



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

Лабораторна робота № 1

з дисципліни “Математичні та алгоритмічні основи комп'ютерної графіки”

Виконав
студент III курсу
групи КП-83

Мричко Богдан
(*прізвище, ім'я, по батькові*)

Зарахована
“ ____ ” “ ____ ” 20 ____ р.
викладачем

Шкурат Оксаною Сергіївною
(*прізвище, ім'я, по батькові*)

варіант № 15

Варіант завдання

Завдання: Створити малюнок за варіантом користуючись графічними примітивами бібліотеки JavaFX.

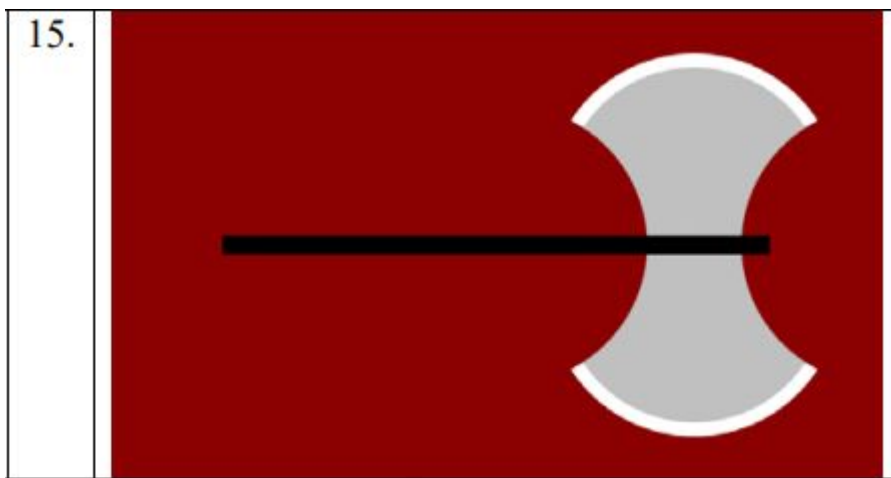
Короткі теоретичні відомості: JavaFX — платформа та набір інструментів для створення насичених застосунків з можливістю підвантаження медіа та змісту. JavaFX включає в себе набір утиліт, за допомогою яких розробники і дизайнери можуть швидко створювати та надавати розвинуті застосунки для десктопів, мобільних пристроїв, телебачення та інших платформ.

Центральним поняттям, впровадженим в JavaFX, є **Stage**. Stage визначають простір - контейнер для сцен. **Scene** - контейнером для елементів.

Окремі елементи сцени називаються **вузлами**. Наприклад, вузлом вважається елемент управління екранною кнопкою. Але самі вузли можуть складатися з груп інших вузлів. Більш того, у кожного вузла може бути нащадок, або породжений вузол, і тоді він називається **батьківським** вузлом, або вузлом розгалуження. А вузли без нащадків є кінцевими і називаються **листками**.

Сукупність усіх вузлів в сцені називається **графом**, що створює дерево.

Завдання за варіантом:



Код програми

Main.java

```
package sample;

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.*;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Lab1");
        Group root = new Group();
        Scene scene = new Scene(root, 400, 400);
        Rectangle plot = new Rectangle();
        Color PlotColor = Color.rgb(138, 0, 0);
        int PlotX = 50;
        int PlotY = 115;
        plot.setX(PlotX);
        plot.setY(PlotY);
        plot.setWidth(300);
        plot.setHeight(190);
        plot.setFill(PlotColor);

        Rectangle handle = new Rectangle();
        handle.setX(PlotX+45);
        handle.setY(PlotY+91);
        handle.setWidth(210);
        handle.setHeight(8);
        handle.setFill(Color.BLACK);

        Ellipse whitePart = new Ellipse();
        whitePart.setCenterX(PlotX+230);
        whitePart.setCenterY(PlotY+95);
        whitePart.setRadiusX(60);
        whitePart.setRadiusY(80);
        whitePart.setFill(Color.WHITE);

        Ellipse grayPart = new Ellipse();
        grayPart.setCenterX(PlotX+230);
        grayPart.setCenterY(PlotY+95);
        grayPart.setRadiusX(55);
        grayPart.setRadiusY(75);
        grayPart.setFill(Color.rgb(206, 206, 206));

        int RedCircleRadius = 50;

        Circle redCircleLeft = new Circle();
        redCircleLeft.setCenterX(PlotX+164);
        redCircleLeft.setCenterY(PlotY+95);
        redCircleLeft.setRadius(RedCircleRadius);
        redCircleLeft.setFill(PlotColor);

        Arc arc = new Arc();
        arc.setCenterX(PlotX+297);
        arc.setCenterY(PlotY+95);
        arc.setRadiusX(RedCircleRadius);
        arc.setRadiusY(RedCircleRadius);
```

```
        arc.setFill(PlotColor);
        arc.setStartAngle(90.0f);
        arc.setLength(180.0f);
        arc.setType(ArcType.ROUND);

        root.getChildren().add(plot);
        root.getChildren().add(whitePart);
        root.getChildren().add(grayPart);
        root.getChildren().add(redCircleLeft);
        root.getChildren().add(arc);
        root.getChildren().add(handle);

        primaryStage.setScene(scene);
        primaryStage.show();
    }

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

Результат

