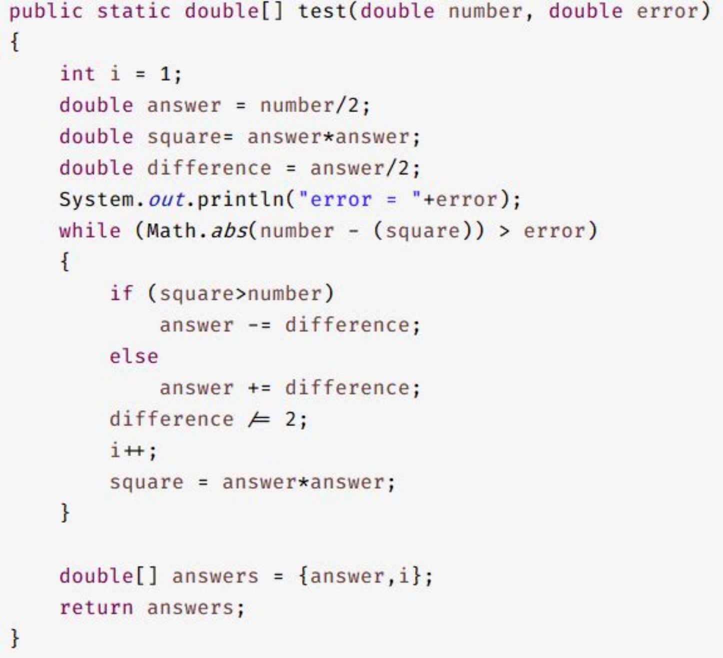
Square root calculator

Algoritmul

In prima iteratie raspunsul estimat este numarul/2.

Iterativ se va adauga sau scadea o valoare care se injumatateste la fiecare Iteratie.

Odata ce eroarea < epsilon se opreste loop-ul.

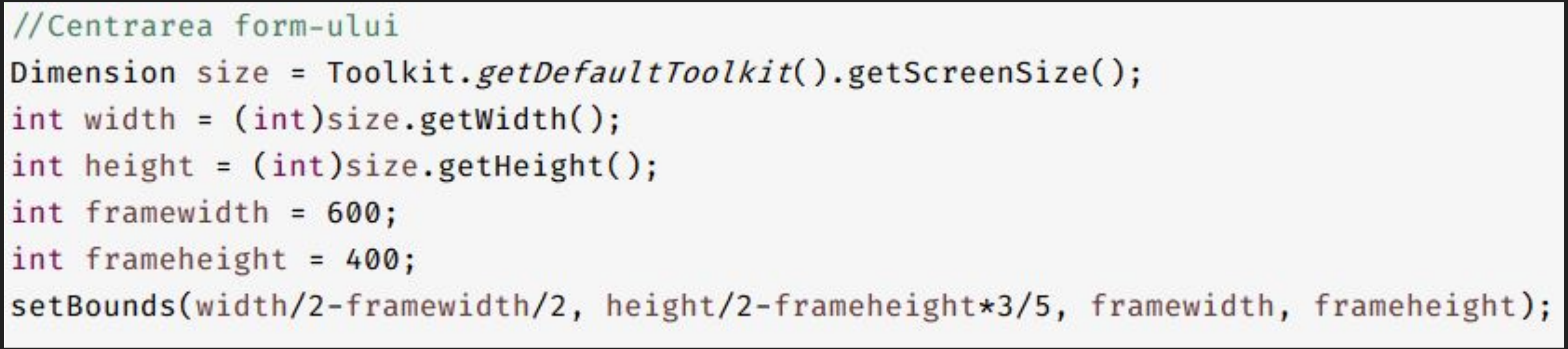


```
public static double[] test(double number, double error)
{
    int i = 1;
    double answer = number/2;
    double square= answer*answer;
    double difference = answer/2;
    System.out.println("error = "+error);
    while (Math.abs(number - (square)) > error)
    {
        if (square>number)
            answer -= difference;
        else
            answer += difference;
        difference /= 2;
        i++;
        square = answer*answer;
    }

    double[] answers = {answer,i};
    return answers;
}
```

Centrarea aplicatiei la pornire

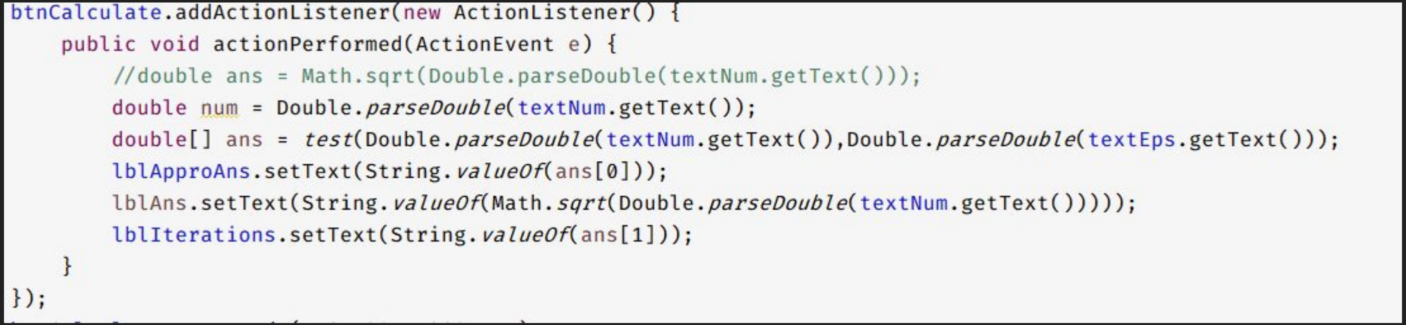
Aplicatia este centrata prin aplicarea unor calcule asupra dimensiunilor ecranului.



```
//Centrarea form-ului
Dimension size = Toolkit.getDefaultToolkit().getScreenSize();
int width = (int)size.getWidth();
int height = (int)size.getHeight();
int framewidth = 600;
int frameheight = 400;
setBounds(width/2-framewidth/2, height/2-frameheight*3/5, framewidth, frameheight);
```

Event buton

La apasarea butonului aplicatia citeste valorile din textbox-uri si le introduce in functie



```
btnCalculate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //double ans = Math.sqrt(Double.parseDouble(textNum.getText()));
        double num = Double.parseDouble(textNum.getText());
        double[] ans = test(Double.parseDouble(textNum.getText()),Double.parseDouble(textEps.getText()));
        lblApproAns.setText(String.valueOf(ans[0]));
        lblAns.setText(String.valueOf(Math.sqrt(Double.parseDouble(textNum.getText()))));
        lblIterations.setText(String.valueOf(ans[1]));
    }
});
```