Перший клас малює кульку, яка бігає по графічному полю:

```
1  import javafx.application.Platform;
2      import javafx.scene.layout.Pane;
3      import javafx.scene.paint.Color;
4      import javafx.scene.shape.Circle;|
    2 usages
5      public class BallAnimation {
    8 usages
6          private Circle ball;
    5 usages
7          private Pane pane;
    3 usages
8          private volatile boolean paused = false;
    7 usages
9          private Thread animationThread;
    2 usages
10         private double x, y;
    6 usages
11         private double radius;
    3 usages
12         private double speed;
    4 usages
13         private double velocityX;
    4 usages
14         private double velocityY;
```

```

16 public BallAnimation(Pane pane, double x, double y, double radius, double speed, double velocityX, double velocityY) {...}
25
26 1 usage
public Circle initializeBall() {...}
32
33 1 usage
public Pane getPane() { return pane; }
36
37 1 usage
public void pauseAnimation() { paused = true; }
40
41 1 usage
public void resumeAnimation() {...}
45 1 usage
public void stopAnimation() {...}
50
51 2 usages
public void startAnimation() {...}
52 }

```

2. Другий клас збільшує розміри прямокутника:

```

1 import javafx.application.Platform;
2 import javafx.scene.layout.Pane;
3 import javafx.scene.paint.Color;
4 import javafx.scene.shape.Rectangle;
5
6 2 usages
public class RectangleAnimation {
7     8 usages
private Rectangle rectangle;
8     5 usages
private Pane pane;
9     3 usages
private volatile boolean paused = false;
10    5 usages
private Thread animationThread;
11    3 usages
private double x, y;
12    2 usages
private double width;
13    2 usages
private double height;
14    4 usages
private double velocityX;
15    4 usages
private double velocityY;

```

```

16 public RectangleAnimation(Pane pane, double x, double y, double width, double height, double velocityX, double velocityY) {...}
    1 usage
25 public Rectangle initializeRectangle() {...}
    1 usage
31 public Pane getPane() { return pane; }
    1 usage
34 public void pauseAnimation() { paused = true; }
    1 usage
37 public void resumeAnimation() { paused = false; }
    1 usage
40 public void stopAnimation() {...}
    1 usage
45 public void startAnimation() {...}
92 }

```

### 3. Третій клас малює синусоїду:

```

1 import javafx.application.Platform;
2 import javafx.scene.layout.Pane;
3 import javafx.scene.shape.MoveTo;
4 import javafx.scene.shape.Path;
5 import javafx.scene.shape.QuadCurveTo;
6 import javafx.scene.shape.StrokeType;
    2 usages
7 public class SineWaveAnimation {
    3 usages
8     private Pane pane;
    8 usages
9     private Path sineWave;
    4 usages
10    private volatile boolean paused = false;
    10 usages
11    private Thread animationThread;
    2 usages
12    private double amplitude;
    2 usages
13    private double frequency;
    4 usages
14    private double width;
    4 usages
15    private double height;
    3 usages
16    private double phase;

```

```

18 public SineWaveAnimation(Pane pane, double amplitude, double frequency, double width, double height) {...}
    1 usage
26 public void initializeSineWave() {...}
    1 usage
34 public Pane getPane() { return pane; }
    1 usage
37 public void pauseAnimation() { paused = true; }
    1 usage
40 public void resumeAnimation() { paused = false; }
    1 usage
43 public void stopAnimation() {...}
    1 usage
48 public void startAnimation() {...}
    2 usages
81 private void updateSineWave() {...}
94 }

```

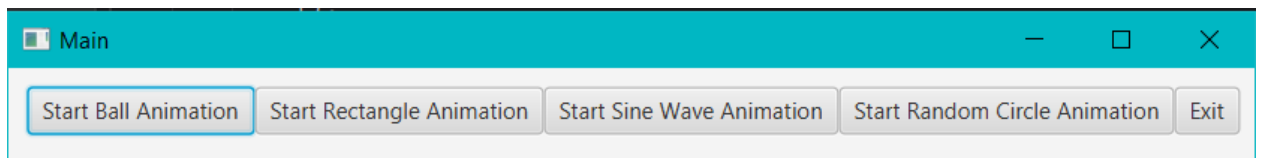
4. Четвертий клас в рандомному місці, рандомного розміру і рандомного кольору малює круг:

```

1 import javafx.animation.KeyFrame;
2 import javafx.animation.Timeline;
3 import javafx.application.Platform;
4 import javafx.scene.layout.Pane;
5 import javafx.scene.paint.Color;
6 import javafx.scene.shape.Circle;
7 import javafx.util.Duration;
    2 usages
8 public class RandomCircleGenerator {
    5 usages
9     private Pane pane;
    4 usages
10    private volatile boolean paused = false;
    1 usage
11    public RandomCircleGenerator(Pane pane) { this.pane = pane; }
    1 usage
14    public Pane getPane() { return pane; }
    2 usages
17    public void initializeCircles() {...}
    1 usage
24    private void createRandomCircle() {...}
    1 usage
39    public void pauseAnimation() { paused = true; }
    1 usage
42    public void resumeAnimation() { paused = false; }
    1 usage
45    public void stopAnimation() { paused = true; }
48 }

```

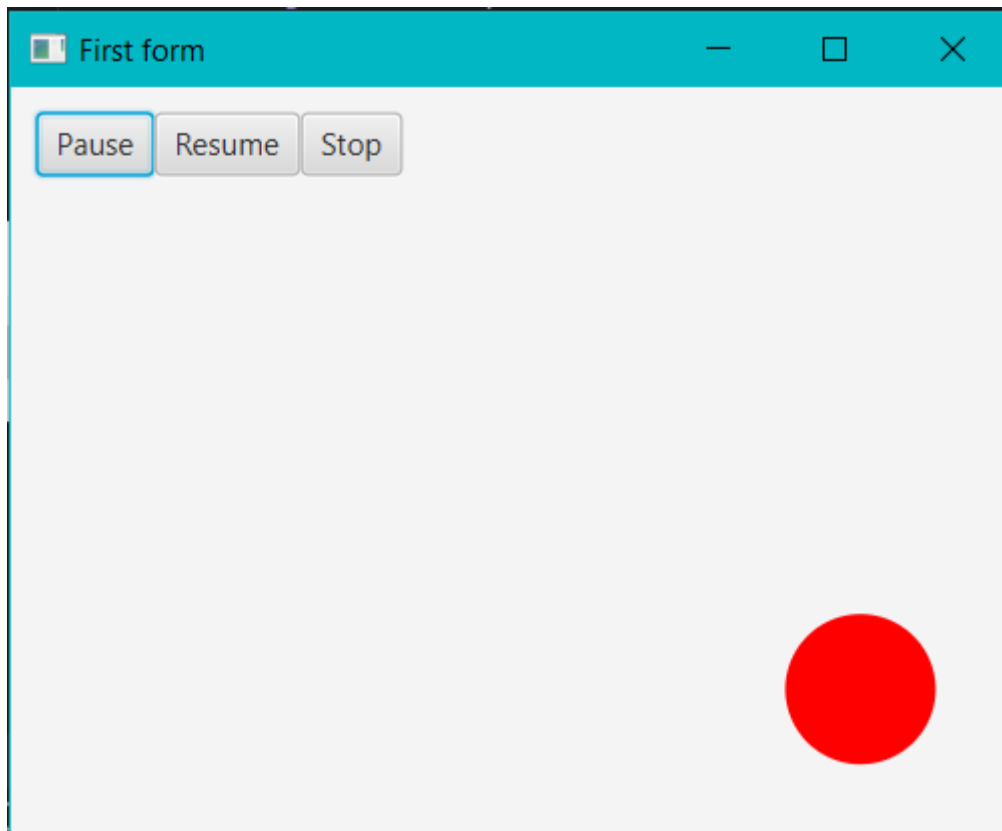
Спочатку в головному потоці запускаю головне вікно, на якому є кнопки.

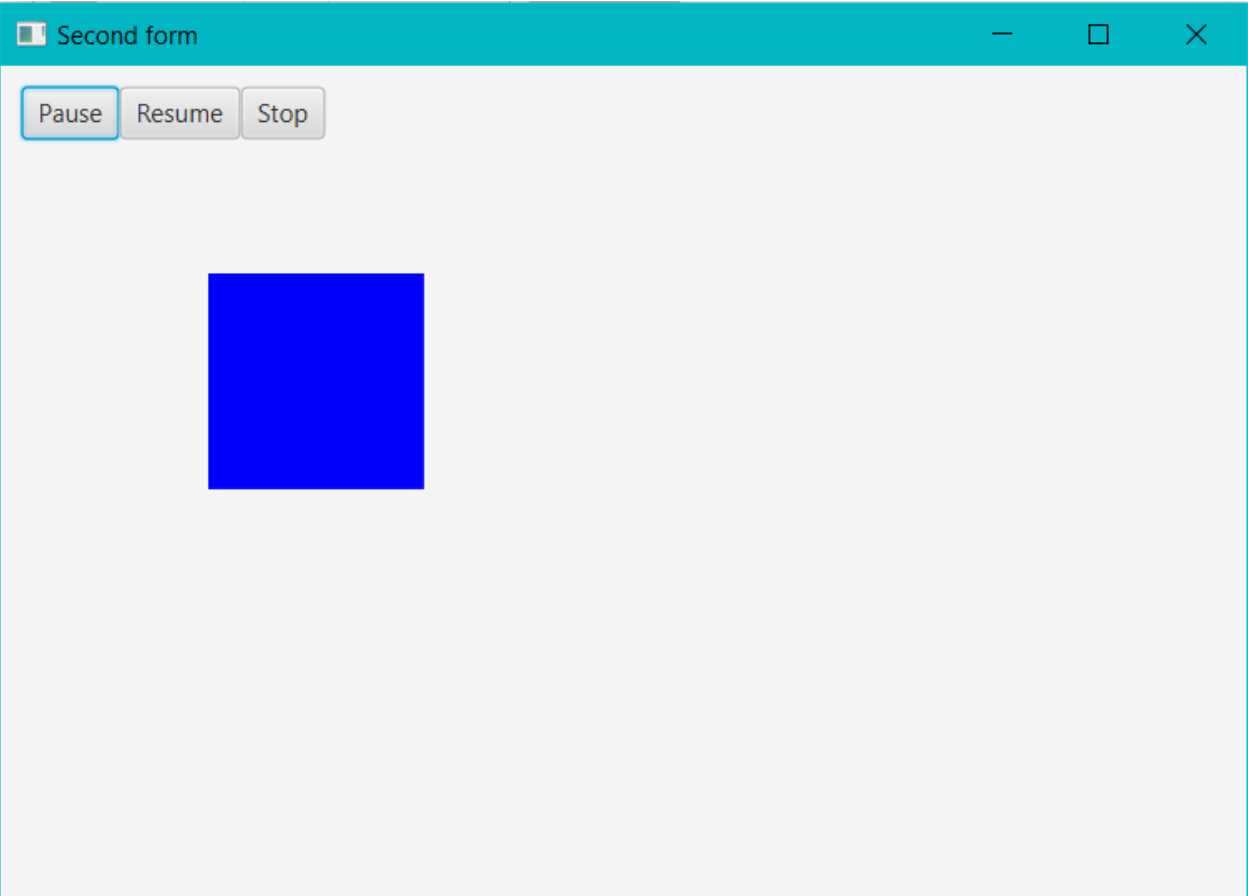


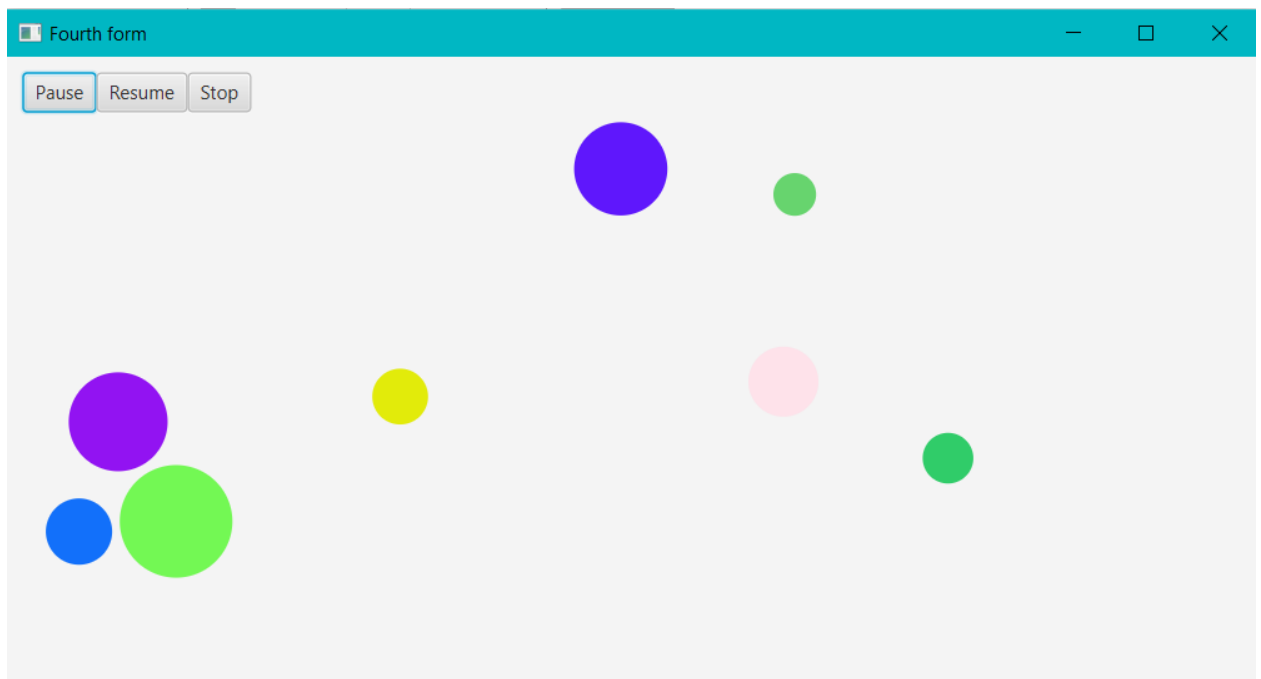
Далі, вибравши певну кнопку, в окремому потоці запускається відповідний клас.

Далі продемонструю кожну з цих кнопок

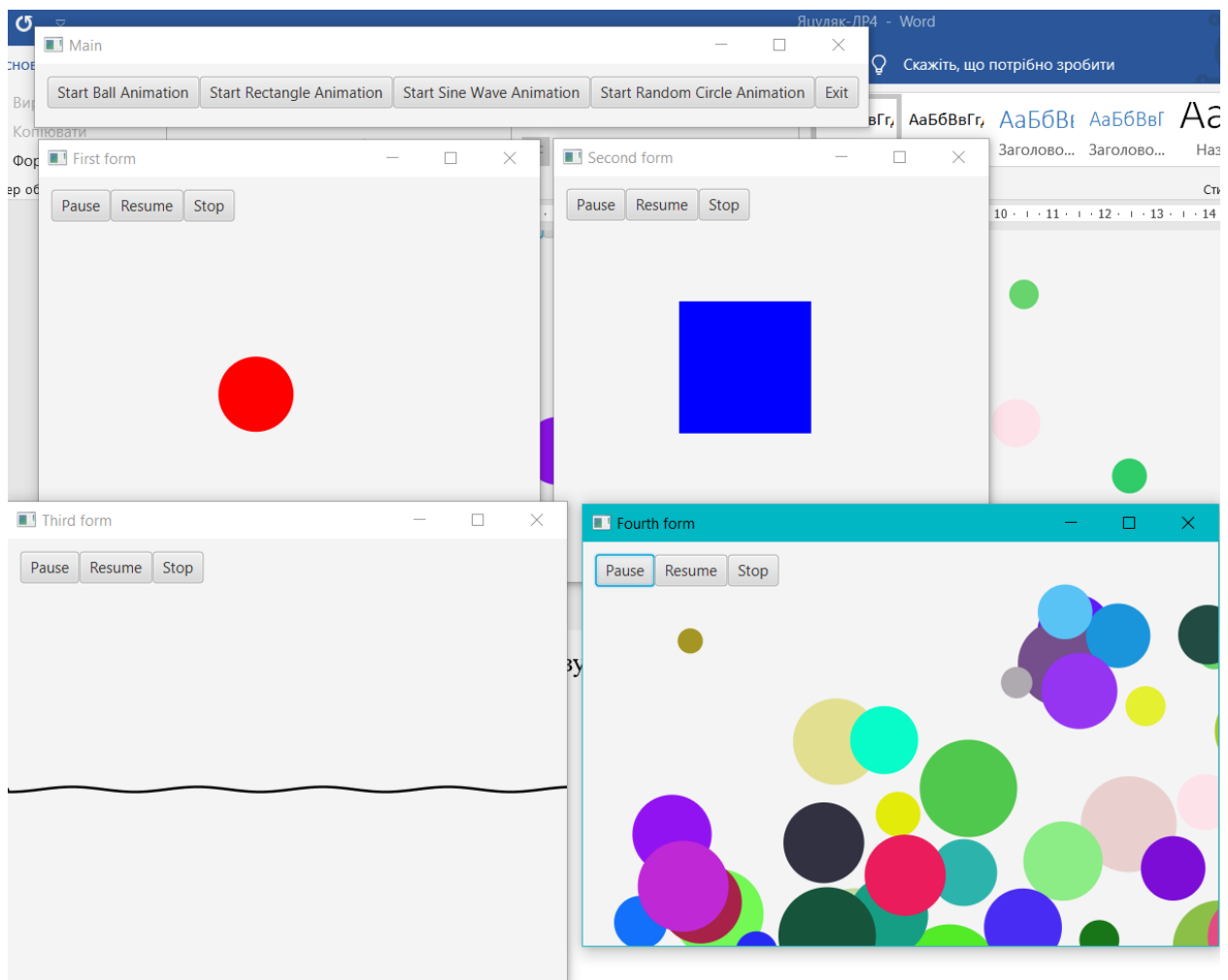
В кожному такому вікні є кнопки "Pause", "Resume" і "Stop". Вони зупиняють, відновлюють і завершають виконання потоку відповідно.







А це результат разом:



**Висновок.** Під час виконання лабораторної роботи я написав програму, яка в кожному окремому потоці малює певне графічне представлення.