# Graphical User Interface

Zbigniew Dendzik
Institute of Physics, University of Silesia

2025

## Introduction

An integral part of the Java platform is the swing library, with which you can create applications with an advanced graphical interface. Below you will find instructions for exercises that aim to familiarise you with the technique of creating a graphical interface and programming event handling related to the graphical interface.

## Example I

Below you will find an example demonstrating the creation of a simple graphical interface and handling events related to the elements of this interface.

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class Demo implements ActionListener
{
   JTextField t1;
   JButton b1;
   JButton b2;

   public void actionPerformed(ActionEvent e)
   {
      Object target = e.getSource();

      if(target==b1||target==t1)
      {
         int k=Integer.parseInt(t1.getText());
         t1.setText(Integer.toString(k*k));
         t1.requestFocus();
      }
      else if (target==b2)
      {
         t1.setText("");
         t1.requestFocus();
      }
   }
```

```
27
28      void init()
29      {
30          t1=new JTextField(6);
31          b1=new JButton("^2");
32          b2=new JButton("clear");
33
34          JPanel p=new JPanel();
35          p.add(t1);
36          p.add(b1);
37          p.add(b2);
38
39          t1.addActionListener(this);
40          b1.addActionListener(this);
41          b2.addActionListener(this);
42
43          JFrame f=new JFrame();
44          Container c=f.getContentPane();
45          c.add(p);
46          f.pack();
47          f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
48          f.setVisible(true);
49      }
50
51      public static void main(String[] args)
52      {
53          //up to version 1.4
54          //new Demo().init();
55
56          //from version 1.5
57          SwingUtilities.invokeLater(new Runnable()
58          {
59              public void run()
60              {
61                  new Demo().init();
62              }
63          });
64      }
65  }
```

Listing 1: Demo.java

## Ex. 7.1

Add appropriate fields and buttons “+” and “=” and implement the operation of adding two numbers. Add handling of exceptional situations, especially the `NumberFormatException` exception.

# Example II

Below you will find the source code of the skeleton of a simple calculator implementation. Analyse the methods of creating graphical interface elements, arranging these elements relative to each other (layout manager) and programming event handling related to the graphical interface.

```
1  import javax.swing.*;
2  import java.awt.event.*;
3  import java.awt.*;
4
5  public class Calc implements ActionListener
6  {
7     JTextField t1;
8     JButton b1;
9     JButton bplus,beq;
10
11    double x,buf;
12
13    public void actionPerformed(ActionEvent e)
14    {
15       Object target = e.getSource();
16
17       if(target==b1)
18       {
19          t1.setText(t1.getText()+((JButton)target).getText());
20          t1.requestFocus();
21       }
22
23       else if(target==bplus)
24       {
25          buf=Double.parseDouble(t1.getText());
26          t1.setText("");
27          t1.requestFocus();
28       }
29
30       else if(target==beq||target==t1)
31       {
32          x=Double.parseDouble(t1.getText());
33          x=buf+x;
34          t1.setText(Double.toString(x));
35          t1.requestFocus();
36       }
37    }
38
39    void init()
40    {
41       //try
42       //{
43       //UIManager.setLookAndFeel(UIManager.
            getSystemLookAndFeelClassName());
44       //}
```

```
45          //catch(Exception e){}
46
47          JFrame f=new JFrame();
48          Container c=f.getContentPane();
49
50          GridBagLayout gbl=new GridBagLayout();
51          GridBagConstraints gbc=new GridBagConstraints();
52          gbc.fill=GridBagConstraints.HORIZONTAL;
53          c.setLayout(gbl);
54
55
56
57          t1=new JTextField(15);
58          t1.addActionListener(this);
59          t1.setHorizontalAlignment(JTextField.RIGHT);
60          gbc.gridx=0;
61          gbc.gridy=0;
62          gbc.gridwidth=5;
63          gbc.ipadx=0;
64          gbc.ipady=5;
65          gbc.insets=new Insets(5,5,0,5);
66          gbl.setConstraints(t1,gbc);
67          c.add(t1);
68
69
70
71          b1=new JButton("1");
72          b1.addActionListener(this);
73          b1.setFocusable(false);
74          gbc.gridx=0;
75          gbc.gridy=1;
76          gbc.gridwidth=1;
77          gbc.ipadx=0;
78          gbc.ipady=0;
79          gbc.insets=new Insets(5,5,0,0);
80          gbl.setConstraints(b1,gbc);
81          c.add(b1);
82
83
84
85          bplus=new JButton("+");
86          bplus.addActionListener(this);
87          bplus.setFocusable(false);
88          bplus.setToolTipText("addition");
89          gbc.gridx=3;
90          gbc.gridy=1;
91          gbc.gridwidth=2;
92          gbc.ipadx=30;
93          gbc.ipady=0;
94          gbc.insets=new Insets(5,5,0,5);
95          gbl.setConstraints(bplus,gbc);
```

```
96          c.add(bplus);
97
98
99
100         beq=new JButton("=");
101         beq.addActionListener(this);
102         beq.setFocusable(false);
103         beq.setToolTipText("execute operation");
104         gbc.gridx=0;
105         gbc.gridy=5;
106         gbc.gridwidth=4;
107         gbc.ipadx=30;
108         gbc.ipady=0;
109         gbc.insets=new Insets(5,5,5,0);
110         gbl.setConstraints(beq,gbc);
111         c.add(beq);
112
113
114
115         f.pack();
116         f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
117         f.setTitle("Calc");
118         f.setVisible(true);
119     }
120
121     public static void main(String[] args)
122     {
123         //up to version 1.4
124         //new Calc().init();
125
126         //from version 1.5
127             SwingUtilities.invokeLater(new Runnable()
128         {
129             public void run()
130             {
131                 new Calc().init();
132             }
133         });
134     }
135 }
```

Listing 2: Calc.java

## Ex. 7.2

Prepare an appropriate compilation unit, then compile and test the above example. Then extend it by adding the remaining buttons with digits and a decimal point and write appropriate event handling functions (remember that at a given moment two decimal points cannot appear on the display).

### Ex. 7.3

Add buttons representing basic mathematical operations, as well as exponentiation, square root, percentages and memory. Write appropriate button handling functions and include handling of situations such as division by zero or square root of a negative number. Basic mathematical functions are implemented in the library as static methods of the `java.lang.Math` class. Appropriate methods of the `java.lang.Double` wrapper class are used for conversion between floating-point numbers (`double`) and text (`String`).

### Ex. 7.4

Refine your calculator so that arbitrary characters cannot be entered into the display from the keyboard, but only digits entered using the graphical interface buttons or numeric keys from the keyboard. To do this, set the text field state (display) to non-editable (method: `setEditable()` of the `JTextField` class). You can handle keyboard events using the `KeyListener` interface.

### Ex. 7.5

Add logging of all operations performed using the calculator and their results to a text file and implement undo operations in the calculator.

### Ex. 7.6

Write a graphical interface for the multithreaded internet messenger that was one of the exercises we did in the previous semester.

## Example III

Below you will find an example of serialising an object to a stream encrypted with the symmetric DES algorithm. The example consists of several stages – generating a key based on a password, writing an object (in this case text) to a file and reading an object from a file. We will use serialisation in the next exercise, which consists of writing an implementation of a diary, equipped with a graphical interface created based on `javax.swing`, which will save notes to a file encrypted with a password.

```
1  import java.io.*;
2  import java.security.*;
3  import javax.crypto.*;
4  import javax.crypto.spec.*;
5
6  public class DesTest
7  {
8     public static void main(String args[]) throws Exception
9     {
10       String msg="greetings from aunt";
11
12       String passwd="secretpassword";
13       byte[] b=passwd.getBytes();
14
```

```
15
16
17          //-- key generation --
18              System.out.println("creating key generator");
19          KeyGenerator kgen = KeyGenerator.getInstance("DES");
20          kgen.init(new java.security.SecureRandom(b));
21
22          System.out.println("generating key");
23          SecretKey key = kgen.generateKey();
24
25
26
27          //-- encryption --
28              System.out.println("creating Cipher object");
29          javax.crypto.Cipher c1 = javax.crypto.Cipher.getInstance("
                DES");
30
31          System.out.println("initialising cipher for encryption");
32          c1.init(Cipher.ENCRYPT_MODE,key);
33
34          ObjectOutputStream out;
35          out=new ObjectOutputStream(new CipherOutputStream(new
                FileOutputStream("file.txt"),c1));
36          out.writeObject(msg);
37          out.flush();
38          out.close();
39
40
41
42          //-- decryption --
43              System.out.println("creating Cipher object");
44          javax.crypto.Cipher c2 = javax.crypto.Cipher.getInstance("
                DES");
45
46          System.out.println("initialising cipher for decryption");
47          c2.init(Cipher.DECRYPT_MODE,key);
48
49          ObjectInputStream in;
50          in=new ObjectInputStream(new CipherInputStream(new
                FileInputStream("file.txt"),c2));
51          String str=(String)in.readObject();
52          in.close();
53
54          System.out.println("\n-- "+str+" --\n");
55      }
56  }
```

Listing 3: DesTest.java

## Ex. 7.7

Write an implementation of an electronic diary whose content is saved to a file in encrypted form using the DES algorithm. Start with the simplest diary that saves text to a file, and then extend it with such capabilities as automatic dating of entries, a system of bookmarks with successive entries, a full-text search engine and a system for adding photos to notes.