```java
package vop;

//Her har jeg løst Opgave 1a for Ellipse.
public class Ellipse extends AbstractShape{
    double r1;
    double r2;

    double ellipsearea = Math.PI*r1*r2;
    double ellipsecircumference = 2*Math.PI*Math.sqrt
(0.5*(r1*r1+r2*r2));


    public Ellipse(double r1, double r2){
        this.r1=r1;
        this.r2=r2;
    }

    public double getArea(){
        return ellipsearea;
    }

    public double getCircumference(){
        return ellipsecircumference;
    }
}

```

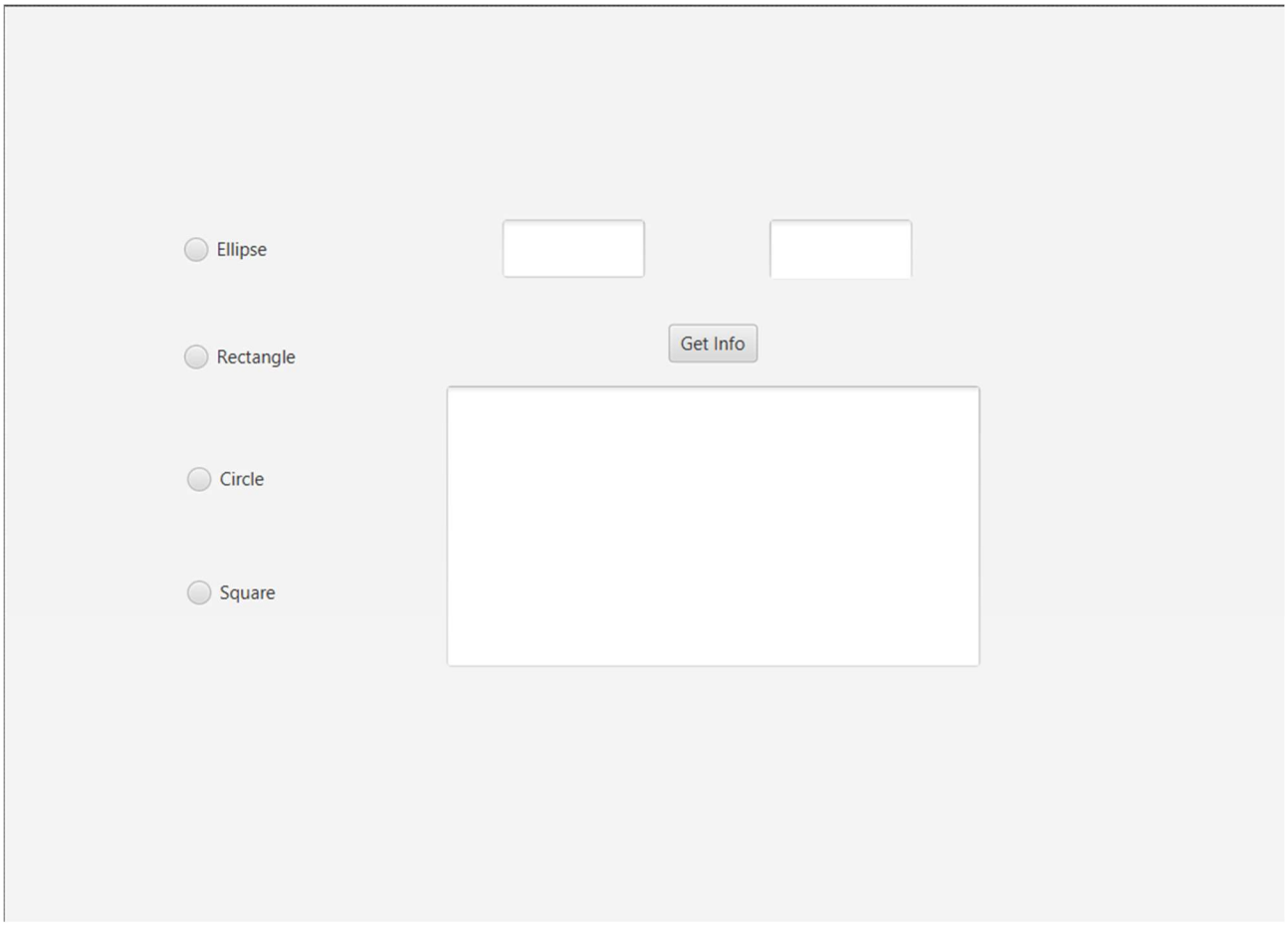
```java
package vop;

//Her har jeg løst Opgave 1a for Rectangle.
public class Rectangle extends AbstractShape{
    double l1;
    double l2;
    double rectanglearea = l1*l2;
    double rectanglecircumference = 2*(l1+l2);

    public Rectangle(double v, double v1) {
        super();
    }

    public double getRectanglearea(){
        return rectanglearea;
    }

    public double getRectanglecircumference(){
        return rectanglecircumference;
    }

    @Override
    public double getArea() {
        return 0;
    }

    @Override
    public double getCircumference() {
        return 0;
    }
}

```

```java
package vop;

//Her har jeg løst Opgave 1b for Circle.
public class Circle extends Ellipse{

    public Circle(double radius){
        super(radius,radius);
    }
}

```

```java
package vop;

//Her har jeg løst Opgave 1b for Square.
public class Square extends Ellipse{
    public Square(double radius){
        super(radius,radius);
    }
}

```

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package vop;

import javafx.scene.control.MenuItem;
import javafx.scene.shape.Shape;

import static vop.ShapeFacade.SHAPES.CIRCLE;

/**
 *
 * @author erso
 */

//Her har jeg løst Opgave 1c for ShapeFacade.
public class ShapeFacade {

    //Singleton Stufff:
    private static ShapeFacade instance = null;

    public static ShapeFacade getInstance() {
        if (instance == null) {
            instance = new ShapeFacade();
        }
        return instance;
    }

    private ShapeFacade() {
    }
    //---------------------------

    public enum SHAPES{CIRCLE, ELLIPSE, RECTANGLE, SQUARE}

    // Facadens public metoder

```

```java
    //Her har jeg løst Opgave 1d med getShapeInfo-Metoden.
    public String getShapeInfo(SHAPES shape, double ... parametre) {
        ShapeInterface currentShape;
        switch (shape) {
            case CIRCLE -> currentShape = new Circle(parametre[0]);
            case ELLIPSE -> currentShape = new Ellipse(parametre[0], parametre[1]);
            case SQUARE -> currentShape = new Square(parametre[0]);
            case RECTANGLE -> currentShape = new Rectangle(parametre[0], parametre[1]);
            default -> {
                return "Unknown Shape";
            }
        }
        return currentShape.toString();
    }


}

```




Ovenpå, kan det ses at jeg har lavet de forskellige komponenter for JavaFX i SceneBuilder. I kan ignorere størrelsen på scenen, det vigtigste er bare at indsætte de rigtige knapper, figurer og labeler med tekst og id.

```java
package vop;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.control.*;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.VBox;

import javax.swing.*;

//Her har jeg bare indsat de nødvendige actionevents og implementeret de rigtige metoder getInfo-Knappen.
//Hvorimod ved Radiobutton_Handler, er det bare gjort  således at parameter-textarea dukker op på scene builder når man trykker på et specifik shape som kræver to parameter.
public class PrimaryController implements Initializable {

    @FXML
    private ToggleGroup ShapeToggle;

    @FXML
    private RadioButton circleradio;

    @FXML
    private RadioButton ellipseradio;

    @FXML
    private Button getinfo;

    @FXML
    private TextArea parameter1;

    @FXML
```

```
37      private TextArea parameter2;
38
39      @FXML
40      private RadioButton rectangleradio;
41
42      @FXML
43      private TextArea result;
44
45      @FXML
46      private RadioButton squareradio;
47
48
49
50      @FXML
51      void getinfo_method(ActionEvent event) {
52          String result;
53          ShapeFacade.SHAPES shape = (ShapeFacade.
    SHAPES) ShapeToggle.getSelectedToggle().getUserData
    ();
54          double p1 = Double.parseDouble(parameter1.
    getText());
55          if(shape == ShapeFacade.SHAPES.CIRCLE ||
    shape == ShapeFacade.SHAPES.SQUARE){
56              result = ShapeFacade.getInstance().
    getShapeInfo(shape,p1);
57          } else{
58              double p2 = Double.parseDouble(parameter2
    .getText());
59              result = ShapeFacade.getInstance().
    getShapeInfo(shape,p1,p2);
60          }
61          System.out.println((result + "\n"));
62          this.result.setText("\n " + result);
63
64      }
65
66      @FXML
67      void parameter1_method(MouseEvent event) {
68
69      }
70
```

```
71      @FXML
72      void parameter2_method(MouseEvent event) {
73
74      }
75
76      @FXML
77      void radiobutton_handler(ActionEvent event){
78          ShapeFacade.SHAPES shape = (ShapeFacade.
    SHAPES) ShapeToggle.getSelectedToggle().getUserData
    ();
79          parameter2.setVisible(!(shape == ShapeFacade
    .SHAPES.CIRCLE || shape == ShapeFacade.SHAPES.SQUARE
    ));
80      }
81
82
83      @FXML
84      void result_method(MouseEvent event) {
85
86      }
87
88
89      @Override
90      public void initialize(URL url, ResourceBundle
    resourceBundle) {
91          ellipseradio.setUserData(ShapeFacade.SHAPES.
    ELLIPSE);
92          rectangleradio.setUserData(ShapeFacade.
    SHAPES.RECTANGLE);
93          circleradio.setUserData(ShapeFacade.SHAPES.
    CIRCLE);
94          squareradio.setUserData(ShapeFacade.SHAPES.
    SQUARE);
95      }
96
97  }
98
```

```java
package vop;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.io.IOException;

/**
 * JavaFX App
 */

//Det er her, hvor vores Main-Metode er og hvor vi kører programmet fra.
public class App extends Application {

    private static Scene scene;

    @Override
    public void start(Stage stage) throws IOException {
        scene = new Scene(loadFXML("primary"), 640, 480);
        stage.setScene(scene);
        stage.show();
    }

    static void setRoot(String fxml) throws IOException {
        scene.setRoot(loadFXML(fxml));
    }

    private static Parent loadFXML(String fxml) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(App.class.getResource(fxml + ".fxml"));
        return fxmlLoader.load();
    }

```

```
36      public static void main(String[] args) {
37          launch();
38      }
39
40  }
```