Technische Universität München
Fakultät für Informatik
Programmieren in Java

WS 1998/99
Übungsblatt 4
29. Oktober 1998

Übungen zum Kurs Programmieren in Java

Anhang zum Übungsblatt 4

Skelett zu Aufgabe 2

Gespeichert in der Rechnerhalle: /usr/proj/javakurs/Tag4/Minimal.skelett

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

public class Minimal extends Frame
  implements ActionListener  {
  public Minimal() {
    Button abutton = new Button("Beenden");
    abutton.addActionListener(this);
    add(abutton);
    pack();
    setVisible(true);
  }

  public void actionPerformed(ActionEvent e) {
     System.exit(0);
  }

  public static void main(String args[]) {
    Minimal m = new Minimal();
  }
}
```


Skelett zu Aufgabe 4

Gespeichert in der Rechnerhalle: /usr/proj/javakurs/Tag4/MitarbeiterListfenster.skelett

```
import java.awt.*;
import java.awt.event.*;
//MitarbeiterListfenster implementiert das Interface ActionListener
//um sicherzustellen, da"s es ActionEvents "uber die
//Methode actionPerformed(..) empfangen kann
public class MitarbeiterListfenster extends Frame
  implements ActionListener  {
  Button buttonende;
  List listmitarbeiter;
  public MitarbeiterListfenster() {
    //erzeugen und einsetzen eines Layout-Managers
    setLayout(new BorderLayout());
    //neue Liste mit 3 gleichzeitig sichtbaren Eintr"agen,
    //ohne Mehrfachselektion -> 2.Konstruktorargument ist false
    listmitarbeiter = new List(3,false);
    //einf"ugen der Listeneintr"age
    listmitarbeiter.add("Huber");
    listmitarbeiter.add("Schmidt");
    listmitarbeiter.add("Obermeier");
    listmitarbeiter.add("Adam");
    //einf"ugen von listmitarbeiter in der Mitte des Frames
    add(listmitarbeiter,BorderLayout.CENTER);
    //neuer Button
    buttonende = new Button("Beenden");
    //beim Bet"atigen des Buttons soll "dieses" Objekt
    //benachrichtigt werden, d.h. die Methode actionPerformed(..)
    //wird aufgerufen
    buttonende.addActionListener(this);
    //der Button soll im Frame unten angeordnet werden -> SOUTH
    add(buttonende,BorderLayout.SOUTH);
    //Window.pack() ordnet die Kindinteraktionsobjekte
    //entsprechend der Anweisungen an den LayoutManager an
    pack();
    //Window.show() zeigt das Fenster am Bildschirm
    show();
    //setTitle funktioniert u.U. nicht auf HP-Rechnern!
    setTitle("MitarbeiterFenster");
  }
  public void actionPerformed(ActionEvent e) {
     System.exit(0);
  }
  public static void main(String args[]) {
    MitarbeiterListfenster abtfenster = new MitarbeiterListfenster();
  }
}
```

Skelett zu Aufgabe 6

Gespeichert in der Rechnerhalle: `/usr/proj/javakurs/Tag4/Abteilungsfenster.skelett`

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;

public class Abteilungsfenster extends Frame
  implements ActionListener   /*ergaenze*/ {
  MenuItem menuitemoeffnen,menuitemspeichern,menuitemende;
  Button buttonchef,buttonneu,buttonaendern,buttonloeschen,buttonende;
  List listmitarbeiter;
  Label labelchef,labelchefdesc;
  TextField textfieldabteilung;
  Mitarbeiterfenster mitarbeiterfenster;
  //auch moeglich: chef-personalnr. anstelle chef-Referenz
  Mitarbeiter chef;
  private static final int LIST_EINTRAG_NAME_LEN=20;
  private static final int LIST_EINTRAG_VORNAME_LEN=10;
  private static final int LIST_EINTRAG_GESCHLECHT_LEN=10;
  public Abteilungsfenster() {
    //Font mit fester Breite der Buchstaben
    setFont(new Font("Courier",java.awt.Font.BOLD,12) );
    //Menu
    MenuBar menubar = new MenuBar();
    this.setMenuBar(menubar);
    Menu menufile = new Menu("Datei");
    menubar.add(menufile);
    menufile.add(menuitemoeffnen = new MenuItem("öffnen"));
    menuitemoeffnen.addActionListener(this);
    menufile.add(menuitemspeichern = new MenuItem("speichern als .."));
    menuitemspeichern.setEnabled(false);
    menuitemspeichern.addActionListener(this);
    menufile.addSeparator();
    menufile.add(menuitemende = new MenuItem("beenden"));
    menuitemende.addActionListener(this);
    setLayout(new BorderLayout());
    Panel panelnorth = new Panel(new GridLayout(2,2));
    panelnorth.add(new Label("Abteilung:"));
    textfieldabteilung = new TextField();
    panelnorth.add(textfieldabteilung);
    labelchefdesc = new Label("ChefIn: ");
    panelnorth.add(labelchefdesc);
    labelchef = new Label();
    panelnorth.add(labelchef);
    //Panel fuer List und TextField
    Panel panelwest = new Panel(new BorderLayout());
    StringBuffer sblistdescr = new StringBuffer();
    sblistdescr.append("Name:");
```



```
    for (int i=0;i<(LIST_EINTRAG_NAME_LEN-5);i++)
      sblistdescr.append(' ');
    sblistdescr.append("Vorname:");
    for (int i=0;i<(LIST_EINTRAG_VORNAME_LEN-8);i++)
      sblistdescr.append(' ');
    sblistdescr.append("Geschl.:");
    for (int i=0;i<(LIST_EINTRAG_GESCHLECHT_LEN-9);i++)
      sblistdescr.append(' ');
     sblistdescr.append("Personal#:");
     // jetzt sieht der in sblistdescr
     // gespeicherte String etwa so aus:
     // Name:               Vorname:  Geschl.:  Personal#:
    Label labellistdescr=new Label(sblistdescr.toString());
    panelwest.add(labellistdescr,BorderLayout.NORTH);
    listmitarbeiter = new List(6,false);
    listmitarbeiter.setBackground(Color.white);
    //ergaenzen um einen Methodenaufruf von listmitarbeiter !
    panelwest.add(listmitarbeiter,BorderLayout.CENTER);
    //Panel fuer Buttons
    Panel paneleast = new Panel(new GridLayout(4,1));
    buttonchef = new Button("als ChefIn");
    buttonchef.addActionListener(this);
    paneleast.add(buttonchef);
    buttonneu = new Button("neu");
    buttonneu.addActionListener(this);
    paneleast.add(buttonneu);
    buttonaendern = new Button("ändern");
    buttonaendern.addActionListener(this);
    paneleast.add(buttonaendern);
    buttonloeschen = new Button("löschen");
    buttonloeschen.addActionListener(this);
    paneleast.add(buttonloeschen);
    //ergaenze um setEnable() der Buttons
    add(panelnorth,BorderLayout.NORTH);
    add(panelwest,BorderLayout.WEST);
    add(paneleast,BorderLayout.EAST);
    buttonende = new Button("Beenden");
    buttonende.addActionListener(this);
    add(buttonende,BorderLayout.SOUTH);
    mitarbeiterfenster = new Mitarbeiterfenster(this);
    pack();
    setVisible(true);
    setTitle("Abteilungsfenster");
  }
  private void neu() {
    setEnabled(false);
    mitarbeiterfenster.setMitarbeiter(
      new Mitarbeiter("","",collections.KeyGen.next(),true));
    mitarbeiterfenster.setVisible(true);
  }
  private void aendern() {
```

```
    setEnabled(false);
    mitarbeiterfenster.setMitarbeiter(getSelListMitarbeiter());
    mitarbeiterfenster.setVisible(true);
  }
  private void loeschen() {
    listmitarbeiter.remove(listmitarbeiter.getSelectedIndex());
    //ergaenze um setEnable() der Buttons
  }
  private void aktualisiere() {
    boolean replace=false;
    for (int i=0;i<listmitarbeiter.getItemCount();i++)
      if (getListMitarbeiter(i).key()==mitarbeiterfenster
           .getMitarbeiter().key()) {
listmitarbeiter.replaceItem(
          makeListEntry(mitarbeiterfenster.getMitarbeiter()),i);
replace=true;
      }
    if (!replace)
      listmitarbeiter.add(makeListEntry(mitarbeiterfenster.getMitarbeiter()));
    if  (chef!=null &&
mitarbeiterfenster.getMitarbeiter().key()==chef.key())
      setChef(mitarbeiterfenster.getMitarbeiter());
    mitarbeiterfenster.setVisible(false);
    setEnabled(true);
    //hier Ergaenzungen zum En/Disablen der Buttons
  }
  private void setChef(Mitarbeiter neuerChef) {
    chef = neuerChef;
    labelchef.setText(chef.nachname);
    labelchefdesc.setText(chef.istWeiblich ? "Chefin:" : "Chef");
    menuitemspeichern.setEnabled(true);
    //ergaenze um setEnable() der Buttons
  }
  private void oeffnen() {
    //ergaenze
  }
  private void speichern() {
    //ergaenze
  }
  /**
   * getAbteilung erzeugt aus dem Inhalt der Interaktionsobjekte
   * ein Abteilungsobjekt.
  */
  private Abteilung getAbteilung() {
    //ergaenze
    return null;
  }
// ************** weiter unten muss keine Ergaenzung erfolgen *****
  /**
   * setAbteilung setzt den Inhalt der Interaktionsobjekte
   * mit dem uebergebenen Abteilungsobjekt.
```



```
  */
  private void setAbteilung(Abteilung abteilung) {
    textfieldabteilung.setText(abteilung.name());
    setChef(abteilung.chef());
    Enumeration e = abteilung.personal().elements();
    listmitarbeiter.removeAll();
    while (e.hasMoreElements()) {
      Mitarbeiter m = (Mitarbeiter)e.nextElement();
      collections.KeyGen.used(m.key());
      listmitarbeiter.add(makeListEntry(m));
    }
    menuitemspeichern.setEnabled(true);
  }
  public void actionPerformed(ActionEvent e) {
    if (e.getSource()==menuitemoeffnen)
      oeffnen();
    else if (e.getSource()==menuitemspeichern)
      speichern();
    else if (e.getSource()==buttonchef)
      setChef(getSelListMitarbeiter());
    else if (e.getSource()==buttonneu)
      neu();
    else if (e.getSource()==buttonaendern)
      aendern();
    else if (e.getSource()==buttonloeschen)
      loeschen();
    else if (e.getSource()==mitarbeiterfenster.buttonok)
      aktualisiere();
    else if (e.getSource()==mitarbeiterfenster.buttonabbrechen) {
      mitarbeiterfenster.setVisible(false);
      setEnabled(true);
    }
    else if (e.getSource()==buttonende || e.getSource()== menuitemende )
      System.exit(0);
 }
  String makeListEntry(String name, String vorname, boolean istWeiblich, int perso
    StringBuffer strbuf = new StringBuffer();
    int diff;
    diff = name.length()-LIST_EINTRAG_NAME_LEN;
    if (diff>=0)
      //Name zu lang oder passend
      strbuf.append(new String(name.substring(0,LIST_EINTRAG_NAME_LEN)));
    else {
      //Name zu kurz -> auf"ullen
      strbuf.append(name);
      for ( ; diff<0 ; diff++ )
        strbuf.append(' ');
    }
    diff = vorname.length()-LIST_EINTRAG_VORNAME_LEN;
    if (diff>=0)
      //Vorname zu lang oder passend
```

```
      strbuf.append(vorname.substring(0,LIST_EINTRAG_VORNAME_LEN));
    else {
      //Vorname zu kurz -> auf"ullen
      strbuf.append(vorname);
      for ( ; diff<0 ; diff++ )
        strbuf.append(' ');
    }
    strbuf.append((istWeiblich ? "weiblich  " : "männlich  "));
    strbuf.append(personalnummer);
    return new String(strbuf);
  }
  String makeListEntry(Mitarbeiter mitarbeiter){
    return makeListEntry(mitarbeiter.nachname,
 mitarbeiter.vorname,
 mitarbeiter.istWeiblich,
 mitarbeiter.key());
  }
  Mitarbeiter getListMitarbeiter(int index) {
    String selItem = listmitarbeiter.getItem(index);
    return new Mitarbeiter(selItem.substring(0,LIST_EINTRAG_NAME_LEN).trim(),
   selItem.substring(LIST_EINTRAG_NAME_LEN,
     LIST_EINTRAG_NAME_LEN+LIST_EINTRAG_VORNAME_LEN).trim(),
   Integer.valueOf(selItem.substring(LIST_EINTRAG_NAME_LEN
     +LIST_EINTRAG_VORNAME_LEN
     +LIST_EINTRAG_GESCHLECHT_LEN)).intValue(),
   selItem.substring(LIST_EINTRAG_NAME_LEN
     +LIST_EINTRAG_VORNAME_LEN).startsWith("w"));
  }
  Mitarbeiter getSelListMitarbeiter() {
    return getListMitarbeiter(listmitarbeiter.getSelectedIndex());
  }
  public static void main(String args[]) {
    Abteilungsfenster abtfenster = new Abteilungsfenster();
  }
}
```


Gespeichert in der Rechnerhalle: `/usr/proj/javakurs/Tag4/Mitarbeiterfenster.java`

```
import java.awt.*;
import java.awt.event.*;
public class Mitarbeiterfenster extends Frame {
  public TextField textfieldname,textfieldvorname;
  public Checkbox checkboxweiblich,checkboxmaennlich;
  public Label labelpersonalnummer;
  public Mitarbeiter mitarbeiter;
  public Button buttonok,buttonabbrechen;
  Mitarbeiterfenster(Abteilungsfenster abteilungsfenster) {
    super("Mitarbeiterfenster");
    setFont(new Font("Courier",java.awt.Font.BOLD,12) );
    setLayout(new GridLayout(5,2));
    add(new Label("Name: "));
    textfieldname = new TextField();
    add(textfieldname);
    add(new Label("Vorname: "));
    textfieldvorname = new TextField();
    add(textfieldvorname);
    CheckboxGroup cbggeschlecht = new CheckboxGroup();
    checkboxweiblich  = new Checkbox("weiblich",cbggeschlecht,true);
    checkboxmaennlich = new Checkbox("männlich",cbggeschlecht,false);
    add(checkboxweiblich);
    add(checkboxmaennlich);
    add(new Label("Personalnummer: "));
    labelpersonalnummer = new Label();
    add(labelpersonalnummer);
    buttonok = new Button("ok");
    buttonok.addActionListener(abteilungsfenster);
    add(buttonok);
    buttonabbrechen = new Button("abbrechen");
    add(buttonabbrechen);
    buttonabbrechen.addActionListener(abteilungsfenster);
    pack();
  }
   public void setMitarbeiter(Mitarbeiter mitarbeiter) {
       textfieldname      .setText(mitarbeiter.nachname);
       textfieldvorname   .setText(mitarbeiter.vorname);
       checkboxweiblich   .setState( mitarbeiter.istWeiblich);
       checkboxmaennlich  .setState(!mitarbeiter.istWeiblich);
       labelpersonalnummer.setText(String.valueOf(mitarbeiter.key()));
  }
  public Mitarbeiter getMitarbeiter() {
    return new Mitarbeiter(textfieldname.getText(),
   textfieldvorname.getText(),
   Integer.valueOf(labelpersonalnummer.
   getText()).intValue(),
   checkboxweiblich.getState());
  }
}
```