

Programmeringsteknik för Bio1 och I1

Extra om grafik

Övningsgrupp 3 (Sal E33)

Johannes Hjorth

hjorth@nada.kth.se

Rum 4538 på plan 5 i D-huset

08 - 790 69 02

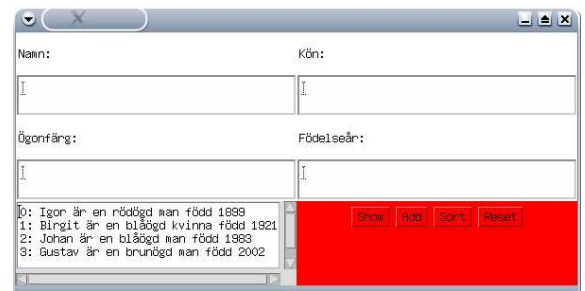
Kurshemsida:

<http://www.nada.kth.se/kurser/kth/2D1310/>

Övningsanteckningar och diagnostiska prov:

<http://www.nada.kth.se/~hjorth/teaching/prgibio04/>

Hur gör man enkel grafik?



Detta är bara ett exempel på hur man kan bygga ut sitt textbaserade program med grafik utan att behöva ändra något i ursprungsprogrammet.

Jag börjar med att lista den kompletta koden.

PersonGrafik.java

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

class PersonGrafik {

    public static void main(String[] args) {
        Personer p = new Personer("personer.txt");
        PersonFrame f = new PersonFrame(p);

        f.show(); // Visa vår frame på skärmen
    }
}

class PersonFrame extends Frame implements ActionListener{

    private TextQuestion tqName, tqSex, tqColour, tqAge;
    private ControlButtons cb;
    private Personer pers;
    private TextArea text;

    public PersonFrame(Personer pers){

        this.pers = pers; // Personer-klassen som gör allt jobb

        // Bestäm vilken layout vi vil ha
        GridLayout g = new GridLayout(3,2);
        setLayout(g);

        // Sätt font och färger
        Font f = new Font("Helvetica", Font.PLAIN, 14);
        Color c = new Color(220,220,220); // R G B värden

        setSize(600,300);
        setBackground(c);
        setFont(f);

        // Skriv ut välkomst text
        text = new TextArea("Hej och välkomna!", 6, 40);

        tqName = new TextQuestion("Namn:", 20);
        tqSex = new TextQuestion("Kön:", 6);
        tqColour = new TextQuestion("Ögonfärg:", 10);
        tqAge = new TextQuestion("Födelseår:", 4);

        cb = new ControlButtons(this);

        add(tqName); // Lägg till dem på skärmen
        add(tqSex);
        add(tqColour);
        add(tqAge);

        add(text);
        add(cb);

        // Se till att fönstret kan stängas
        WindowAdapter w = new WindowAdapter()
        { public void windowClosing(WindowEvent e) { System.exit(0); } };

        addWindowListener(w);

    }

    public void actionPerformed(ActionEvent e) {
        Object s = e.getSource(); // Ta reda på var eventen skedde

        if(cb.equals>ShowButton(s)) {
            //Skriv ut på grafikskärmen
            text.setText(pers.toString());
        }
        else if(cb.equalsAddButton(s)) {

            try{
                String namn = tqName.getText();
                String kön = tqSex.getText();
                String färg = tqColour.getText();
            }
        }
    }
}
```

```

        int ålder    = Integer.parseInt(tqAge.getText());

        if(ålder > 0 && namn != null && kön != null && färg != null) {

            Person p = new Person(namn, ålder, kön, färg);

            pers.addPerson(p);

            tqName.clearText(); // clear fields
            tqAge.clearText();
            tqSex.clearText();
            tqColour.clearText();

            text.setText(pers.toString());
        }
        else {
            text.setText("Malformed data!");
        }
    }

    catch(NumberFormatException nfe) {
        text.setText("Malformed data!");
    }
}

else if(cb.equalsSortButton(s)) {
    pers.sorteraÅlder();
    text.setText(pers.toString());
}

else if(cb.equalsResetButton(s)) {
    pers = new Personer("personer.txt");
    text.setText(pers.toString());
}
}
}

```

```

class TextQuestion extends Panel {

    private Label text;

```

```

        add(resetButton);

        showButton.addActionListener(a);
        addButton.addActionListener(a);
        sortButton.addActionListener(a);
        resetButton.addActionListener(a);
    }

    public boolean equalsShowButton(Object o) {
        return (o == showButton);
    }

    public boolean equalsAddButton(Object o) {
        return (o == addButton);
    }

    public boolean equalsSortButton(Object o) {
        return (o == sortButton);
    }

    public boolean equalsResetButton(Object o) {
        return (o == resetButton);
    }
}

```

```

private TextField textField;

public TextQuestion(String text, int fieldLength) { // ta actionListener

    GridLayout g = new GridLayout(2,1);

    setLayout(g);
    setBackground(Color.white);

    this.text = new Label(text);
    textField = new TextField(fieldLength);

    add(this.text);
    add(textField);
}

public String getText() {
    return textField.getText();
}

public void clearText() {
    textField.setText("");
}

}

class ControlButtons extends Panel {

    private Button showButton, addButton, sortButton, resetButton;

    public ControlButtons(ActionListener a) { // Ta actionListener som inparam
        setBackground(Color.red);

        showButton = new Button("Show");
        addButton = new Button("Add");
        sortButton = new Button("Sort");
        resetButton = new Button("Reset");

        add(showButton);
        add(addButton);
        add(sortButton);

```

Ej komplett genomgång

För att nå grafikfunktionerna måste vi inkludera följande,

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

Notera sedan hur vi instansierar vår gamla Personer-klass och låter den utföra allt arbete.

Genomgående använder vi add för att lägga ihop olika komponenter, och genom att sedan göra en show visas de på skärmen.

```

class PersonFrame extends Frame implements
ActionListener

```

Raden ovan betyder att PersonFrame ärver egenskaper från klassen Frame samt lovar att göra allt som en ActionListener ska göra.

Klassen TextQuestion är en egenskriven klass för att slippa kodupprepning.

All action sker i actionPerformed vilken körs när någon av knapparna trycks ner. Observera addActionListener anropen!

Mycket mer kan sägas, men förhoppningsvis kan detta exempel hjälpa er att komma igång med grafik.

Person.java

```
class Person implements Comparable {

    private String namn;
    private int födelseår;
    private String ögonfärg;
    private String kön;

    Person(String namn, int födelseår, String kön, String ögonfärg) {
        this.namn = namn;
        this.födelseår = födelseår;
        this.kön = kön;
        this.ögonfärg = ögonfärg;
    }

    public String toString() {
        return namn + " är en " + ögonfärg + "ögd " + kön + " född " + födelseår;
    }

    public int getFödelseår() {
        return födelseår;
    }

    public int compareTo(Object o) {
        Person p = (Person) o;

        return födelseår - p.getFödelseår();
    }
}
```

Personer.java

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

class Personer {

    private Vector personer;

    Personer(String filnamn) {
        personer = new Vector();
        läsInFrånFil(filnamn);
        sorteraÅlder();
    }

    private void läsInFrånFil(String filnamn) {
        SimpleTextFileReader f = new SimpleTextFileReader(filnamn);
        StringTokenizer st;
        Person p;

        String namn, kön, ögonfärg, s;
        int ålder;

        s = f.readLine();          // läser in första raden

        while(s != null) {         // kollar om aktuell rad är tom

            //System.out.println("Inläst rad: " + s);

            st = new StringTokenizer(s);

            namn = st.nextToken();
            ålder = Integer.parseInt(st.nextToken());
            kön = st.nextToken();
            ögonfärg = st.nextToken();

            p = new Person(namn, ålder, kön, ögonfärg);
            personer.addElement(p);
        }
    }
}
```

```
        //System.out.println("Persondata: " + p);

        s = f.readLine();          // läser in nästa rad
    }

    f.close();                    // stänger filen
}

//Ändrad till public från private
public void sorteraÅlder() {
    Collections.sort(personer);
}

public String toString() {

    StringBuffer s = new StringBuffer();
    Person p;

    for(int i = 0; i < personer.size(); i++) {
        p = (Person) personer.elementAt(i);
        s.append(i + ": " + p.toString() + "\n");
    }

    return s.toString();
}

//Ny metod
public void addPerson(Person p) {
    personer.addElement(p);
}

public static void main(String[] args) {
    Personer p = new Personer("personer.txt");
    System.out.println(p);
}
}
```

Jag har lagt till en metod addPerson och ändrat en private till public, i övrigt har ingenting ändrats.

personer.txt

```
Gustav 2002 man brun  
Johan 1983 man blå  
Birgit 1921 kvinna blå  
Igor 1899 man röd
```

Personerna ligger lagrade i `personer.txt`. En övning för den med lite tid och intresse skulle kunna vara att lägga till en metod som kan spara de nyinmatade personerna på fil med valfritt namn.