

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ КИЇВСЬКИЙ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики
Інженерія програмного забезпечення

ЛАБОРАТОРНА РОБОТА №3
з дисципліни *“Математичні та алгоритмічні основи
комп'ютерної графіки”*

Виконав:

студент 3-го курсу, групи КП-83,
спеціальності 121 – Інженерія
програмного забезпечення *Медведєв
Ілля Анатолійович*

Зарахована:

Шкурат О. С.

Київ – 2021

Мета:

- 1) вивчення структури та особливостей використання файлів формату *.bmp*;
- 2) вивчення стандартних засобів *JavaFX* для візуалізації зображення;
- 3) вивчення засобів анімації примітивів в *JavaFX*.

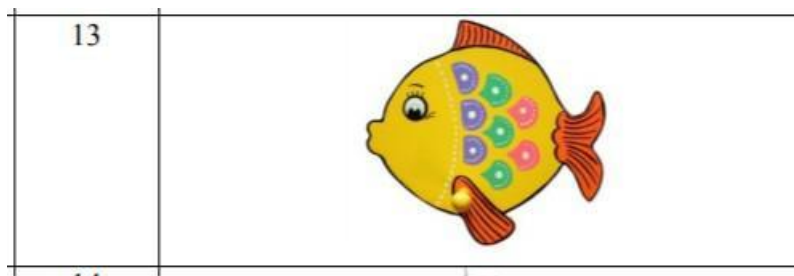
Завдання

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.

Студентам пропонується скористатися розглянутими класами для читання, обробки та збереження зображень формату *.bmp* з метою використання рисунку для створення траєкторії руху або меж, в яких дозволений рух об'єктів. В даному випадку рекомендується використовувати кольори великої контрастності для різних призначень (наприклад, чорний колір відповідатиме за траєкторію руху, а інші кольори – заборонятимуть рух).

Варіант 13:



Текст коду програм

App.java

```
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.scene.shape.*;
import javafx.stage.Stage;
import javafx.util.Duration;

public class App extends Application {

    private static double X(double originalX) {
        return originalX + 300;
    }

    private static double Y(double originalY) {
        return originalY + 50;
    }

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

    @Override
    public void start(Stage primaryStage) {
        Group root = new Group();
        Scene scene = new Scene(root, 600, 300);

        topFin(root);
        body(root);
        bottomFin(root);
        rightFin(root);
        eye(root);
        brow(root);
        faceLine(root);
        squama(root);

        animation(root);

        primaryStage.setResizable(false);
        primaryStage.setTitle("lab3");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    private void body(Group root) {
        double x = X(10);
        double y = Y(40);
        MoveTo mt2 = new MoveTo(x, y);
        QuadCurveTo qc8 = new QuadCurveTo(x - 10, y, x -= 8, y += 5);
        QuadCurveTo qc9 = new QuadCurveTo(x - 2, y + 8, x += 8, y += 8);
        QuadCurveTo qc10 = new QuadCurveTo(x + 40, y + 60, x += 80, y -=
        6.5); QuadCurveTo qc11 = new QuadCurveTo(x - 40, y - 60, x -= 80, y -
        = 6.5); Path background = new Path();
```

```

        background.setStrokeWidth(4);
        background.setStroke(Color.BLACK);
        background.setFill(Color.rgb(232, 194, 18));
        background.getElements().addAll(mt2, qc8, qc9, qc10, qc11);
        root.getChildren().add(background);
    }

    private void squama(Group root) {
        double x = X(49);
        double y = Y(26);
        double[] angles = new double[]{-80, -97, -105};
        for (int i = 0; i < 3; i++) {
            // Purple
            squamaElement(root, x, y, angles[i], Color.rgb(128, 119, 181));
            // Green
            squamaElement(root, x + 10, y + 7, angles[i], Color.rgb(100, 183, 132));
            // Rose
            if(i > 0)
                squamaElement(root, x + 20, y + 1, angles[i], Color.rgb(242, 113, 122));

            y += 15;
        }
    }

    private void squamaElement(Group root, double x, double y, double angle,
Paint fillColor) {
        Arc outerArc = new Arc(x, y, 7, 7, angle, 200);
        outerArc.setFill(fillColor);

        Arc nestedArc = new Arc(x, y, 5.5, 6, angle,
200); nestedArc.setStroke(Color.WHITE);
        nestedArc.setFill(Color.TRANSPARENT);
        nestedArc.getStrokeDashArray().addAll(2d);

        Ellipse dot = new Ellipse(x, y, 1, 1);
        dot.setFill(Color.WHITE);

        root.getChildren().addAll(outerArc, nestedArc, dot);
    }

    private void eye(Group root) {
        double stX = X(30);
        double stY = Y(38);
        double x = stX;
        double y = stY;
        double d = 12;
        Ellipse whiteEllipse = new Ellipse(x, y, d / 2, d /
2); whiteEllipse.setFill(Color.WHITE);
        whiteEllipse.setStroke(Color.BLACK);
        whiteEllipse.setStrokeWidth(2);

        Ellipse lastEllipse = new Ellipse(x, y, d / 2, d /
2); lastEllipse.setFill(Color.TRANSPARENT);
        lastEllipse.setStroke(Color.BLACK);
        lastEllipse.setStrokeWidth(2);

        y += 1;
        d = 9;
        Ellipse blueEllipse = new Ellipse(x, y, d / 2, d /
2); blueEllipse.setFill(Color.LIGHTBLUE);

        y += 2;
    }

```

```

        d = 7;
        Ellipse blackEllipse = new Ellipse(x, y, d / 2, d /
2); blackEllipse.setFill(Color.BLACK);

        y = stY + 1;
        d = 3;
        Ellipse smallWhiteEllipse = new Ellipse(x, y, d / 2, d /
2); smallWhiteEllipse.setFill(Color.WHITE);

        root.getChildren().addAll(whiteEllipse, blueEllipse,
blackEllipse, smallWhiteEllipse, lastEllipse);
    }

    private void topFin(Group root) {
        double x = X(40);
        double y = Y(15);
        MoveTo mt2 = new MoveTo(x, y);
        QuadCurveTo qc8 = new QuadCurveTo(x + 4, y - 10, x += 8, y -= 15);
        QuadCurveTo qc9 = new QuadCurveTo(x + 18, y + 10, x += 30, y += 30);
        Path background = new Path();
        background.setStrokeWidth(4);
        background.setStroke(Color.BLACK);
        background.setFill(Color.rgb(237, 104, 62));
        background.getElements().addAll(mt2, qc8, qc9);
        root.getChildren().add(background);
    }

    private void bottomFin(Group root) {
        double stX = X(45);
        double stY = Y(65);
        double x = stX;
        double y = stY;
        MoveTo mt2 = new MoveTo(x, y);
        QuadCurveTo qc8 = new QuadCurveTo(x + 8, y + 3, x += 10, y += 5);
        QuadCurveTo qc9 = new QuadCurveTo(x + 10, y + 10, x += 14, y += 15);
        QuadCurveTo qc10 = new QuadCurveTo(x - 7, y + 10, x -= 20, y += 8);
        QuadCurveTo qc11 = new QuadCurveTo(stX - 3, stY + 10, stX, stY);
        Path background = new Path();
        background.setStrokeWidth(4);
        background.setStroke(Color.BLACK);
        background.setFill(Color.rgb(237, 104, 62));
        background.getElements().addAll(mt2, qc8, qc9, qc10,
qc11); root.getChildren().add(background);
    }

    private void rightFin(Group root) {
        double stX = X(90);
        double stY = Y(48);
        double x = stX;
        double y = stY;
        MoveTo mt2 = new MoveTo(x, y);
        QuadCurveTo qc8 = new QuadCurveTo(x + 10, y - 3, x += 20, y -= 16);
        QuadCurveTo qc9 = new QuadCurveTo(x + 3, y + 13, x -= 2, y += 16);
        QuadCurveTo qc10 = new QuadCurveTo(x + 5, y + 13, x, y += 16);
        QuadCurveTo qc11 = new QuadCurveTo(stX + 8, stY - 3, stX, stY);
        Path background = new Path();
        background.setStrokeWidth(4);
        background.setStroke(Color.BLACK);
        background.setFill(Color.rgb(237, 104, 62));
        background.getElements().addAll(mt2, qc8, qc9, qc10,
qc11); root.getChildren().add(background);
    }

```

```

private void faceLine(Group root) {
    double x = X(14);
    double y = Y(43);
    Arc arc = new Arc(x, y, 30, 40, -47, 78);
    arc.setStroke(Color.WHITE);
    arc.setFill(Color.TRANSPARENT);
    arc.getStrokeDashArray().addAll(2d);
    arc.setStrokeDashOffset(5);
    root.getChildren().add(arc);
}

private void brow(Group root) {
    double x = X(30);
    double y = Y(127);
    Arc brow = new Arc(x, y, 20, 100, 80, 20);
    brow.setStroke(Color.BLACK);
    root.getChildren().add(brow);
}

private void animation(Group root) {
    int cycleCount = 2;
    int time = 2000;

    ScaleTransition scaleTransition = new ScaleTransition(Duration.millis(time),
root);
    scaleTransition.setToX(2);
    scaleTransition.setToY(2);
    scaleTransition.setAutoReverse(true);

    RotateTransition rotateTransition = new RotateTransition(Duration.millis(time),
root);
    rotateTransition.setByAngle(360f);
    rotateTransition.setCycleCount(cycleCount + 1);
    rotateTransition.setAutoReverse(true);

    TranslateTransition translateTransition = new
TranslateTransition(Duration.millis(time), root);
    translateTransition.setFromX(150);
    translateTransition.setToX(50);
    translateTransition.setCycleCount(cycleCount +
1); translateTransition.setAutoReverse(true);

    TranslateTransition translateTransition2 =
new TranslateTransition(Duration.millis(time), root);
    translateTransition2.setFromX(50);
    translateTransition2.setToX(150);
    translateTransition2.setCycleCount(cycleCount +
1); translateTransition2.setAutoReverse(true);

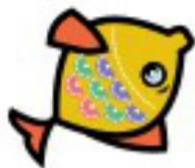
    ScaleTransition scaleTransition2 = new ScaleTransition(Duration.millis(time),
root);
    scaleTransition2.setToX(0.1);
    scaleTransition2.setToY(0.1);
    scaleTransition2.setCycleCount(cycleCount);
    scaleTransition2.setAutoReverse(true);

    ParallelTransition parallelTransition = new
ParallelTransition(); parallelTransition.getChildren().addAll(
        scaleTransition,
        scaleTransition2,
        translateTransition,
        translateTransition2,
        rotateTransition

```

```
        );  
        parallelTransition.setCycleCount(Timeline.INDEFINITE);  
        parallelTransition.play();  
    }  
  
}
```

Результати роботи програми



Висновки

Виконавши дану лабораторну роботу, я вивчив структуру та особливості використання файлів формату *.bmpbmp*, вивчив стандартні засоби *JavaFX* для візуалізації зображення, вивчив засоби анімації примітивів в *JavaFX*.