

Python & Java {4} Teachers



GUI-laskin

Level 3 - Java



Funded by the
Erasmus+ Programme
of the European Union

{4}

Python & Java 4 Teachers

JAVA

{4}

Python & Java 4 Teachers

Java on korkeatasoinen, luokkapohjainen, oliopohjainen ohjelmointikieli, joka on suunniteltu niin, että siinä on mahdollisimman vähän riippuvuuksia toteutuksesta eri laitealustoille

Java on yleiskäyttöinen ohjelmointikieli, jonka tarkoituksena on antaa ohjelmoijille mahdollisuus kirjoittaa kerran ja suorittaa missä tahansa (write once, run anywhere - WORA), mikä tarkoittaa, että käännetty Java-koodi voi toimia kaikilla Javaa tukevilla alustoilla ilman, että sitä tarvitsee kääntää uudelleen.

Javan kehitti alun perin Sun Microsystemsin James Gosling (Oracle on sittemmin ostanut Sun Microsystemsin).

The Minecraft logo is displayed in a white rectangular box. It features the word "MINECRAFT" in a bold, blocky, 3D-style font with a dark grey shadow effect.

Tehtävä

Matematiikan luokan laskimet ovat loppuneet!

Luo Javaa käyttäen graafiseen käyttöliittymään perustuva laskin, jota luokkasi voi käyttää odottaessaan uusia laskimia.



Funded by the
Erasmus+ Programme
of the European Union

{4}

Python & Java 4 Teachers



Prosessi

Tehtävässä luodaan GUI-laskuri käyttäen Java-ohjelmaa.

- Annetaan käyttäjän kirjoittaa 2 numeroa.
- Annetaan käyttäjän valita matemaattinen perusoperaatio.
- Suoritetaan operaatio ja näytetään tulos käyttäjälle.

Laajennus

- Suunnittele ja tee oma ulkoasu laskinta varten.

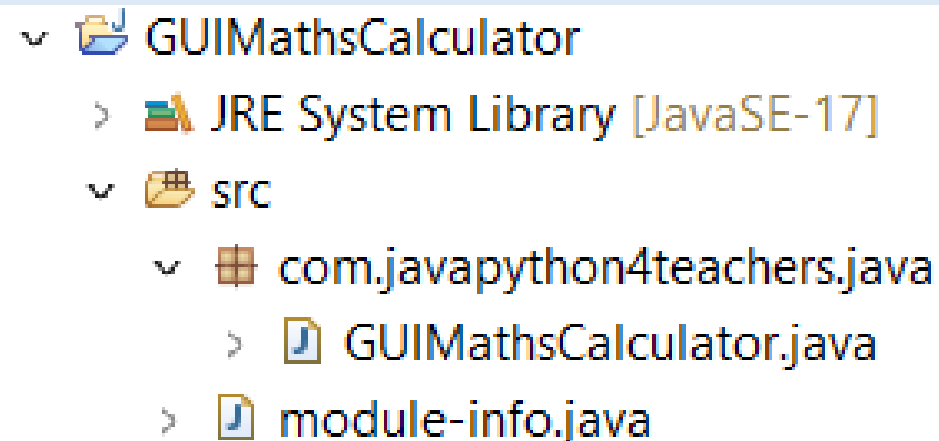




Ennen aloittamista...

Aloita uusi Java-projekti (new Java project).
Jos et ole täysin varma, miten se tehdään, lue
JohdatusJavaan.pdf-tiedosto.

Varmista, että IDEsi tiedostojärjestelmä näyttää
seuraavalta:



Virheistä

Java on aavistuksen monimutkaisempi kuin Python-kieli. Kuitenkin, monimutkaisuus tuo lisää mahdollisuuksia sekä erilaisia virheilmoituksia.

Siirrä hiiri punaisella alleviivatun koodinosan päälle, niin Eclipse IDE näyttää erilaisia ratkaisuvaihtoehtoja.

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Polygon;

import javax.swing.JFrame;
import javax.swing.SwingUtilities;
```

```
public class RoadSi

    public void pai
        super.paint

    Polygon b
    for (int i
        b.addPoin
            (int)

    g.setColor(
    g.fillPolyg
    g.drawPolyg
```

The type javax.swing.SwingUtilities is not accessible

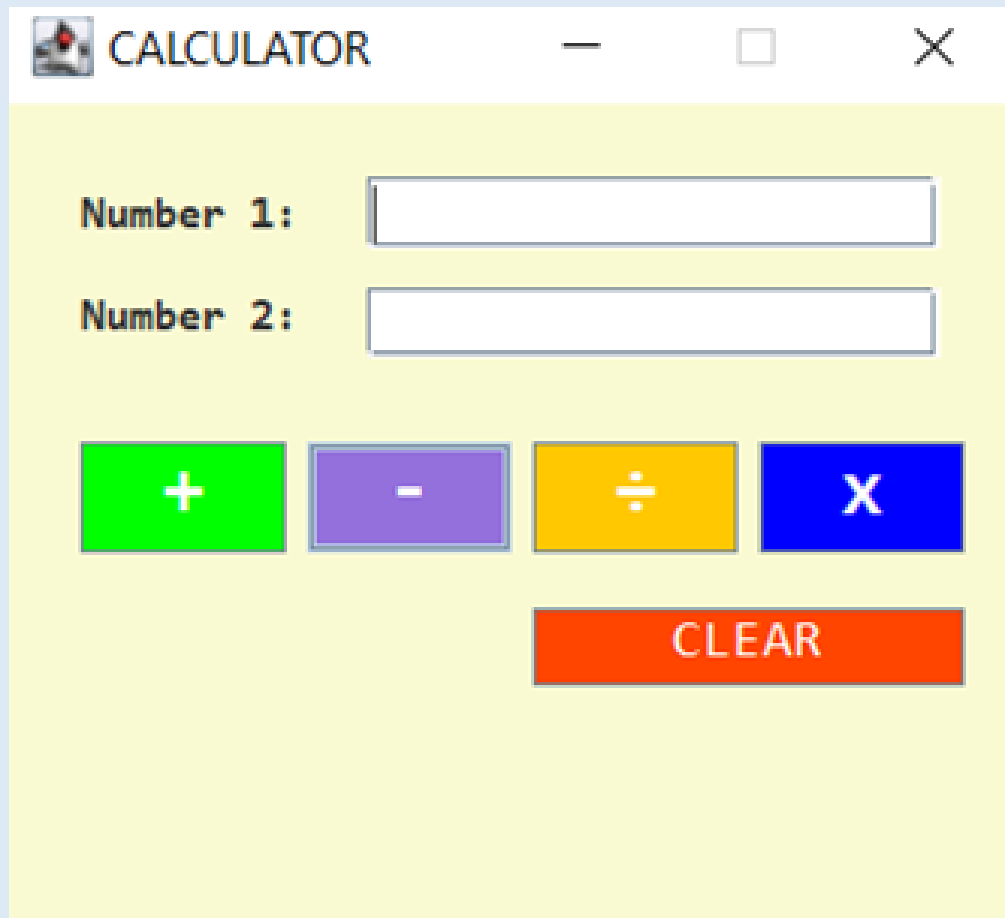
7 quick fixes available:

- [Create class 'SwingUtilities' in package 'javax.swing'](#)
- [Create record 'SwingUtilities' in package 'javax.swing'](#)
- [Create interface 'SwingUtilities' in package 'javax.swing'](#)
- [Create annotation 'SwingUtilities' in package 'javax.swing'](#)
- [Create enum 'SwingUtilities' in package 'javax.swing'](#)
- [Add 'requires java.desktop' to module-info.java](#)
- [Fix project setup...](#)

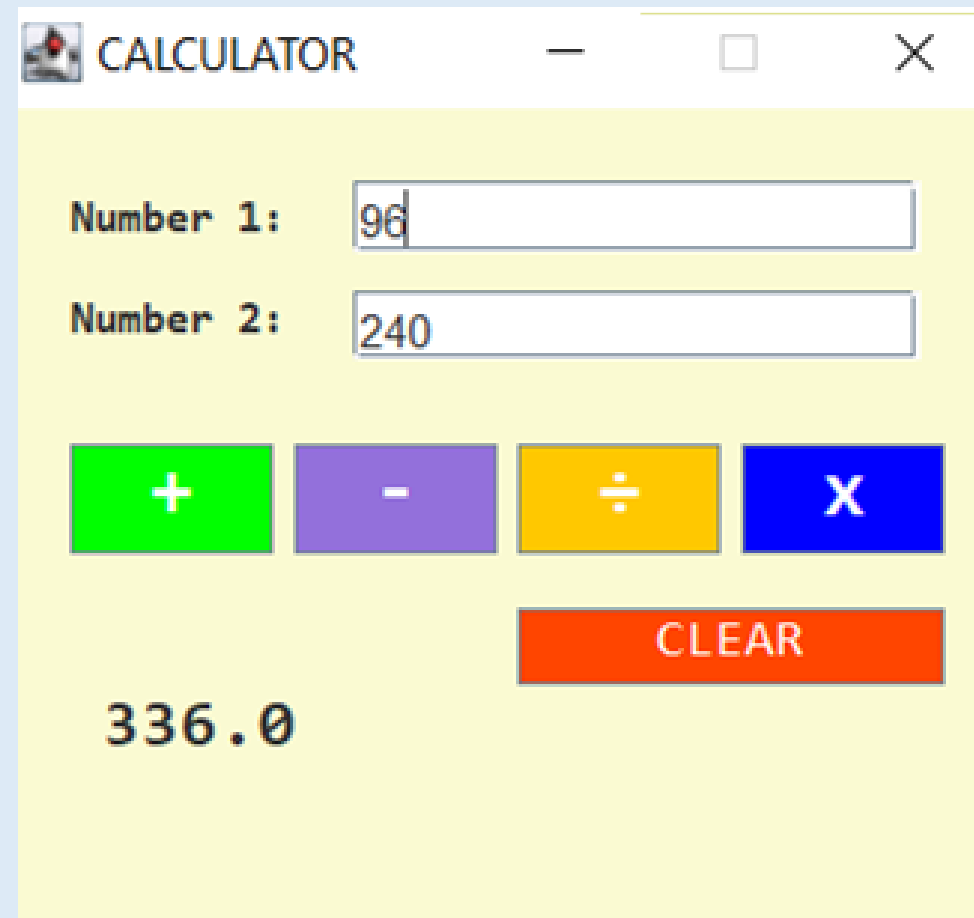
Esimerkiksi, jos saat virheilmoituksen 'this package is not accessible' (tätä pakettia ei löydy), valitse jokin näistä:



Miltä se näyttää...



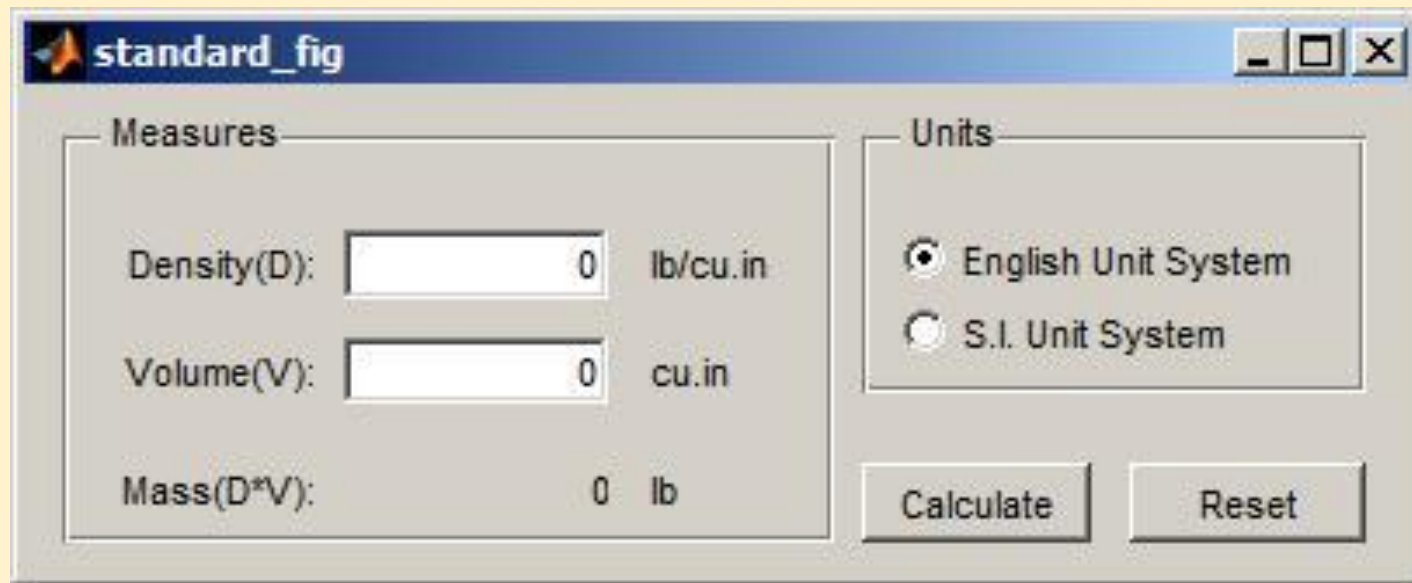
A screenshot of a simple calculator application window titled "CALCULATOR". The window has a yellow background and a white title bar with standard window controls. It features two input fields labeled "Number 1:" and "Number 2:". Below these fields are four buttons for arithmetic operations: a green "+" button, a purple "-" button, an orange "÷" button, and a blue "X" button. At the bottom center is a red "CLEAR" button.



A screenshot of the same calculator application window, but with values entered. "Number 1:" contains "96" and "Number 2:" contains "240". The result "336.0" is displayed at the bottom of the window. The layout and buttons are identical to the first screenshot.

Graafinen käyttöliittymä (GUI)

GUI (eli Graphical User Interface) on käyttöliittymä, jossa on kuvia, ikoneita ja nappeja sekä tulosruutuja.



1. vaihe

Tee GUIlle ikkuna

Kun olet IDEssä saanut luokan luotua, kirjoita alla oleva koodi GUIDemoon.

```
public class GUIDemo {  
  
    private JFrame windowCalculator;  
  
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                try {  
                    GUIDemo window = new GUIDemo();  
                    window.windowCalculator.setVisible(true);  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
    }  
  
    public GUIDemo() {  
        initialize();  
    }  
  
    private void initialize() {  
        windowCalculator = new JFrame();  
        windowCalculator.setResizable(false);  
        windowCalculator.setTitle("CALCULATOR");  
        windowCalculator.setBounds(100, 100, 285, 258);  
        windowCalculator.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        windowCalculator.getContentPane().setLayout(null);  
    }  
}
```

Käytetään JFrame-olio nimeltä 'windowCalculator', josta tehdään laskukoneemme.

Pääohjelma main() käynnistyy ensimmäiseksi.

Aliohjelma initialize() on funktio, jossa itse laskukone tehdään.



Funded by the
Erasmus+ Programme
of the European Union

{4}

Python & Java 4 Teachers

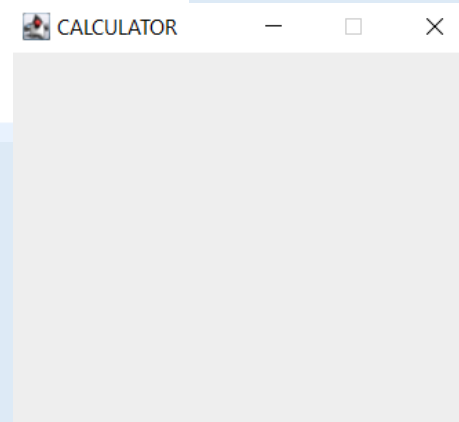


2. vaihe

Ikkunat ja initialize()

```
31
32 private void initialize() {
33     windowCalculator = new JFrame();
34     windowCalculator.setResizable(false);
35     windowCalculator.setTitle("CALCULATOR");
36     windowCalculator.setBounds(100, 100, 285, 258);
37     windowCalculator.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
38     windowCalculator.getContentPane().setLayout(null);
39 }
40
41 }
```

Tämä function säätää laskimen ikkunaa, nyt laitetaan koko ja otsikko kuntoon. Myöhemmin lisätään nappeja ja toiminnallisuutta.



Funded by the
Erasmus+ Programme
of the European Union

{4}

Python & Java 4 Teachers



3. vaihe

Lisätään osia ikkunaan

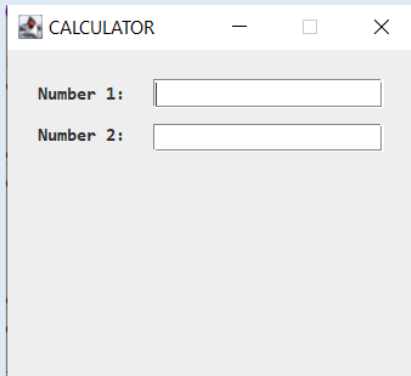
```
13 private JFrame windowCalculator;  
14 private JTextField textField1;  
15 private JTextField textField2;  
16 private JButton addition;  
17 private JButton subtraction;  
18 private JButton division;  
19 private JButton multiplication;  
20 private JButton clear;
```

Lisätään peruslaskimen vaatimat osaset koodiin.

Samalla IDEen tulee virheilmoituksia, mutta korjataan ne myöhemmin.

Lisää vielä viereinen koodin pätkä initialize()-funktioon.

Laskin näyttää tällaiselta.



```
46  
47 textField1 = new JTextField();  
48 textField1.setBounds(96, 20, 152, 19);  
49 windowCalculator.getContentPane().add(textField1);  
50 textField1.setColumns(10);  
51  
52 textField2 = new JTextField();  
53 textField2.setColumns(10);  
54 textField2.setBounds(96, 49, 152, 19);  
55 windowCalculator.getContentPane().add(textField2);  
56  
57 JLabel label1 = new JLabel("Number 1:");  
58 label1.setFont(new Font("Consolas", Font.BOLD, 12));  
59 label1.setBounds(20, 25, 76, 13);  
60 windowCalculator.getContentPane().add(label1);  
61  
62 JLabel label2 = new JLabel("Number 2:");  
63 label2.setFont(new Font("Consolas", Font.BOLD, 12));  
64 label2.setBounds(20, 52, 76, 13);  
65 windowCalculator.getContentPane().add(label2);  
66
```



4. vaihe

Lisää osia

Kirjoita allaolevat initialize()-funktioon. Lopputuloksena on napit peruslaskutoimituksille.

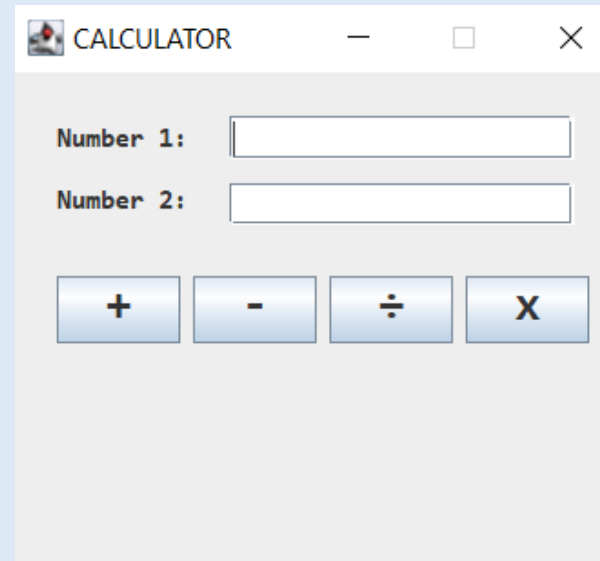
```
JLabel result = new JLabel("");
result.setFont(new Font("Consolas", Font.BOLD, 18));
result.setBounds(29, 126, 219, 85);
windowCalculator.getContentPane().add(result);

addition = new JButton("+");
addition.setFont(new Font("Consolas", Font.BOLD, 21));
addition.setBounds(20, 90, 55, 30);
windowCalculator.getContentPane().add(addition);

subtraction = new JButton("-");
subtraction.setFont(new Font("Consolas", Font.BOLD, 21));
subtraction.setBounds(80, 90, 55, 30);
windowCalculator.getContentPane().add(subtraction);

division = new JButton("\u00F7");
division.setFont(new Font("Consolas", Font.BOLD, 21));
division.setBounds(140, 90, 55, 30);
windowCalculator.getContentPane().add(division);

multiplication = new JButton("x");
multiplication.setFont(new Font("Consolas", Font.BOLD, 21));
multiplication.setBounds(200, 90, 55, 30);
windowCalculator.getContentPane().add(multiplication);
```



Action Listeners

Action Listeners ovat metodeja, jotka kuuntelevat Java-ohjelmaa. Esimerkiksi, kun hiirellä klikataan nappia, Action Listener huomaa sen ja tekee jotain.

1. Luokka implementoi eli käynnistää kuuntelijan, tässä se on ActionListener

2. Liitä kuuntelija osaan (tässä nappiin), jota halutaan kuunnella

3. Implementoi kuuntelijalle funktio, eli mitä tehdään kun kuuntelija herää

4. Tee jotain

```
Class OurProg implements ActionListener {  
    ...  
    public void aMethod() {  
        JButton btn = new JButton("btn");  
        btn.addActionListener(this);  
        ...  
    }  
    public void actionPerformed(ActionEvent e) {  
        btn.setText("You clicked me!");  
        ...  
    }  
}
```

Kun nappia klikataan hiirellä, ActionEvent e tietää sen.

5. vaihe

Lisätään toiminnat nappiin

Jokainen nappi tarvitsee oman kuuntelijan, ActionListener:n, joka lukee numerot tekstikentistä, tekee vaadittavan operation ja näyttää vastauksen käyttäjälle

```
addition = new JButton("+");

addition.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        float sum = Float.parseFloat(textField1.getText()) + Float.parseFloat(textField2.getText());
        result.setText( Float.toString(sum) );
    }
});

addition.setFont(new Font("Consolas", Font.BOLD, 21));
addition.setBounds(20, 90, 55, 30);
windowCalculator.getContentPane().add(addition);
```

Muuttuja sum sisältää käyttäjän syöttämien lukujen kentistä textField1 ja textField2 summan. Kenttien luvut muutetaan liukuluvuiksi (floating number), jotta laskin osaisi laskea myös desimaalilukuja.

Haaste: Tee kuuntelijat (ActionListener) muille napeille.



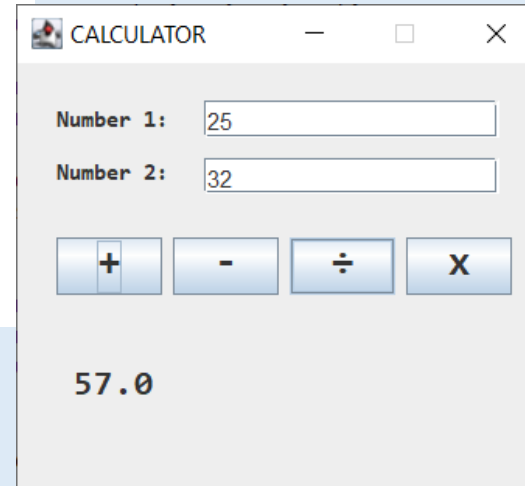
Vähän tähän tapaan...

```
subtraction = new JButton("-");
subtraction.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        float sum = Float.parseFloat(textField1.getText()) - Float.parseFloat(textField2.getText());
        result.setText( Float.toString(sum) );
    }
});
subtraction.setFont(new Font("Consolas", Font.BOLD, 21));
subtraction.setBounds(80, 90, 55, 30);
windowCalculator.getContentPane().add(subtraction);

division = new JButton("\u00F7");
division.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        float sum = Float.parseFloat(textField1.getText()) / Float.parseFloat(textField2.getText());
        result.setText( Float.toString(sum) );
    }
});
division.setFont(new Font("Consolas", Font.BOLD, 21));
division.setBounds(140, 90, 55, 30);
windowCalculator.getContentPane().add(division);

multiplication = new JButton("x");
multiplication.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        float sum = Float.parseFloat(textField1.getText()) * Float.parseFloat(textField2.getText());
        result.setText( Float.toString(sum) );
    }
});
multiplication.setFont(new Font("Consolas", Font.BOLD, 21));
multiplication.setBounds(200, 90, 55, 30);
windowCalculator.getContentPane().add(multiplication);
```

Haaste: Tee CLEAR eli tyhjennysnappi, jolla voit siivota tekstikenttien numerot.

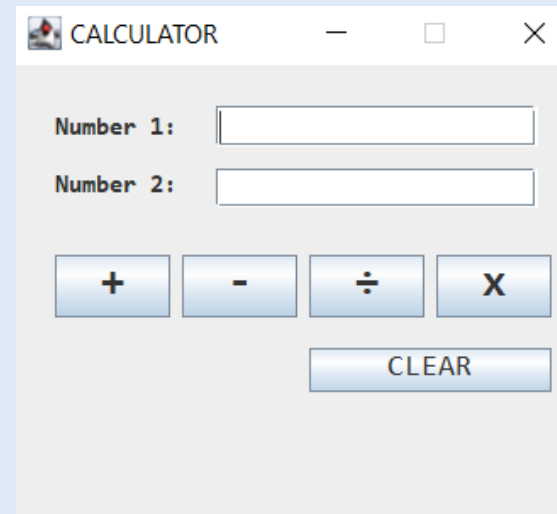


6. vaihe

Lisää tyhjennysnappi

Lisätään tyhjennysnappi CLEAR ja sille kuuntelija (ActionListener), joka muuttaa tekstikenttien arvoksi NULL, eli tyhjä.

```
117
118     clear = new JButton("CLEAR");
119     clear.addActionListener(new ActionListener() {
120         public void actionPerformed(ActionEvent e) {
121             result.setText(null);
122             textField1.setText(null);
123             textField2.setText(null);
124         }
125     });
126     clear.setFont(new Font("Consolas", Font.PLAIN, 15));
127     clear.setBounds(140, 134, 115, 21);
128     windowCalculator.getContentPane().add(clear);
129
130 }
```

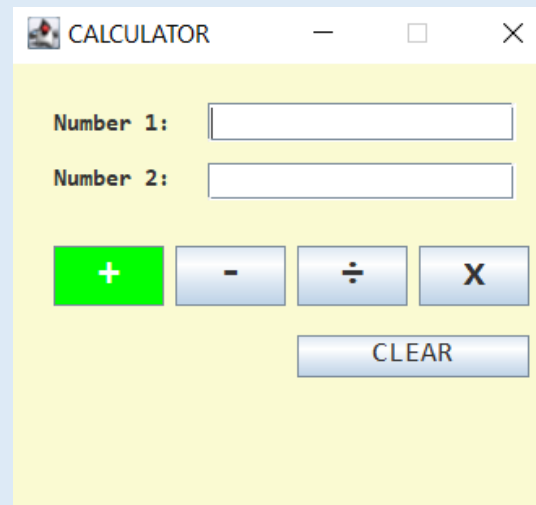


Laajennos

Lisää värit nappeihin. Tee laskukoneestasi vähän fiinimpi.

```
addition = new JButton("+");  
addition.setForeground(new Color(255, 255, 255));  
addition.setBackground(Color.GREEN);
```

```
windowCalculator.setTitle("CALCULATOR");  
windowCalculator.setBounds(100, 100, 285, 258);  
windowCalculator.getContentPane().setBackground(new Color(250, 250, 210));
```



Ole luova ja kekseliäs! Tee erimuotoista nappeja, muuta värejä ja fontteja. Siirrä eri paikkaan.



Lopuksi...

Osaat

- ✓ tehdä Java-projektin
- ✓ tehdä GUI:n ikkunan
- ✓ kuunnella käyttäjää GUIlla
- ✓ tehdä toimivia nappeja
- ✓ tulostaa tietoa GUIhin



Arkielämässä...

Töissä

Laskukoneita on joka paikassa, ei ainoastaan matikantunneilla, vaan myös kännykässäsi ja tietokoneessasi.

GUIhin perustuvat ohjelmat ovat kaikkialla! GUI lisää saavutettavuutta.



Onnittelut!

Olet tehnyt GUI-pohjaisen laskimen Javalla!



Funded by the
Erasmus+ Programme
of the European Union

{4}

Python & Java 4 Teachers