Bereich: Grafische Benutzeroberflächen (UI), Events (1)

Währungsumrechner (2)	**Musterlösung**
Package: `de.dhbwka.java.exercise.ui.event`	**Klasse:** `CurrencyCalculator`

```java
package de.dhbwka.java.exercise.ui.event;

import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JTextField;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016-2018 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.1
 */
public class CurrencyCalculator implements ActionListener {

   public final static float CHANGE_RATE = 1.09f;

   private JFrame frame = new JFrame( "Currency converter" );

   private JTextField input =
         new JTextField( "Please enter amount to convert!" );

   private JButton btnEur2usd = new JButton( "EUR -> USD" );
   private JButton btnUsd2eur = new JButton( "USD -> EUR" );
   private JButton btnCancel = new JButton( "Cancel" );

   public CurrencyCalculator() {
      // default BorderLayout has no margin!
      this.frame.setLayout( new BorderLayout( 10, 10 ) );
      this.frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
      this.frame.add( this.input, BorderLayout.NORTH );
      this.frame.add( this.btnEur2usd, BorderLayout.WEST );
      this.frame.add( this.btnUsd2eur, BorderLayout.CENTER );
      this.frame.add( this.btnCancel, BorderLayout.EAST );

      // Add ActionListener to buttons (2nd part)
      this.btnEur2usd.addActionListener( this );
      this.btnUsd2eur.addActionListener( this );
      this.btnCancel.addActionListener( this );

      this.frame.setSize( 350, 90 );
      this.frame.setVisible( true );
   }
```

```
   /**
    * Event handling (2nd part)
    */
   @Override
   public void actionPerformed( ActionEvent e ) {
      if ( e.getSource() == this.btnCancel ) {
         System.exit( 0 );
      }
      try {
         float amount = Float.parseFloat( this.input.getText() );
         if ( e.getSource() == this.btnEur2usd ) {
            amount = amount * CurrencyCalculator.CHANGE_RATE;
         }
         if ( e.getSource() == this.btnUsd2eur ) {
            amount = amount / CurrencyCalculator.CHANGE_RATE;
         }

         this.input.setText( Float.toString( amount ) );
      } catch ( Exception ex ) {
         this.input.setText( "Error parsing amount." );
      }
   }

   public static void main( String[] args ) {
      new CurrencyCalculator();
   }

}
```

Grafische Benutzeroberflächen (UI), Events (1)

Body-Mass-Index (BMI)	**Musterlösung**
Package: `de.dhbwka.java.exercise.ui.event`	**Klasse:** `BMICalculator`

```
package de.dhbwka.java.exercise.ui.event;

import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextField;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016-2018 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.1
 */
public class BMICalculator implements ActionListener {

   private JTextField weightField = new JTextField( 10 );
   private JTextField heightField = new JTextField( 10 );
   private JTextField bmiField = new JTextField( 15 );
   private JTextField messageField = new JTextField( 20 );
   private JRadioButton rbMale = new JRadioButton( "male", true );
   private JRadioButton rbFemale = new JRadioButton( "female", false );

   public BMICalculator() {
      JFrame frame = new JFrame( "BMI Calculator" );
      frame.setLayout( new GridLayout( 6, 1 ) );

      JPanel weightPanel = new JPanel();
      JPanel heightPanel = new JPanel();
      JPanel sexPanel = new JPanel();
      JPanel calcBtnPanel = new JPanel();
      JPanel bmiPanel = new JPanel();
      JPanel messagePanel = new JPanel();

      ButtonGroup radioButtonGroup = new ButtonGroup();
      radioButtonGroup.add( this.rbMale );
      radioButtonGroup.add( this.rbFemale );


      // Continued on next page
```

```
      weightPanel.add( new JLabel( "Weight [kg]:" ) );
      weightPanel.add( this.weightField );
      heightPanel.add( new JLabel( "Body height [m]" ) );
      heightPanel.add( this.heightField );

      sexPanel.add( this.rbMale );
      sexPanel.add( this.rbFemale );

      JButton calcBtn = new JButton( "Calculate" );
      calcBtn.addActionListener( this );
      calcBtnPanel.add( calcBtn );

      bmiPanel.add( new JLabel( "BMI:" ) );
      bmiPanel.add( this.bmiField );
      messagePanel.add( this.messageField );

      frame.add( weightPanel );
      frame.add( heightPanel );
      frame.add( sexPanel );
      frame.add( calcBtnPanel );
      frame.add( bmiPanel );
      frame.add( messagePanel );

      frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
      frame.pack();
      frame.setVisible( true );
   }

   @Override
   public void actionPerformed( ActionEvent e ) {
      try {
         int weight = Integer.parseInt( this.weightField.getText() );
         double height = Double.parseDouble( this.heightField.getText() );
         double bmi = weight / (height * height);

         this.bmiField.setText( Double.toString( bmi ) );
         this.messageField
               .setText( this.getBMIType( this.rbMale.isSelected(), bmi ) );
      } catch ( Exception ex ) {
         this.bmiField.setText( "Bad input" );
      }
   }


   // Continued on next page
```

```
   public String getBMIType( boolean male, double bmi ) {
      String erg;
      if ( male ) {
         if ( bmi < 20 ) {
            erg = "Short weight";
         } else if ( bmi < 25 ) {
            erg = "Normal weight";
         } else if ( bmi < 30 ) {
            erg = "Overweight";
         } else if ( bmi < 40 ) {
            erg = "Adiposity";
         } else {
            erg = "Massive Adiposity";
         }
      } else {
         if ( bmi < 19 ) {
            erg = "Short weight";
         } else if ( bmi < 24 ) {
            erg = "Normal weight";
         } else if ( bmi < 30 ) {
            erg = "Overweight";
         } else if ( bmi < 40 ) {
            erg = "Adiposity";
         } else {
            erg = "Massive Adiposity";
         }
      }
      return erg;
   }

   public static void main( String[] args ) {
      new BMICalculator();
   }

}
```

Grafische Benutzeroberflächen (UI), Events (1)

Binäre Zahlen	**Musterlösung**
Package: `de.dhbwka.java.exercise.ui.event`	**Klasse:** `BinaryNumber`

```java
package de.dhbwka.java.exercise.ui.event;

import java.awt.BorderLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JToggleButton;
import javax.swing.SwingConstants;
import javax.swing.WindowConstants;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2018 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.0
 */
@SuppressWarnings( "serial" )
public class BinaryNumber extends JFrame implements ActionListener {

   /**
    * Label to display value
    */
   private JLabel lblValue = new JLabel( "0", JLabel.CENTER );

   /**
    * Current value
    */
   private int value = 0;

   /**
    * Create the binary number and init UI
    */
   public BinaryNumber() {
      super( "Binary Number" );
      this.setLayout( new BorderLayout( 5, 5 ) );
      this.setDefaultCloseOperation( WindowConstants.EXIT_ON_CLOSE );

      final int countBits = 8;
      JPanel panSwitches = new JPanel( new GridLayout( 2, countBits, 5, 5 ) );

      ImageIcon imgOff = new ImageIcon( "off.png" );
      ImageIcon imgOn = new ImageIcon( "on.png" );
```

```
      // 1st row: buttons
      for ( int i = countBits - 1; i >= 0; i-- ) {
         JToggleButton btn = new JToggleButton();
         int num = (int) Math.pow( 2, i ); // calculate value for JToggleButton
         btn.setName( Integer.toString( num ) ); // and set value as name

         btn.setIcon( imgOff ); // icon if not selected
         btn.setSelectedIcon( imgOn ); // icon if selected

         btn.addActionListener( this );
         panSwitches.add( btn );
      }
      // 2nd row: labels
      for ( int i = countBits - 1; i >= 0; i-- ) {
         panSwitches.add( new JLabel( "2^" + i, SwingConstants.CENTER ) );
      }
      // enlarge font for value label
      this.lblValue.setFont( this.lblValue.getFont().deriveFont( 24f ) );

      // add to JFrame
      this.add( panSwitches, BorderLayout.CENTER );
      this.add( this.lblValue, BorderLayout.SOUTH );
      // Adapt size to content
      this.pack();
   }

   /**
    * {@inheritDoc}
    */
   @Override
   public void actionPerformed( ActionEvent e ) {
      // get the source of the event, must be a JToggleButton
      JToggleButton src = ((JToggleButton) e.getSource());

      // parse the name as number
      int num = Integer.parseInt( src.getName() );

      // and add or subtract depending on selected state
      if ( src.isSelected() ) {
         this.value += num;
      } else {
         this.value -= num;
      }
      // update label
      this.lblValue.setText( Integer.toString( this.value ) );
   }

   /**
    * Application entry point
    *
    * @param args command line arguments
    */
   public static void main( String[] args ) {
      BinaryNumber bin = new BinaryNumber();
      bin.setVisible( true );
   }
}
```

Grafische Benutzeroberflächen (UI), Events (1)

Zahlenraten (2) — Musterlösung

Package: `de.dhbwka.java.exercise.ui.event` **Klasse:** `NumberGuess`

```
package de.dhbwka.java.exercise.ui.event;

import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.StringTokenizer;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016-2018 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.1
 */
@SuppressWarnings( "serial" )
public class NumberGuess extends JFrame {

   private int numberToGuess;
   private int countAttempts;
   private int limit = 1000;

   private JButton btnExit = new JButton( "Exit" );
   private JButton btnOk = new JButton( "OK" );
   private JButton btnNew = new JButton( "New Game" );
   private JButton btnStat = new JButton( "Best Player" );
   private JTextField txtName = new JTextField( "Name", 20 );

   private JTextField txtGuess = new JTextField( 10 );
   private JTextField txtOutput = new JTextField( 40 );

   private String statFileName = "stat.txt";

   // Continued on next page
```

```
public NumberGuess() {
   super( "Number Guessing Game" );
   JPanel panName = new JPanel();
   JPanel panNumberinput = new JPanel();
   JPanel panButtons = new JPanel();
   JPanel panOutput = new JPanel();

   panName.add( new JLabel( "Player Name" ) );
   panName.add( this.txtName );

   panNumberinput.add(
         new JLabel( "Enter number beween 1 and " + this.limit ) );
   panNumberinput.add( this.txtGuess );

   panButtons.add( this.btnNew );
   panButtons.add( this.btnOk );
   panButtons.add( this.btnStat );
   panButtons.add( this.btnExit );

   panOutput.add( this.txtOutput );

   this.setLayout( new GridLayout( 4, 1 ) );
   this.add( panName );
   this.add( panNumberinput );
   this.add( panButtons );
   this.add( panOutput );

   this.addEventHandling();
   this.createRandomNumber();

   this.setSize( 500, 250 );
   this.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
   this.setVisible( true );
}
public void addEventHandling() {
   this.btnNew.addActionListener( new ActionListener() {
      @Override
      public void actionPerformed( ActionEvent event ) {
         NumberGuess.this.createRandomNumber();
      }
   } );

   this.btnExit.addActionListener( new ActionListener() {
      @Override
      public void actionPerformed( ActionEvent event ) {
         System.exit( 0 );
      }
   } );

   ActionListener okListener = new ActionListener() {
      @Override
      public void actionPerformed( ActionEvent event ) {
         NumberGuess.this.okActionListener( event );
      }
   };
   this.txtGuess.addActionListener( okListener );
   this.btnOk.addActionListener( okListener );
```

```
      this.btnStat.addActionListener( new ActionListener() {
         @Override
         public void actionPerformed( ActionEvent event ) {
            NumberGuess.this.showBestPlayer();
         }
      } );
   }

   public void okActionListener( ActionEvent event ) {
      try {
         int guess = Integer.parseInt( this.txtGuess.getText() );
         this.countAttempts++;
         this.txtGuess.setText( "" );
         if ( guess > this.numberToGuess ) {
            this.txtOutput.setText( String.format(
                  "Attempt #%s: %s => too big!", this.countAttempts, guess ) );
         }

         else if ( guess < this.numberToGuess ) {
            this.txtOutput.setText( String.format(
                  "Attempt #%s: %s => too small!", this.countAttempts,
                  guess ) );
         }

         else {
            this.txtOutput.setText(
                  String.format( "Attempt #%s: %s => correct!!! New Game!",
                        this.countAttempts, guess ) );
            this.writeStatFile();
            this.createRandomNumber();
         }
      } catch ( NumberFormatException nfe ) {
         this.txtOutput.setText( "Bad input!" );
      }
   }

   private void writeStatFile() {
      try ( FileWriter f = new FileWriter( this.statFileName, true ) ) {
         String name = this.txtName.getText();
         f.write( name + " " + this.countAttempts + " attempts\n" );
      } catch ( Exception e ) {
      }
   }


   // Continued on next page
```

```
   private void showBestPlayer() {
      int minAttempts = Integer.MAX_VALUE;
      String name = "";
      try ( BufferedReader in =
            new BufferedReader( new FileReader( this.statFileName ) ) ) {

         String line;
         do {
            try {
               line = in.readLine();
               if ( line != null ) {
                  StringTokenizer st = new StringTokenizer( line );
                  String currLineName = st.nextToken();
                  try {
                     int currLinePoints = Integer.parseInt( st.nextToken() );
                     if ( currLinePoints < minAttempts ) {
                        name = currLineName;
                        minAttempts = currLinePoints;
                     }
                  } catch ( Exception e ) {
                  }
               }
            } catch ( IOException e ) {
               line = null;
            }
         } while ( line != null );

         this.txtOutput.setText( "Best Player: " + name + ",  " + minAttempts
                     + " attempts" );

      } catch ( Exception ex ) {
      }
   }

   private void createRandomNumber() {
      this.txtGuess.setText( "" );
      this.txtOutput.setText( "New Game!" );
      this.numberToGuess = (int) (Math.random() * this.limit + 1);
      this.countAttempts = 0;

      // For debugging purposes
      System.out.println( "Number to guess: " + this.numberToGuess );
   }

   public static void main( String args[] ) {
      new NumberGuess();
   }

}
```