

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ
УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота №3

з дисципліни “Структура файлів формату .bmp. Анімація примітивів
за допомогою засобів
бібліотеки JavaFX ”

Виконав
студент III курсу
групи КП-83
Коваль Андрій Олександрович
(прізвище, ім’я, по батькові)
варіант № 8

Зарахована
“ ____ ” “ _____ ” 20__ р.
викладачем Шкурат Оксаною Сергіївною
(прізвище, ім’я, по батькові)


Київ 2021

Завдання

За допомогою примітивів JavaFX максимально реально зобразити персонажа за варіантом та виконати його 2D анімацію. Для анімації скористатися стандартними засобами бібліотеки JavaFX.

Обов'язковою є реалізація таких видів анімації:

- 1) переміщення;
- 2) поворот;
- 3) масштабування.

Варіант	Сцена
8	

Лістинг коду програми

Drum.java

```
package com.lab3;

import javafx.scene.Node;
import javafx.scene.paint.Color;
import javafx.scene.shape.*;

import java.util.ArrayList;
import java.util.List;

public class Drum {

    private static Color stickColor = Color.rgb(191, 162, 89);
    private static Color surfaceColor = Color.rgb(237, 208, 66);
    private static Color bodyColor = Color.RED;
    private static Color skeletonStickColor = Color.rgb(200, 200, 200);
    private static Color beltColor = Color.rgb(45, 124, 181);

    public static Node[] getRightStick() {
        Polygon stick = new Polygon(
            355, 255,
            496, 122,
            525, 108,
            532, 122,
            515, 145,
            368, 259
        );
    }
}
```

```

    );
    stick.setFill(stickColor);
    stick.setStroke(Color.BLACK);
    stick.setStrokeType(StrokeType.OUTSIDE);

    Circle circle = new Circle(350, 270, 30);
    circle.setFill(stickColor);
    circle.setStroke(Color.BLACK);
    circle.setStrokeType(StrokeType.OUTSIDE);

    Circle shiny = new Circle(360, 265, 10);
    shiny.setFill(Color.WHEAT);
    return new Node[] { stick, circle, shiny };
}

public static Node[] getLeftStick() {
    Polygon stick = new Polygon(
        387, 144,
        179, 117,
        165, 121,
        160, 140,
        170, 145,
        384, 151
    );
    stick.setFill(stickColor);
    stick.setStroke(Color.BLACK);
    stick.setStrokeType(StrokeType.OUTSIDE);

    Circle circle = new Circle(404, 156, 30);
    circle.setFill(stickColor);
    circle.setStroke(Color.BLACK);
    circle.setStrokeType(StrokeType.OUTSIDE);

    Circle shiny = new Circle(414, 161, 10);
    shiny.setFill(Color.WHEAT);
    return new Node[] { stick, circle, shiny };
}

public static Node[] getDrumSurface() {
    Ellipse ellipse = new Ellipse(356, 272, 200, 50);
    ellipse.setFill(surfaceColor);
    ellipse.setStroke(Color.BLACK);
    ellipse.setStrokeType(StrokeType.OUTSIDE);
    return new Node[] { ellipse };
}

public static Node[] getDrumBody() {
    Polygon body = new Polygon(
        161, 271,
        555, 271,
        555, 274,
        555, 440,
        481, 460,
        424, 465,
        389, 467,
        343, 468,
        312, 467,
        247, 459,
        197, 450,
        160, 440
    );
    body.setFill(bodyColor);
    body.setStroke(Color.BLACK);
    body.setStrokeType(StrokeType.OUTSIDE);
    return new Node[] { body };
}

public static Node[] getForegroundBodySkeleton() {
    List<Node> nodes = new ArrayList<Node>();
    Polygon[] skeletonSticks = new Polygon[] {
        new Polygon(
            552, 277,
            548, 439,

```

```

        553, 446,
        556, 445
    ),
    new Polygon(
        458, 310,
        449, 462,
        460, 468,
        465, 461
    ),
    new Polygon(
        290, 318,
        281, 465,
        285, 471,
        295, 465
    ),
    new Polygon(
        158, 270,
        155, 433,
        166, 450,
        170, 445
    )
);
for (Polygon skeletonStick : skeletonSticks) {
    skeletonStick.setFill(skeletonStickColor);
    skeletonStick.setStroke(Color.BLACK);
    skeletonStick.setStrokeType(StrokeType.OUTSIDE);
    nodes.add(skeletonStick);
}

Circle[] circles = new Circle[] {
    new Circle(553, 266, 10),
    new Circle(456, 305, 10),
    new Circle(288, 308, 10),
    new Circle(160, 266, 10)
};
for (Circle circle : circles) {
    circle.setFill(skeletonStickColor);
    circle.setStroke(Color.BLACK);
    circle.setStrokeType(StrokeType.OUTSIDE);
    nodes.add(circle);
}
Node[] nodesArr = new Node[nodes.size()];
return nodes.toArray(nodesArr);
}

public static Node[] getBackgroundBodySkeleton() {
    Circle[] circles = new Circle[] {
        new Circle(456, 220, 10),
        new Circle(288, 215, 10),
    };
    for (Circle circle : circles) {
        circle.setFill(skeletonStickColor);
        circle.setStroke(Color.BLACK);
        circle.setStrokeType(StrokeType.OUTSIDE);
    }

    return circles;
}

public static Node[] getBelt() {
    Polygon belt = new Polygon(
        551, 272,
        621, 303,
        640, 344,
        636, 392,
        622, 465,
        592, 525,
        494, 560,
        449, 526,
        416, 500,
        372, 495,
        325, 499,

```

```

        281, 510,
        245, 537,
        174, 530,
        98, 516,
        87, 477,
        98, 436,
        70, 397,
        60, 356,
        76, 302,
        111, 274,
        154, 270,
        // back
        154, 256,
        87, 254,
        58, 278,
        30, 326,
        31, 396,
        64, 439,
        63, 514,
        87, 555,
        160, 565,
        235, 566,
        287, 559,
        309, 533,
        375, 521,
        423, 539,
        460, 581,
        511, 587,
        579, 576,
        624, 541,
        654, 514,
        675, 455,
        679, 389,
        669, 309,
        622, 278,
        551, 261
    );
    belt.setFill(beltColor);
    belt.setStroke(Color.BLACK);
    belt.setStrokeType(StrokeType.OUTSIDE);
    return new Node[] { belt };
}
}

```

Main.java

```

package com.lab3;

import javafx.animation.*;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.paint.Paint;
import javafx.stage.Stage;
import javafx.util.Duration;

public class Main extends Application {
    Paint bgColor = Color.rgb(238, 255, 138);

    int WIDTH = 900;
    int HEIGHT = 700;
    int ANIMATION_TRANSLATION_DURATION = 200;
    int ANIMATION_SCALE_DURATION = 400;
    int ANIMATION_ROTATE_DURATION = 300;
}

```

```

Group root;
Scene scene;

public static void main (String args[]) {
    launch(args);
}

@Override
public void start(Stage primaryStage) {
    root = new Group();
    scene = new Scene (root, WIDTH, HEIGHT);
    scene.setFill(bgColor);
    root.getChildren().addAll(Drum.getDrumBody());
    root.getChildren().addAll(Drum.getBackgroundBodySkeleton());
    root.getChildren().addAll(Drum.getBelt());
    root.getChildren().addAll(Drum.getDrumSurface());
    root.getChildren().addAll(Drum.getRightStick());
    root.getChildren().addAll(Drum.getLeftStick());
    root.getChildren().addAll(Drum.getForegroundBodySkeleton());

    TranslateTransition translateTo = new
TranslateTransition(Duration.millis(ANIMATION_TRANSLATION_DURATION), root);
    translateTo.setFromY(-100);
    translateTo.setToY(100);
    translateTo.setAutoReverse(true);

    TranslateTransition translateFrom = new
TranslateTransition(Duration.millis(ANIMATION_TRANSLATION_DURATION), root);
    translateFrom.setFromY(100);
    translateFrom.setToY(-100);
    translateFrom.setAutoReverse(true);

    ScaleTransition scaleFrom = new ScaleTransition(Duration.millis(ANIMATION_SCALE_DURATION),
root);
    scaleFrom.setToX(1);
    scaleFrom.setToY(1);

    ScaleTransition scaleTo = new ScaleTransition(Duration.millis(ANIMATION_SCALE_DURATION),
root);
    scaleTo.setToY(0.6);
    scaleTo.setToX(0.8);

    RotateTransition rotate = new RotateTransition(Duration.millis(ANIMATION_ROTATE_DURATION),
root);
    rotate.setByAngle(360f);
    rotate.setCycleCount(1);

    SequentialTransition scale = new SequentialTransition();
    scale.getChildren().addAll(
        scaleTo,
        scaleFrom
    );
    scale.setCycleCount(1);

    SequentialTransition translate = new SequentialTransition();
    translate.getChildren().addAll(
        translateTo,
        translateFrom
    );
    translate.setCycleCount(2);

    SequentialTransition animations = new SequentialTransition();
    animations.getChildren().addAll(
        scale,
        rotate,
        translate
    );
    animations.setCycleCount(Timeline.INDEFINITE);
    animations.play();

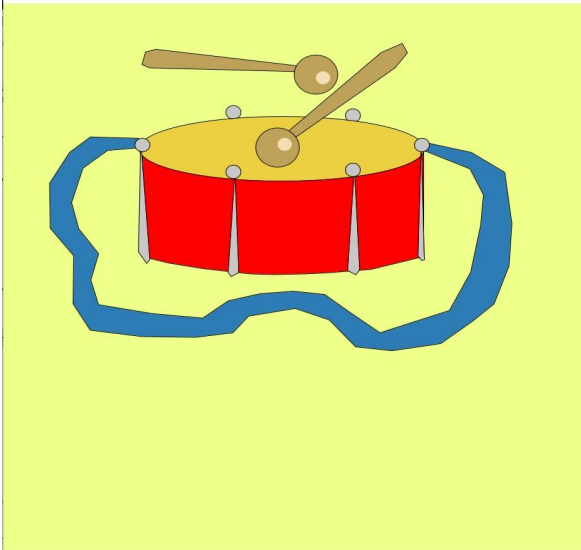
    primaryStage.setTitle("Коваль Андрій КП-83 Лаб3");
    primaryStage.setScene(scene);
}

```

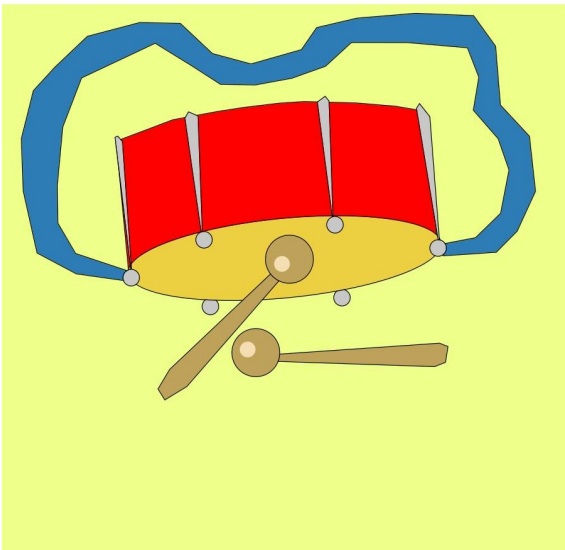
```
    primaryStage.show();  
}  
}
```

Результат

Коваль Андрій КТ-83 Ла63



Коваль Андрій КТ-83 Ла63



Коваль Андрій КТ-83 Ла63

