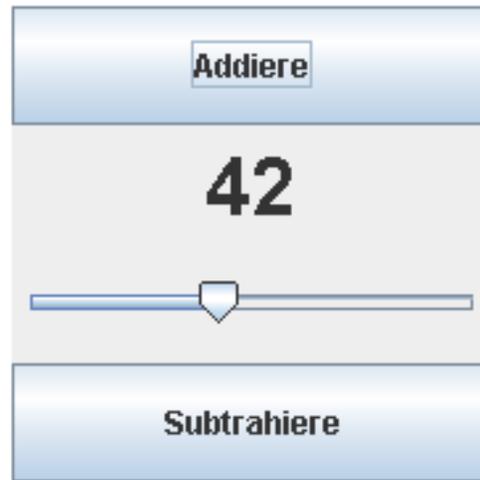


Kapitel 2: Grafische Benutzeroberflächen



GUI-Programmierung

- Windowmanager
- AWT (Abstract Window Toolkit)
 - reicht Kommandos weiter an Betriebssystem
 - plattformabhängig
- Swing
 - pure Java
 - einheitliches Look & Feel

GUI-Komponenten

- JFrame
- GridLayout
- JButton
- JLabel
- JSlider
- ActionListener
- actionPerformed

RaufRunterApplikation.java, Teil 1

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

public class RaufRunterApplikation extends JFrame {

    private int      zaehler    = 42;
    private JButton  rauf        = new JButton("Addiere");
    private JLabel   ergebnis  = new JLabel("42 ",JLabel.CENTER);
    private JSlider  schieber    = new JSlider(0, 100, zaehler);
    private JButton  runter      = new JButton("Subtrahiere");

    public RaufRunterApplikation() { ... }

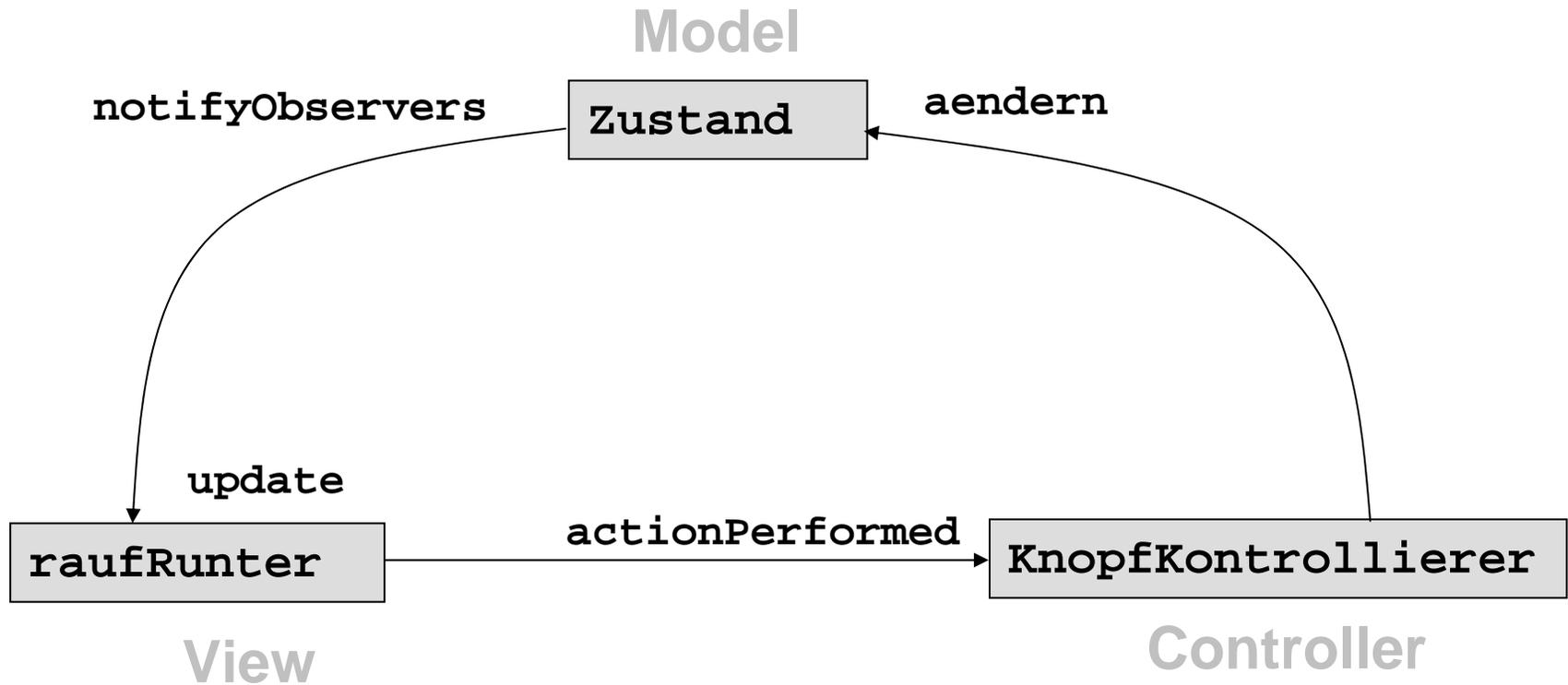
    public static void main (String [] args) {
        new RaufRunterApplikation();
    }
}
```

RaufRunterApplikation, Teil 2

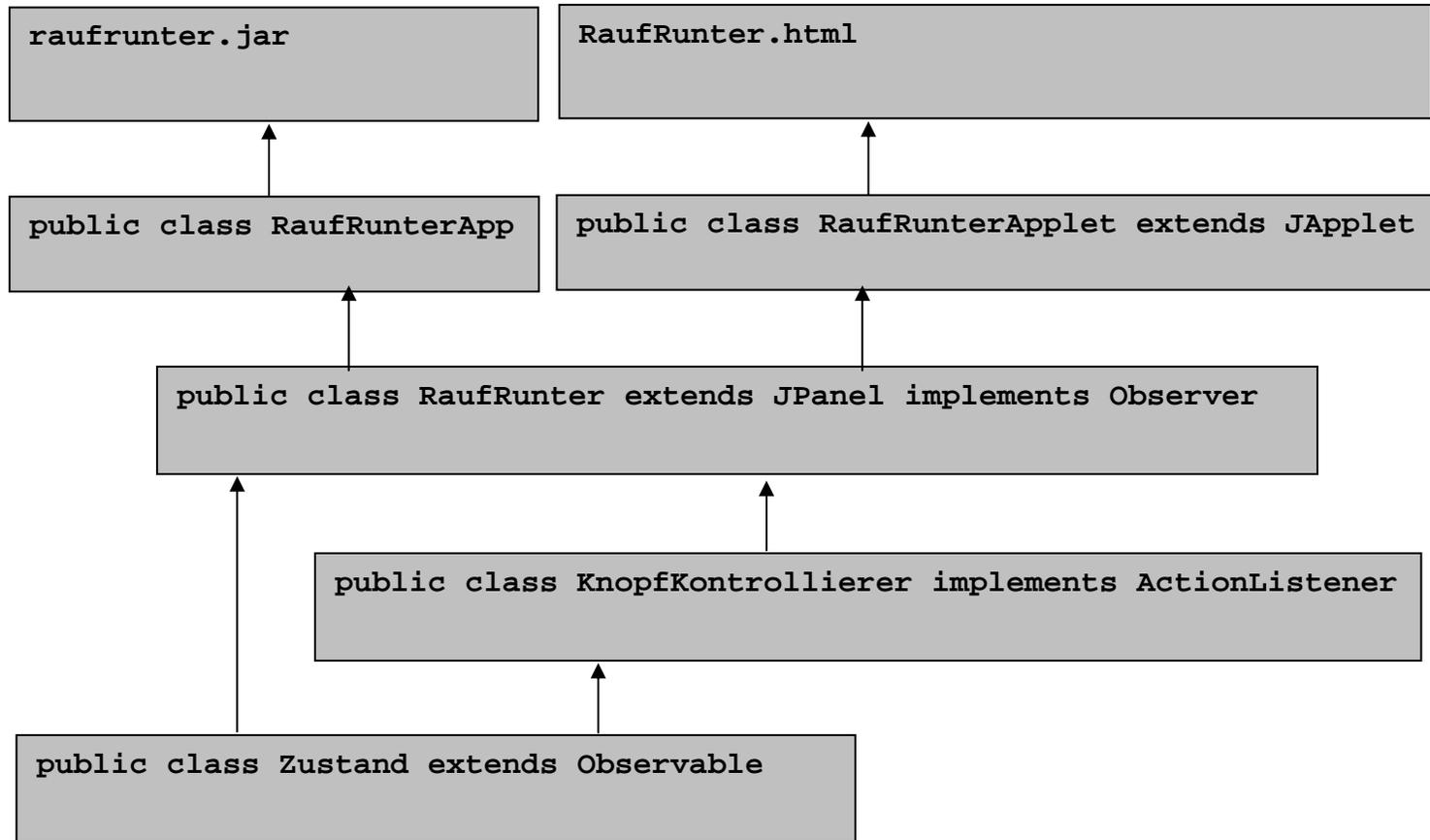
```
public RaufRunterApplikation() {  
  
    setLayout(new GridLayout(0,1));  
    add(rauf); add(ergebnis); add(schieber); add(runter);  
  
    rauf.addActionListener(new ActionListener(){  
        public void actionPerformed(ActionEvent e){  
            zaehler++;  
            ergebnis.setText(zaehler + " ");  
            schieber.setValue(zaehler);  
        }  
    });  
  
    runter.addActionListener(new ActionListener(){  
        public void actionPerformed(ActionEvent e){  
            zaehler--;  
            ergebnis.setText(zaehler + " ");  
            schieber.setValue(zaehler);  
        }  
    });  
  
    pack(); setVisible(true);  
}
```

Controller
Model
View

Model-View-Controller



Dateisystem



Zustand.java

```
import java.util.Observer;
import java.util.Observable;

public class Zustand extends Observable{

    private int zaehler;

    public Zustand(int zaehler){
        this.zaehler=zaehler;
    }

    int get(){return zaehler;}

    void aendern(int delta){
        zaehler = zaehler + delta;
        setChanged();
        notifyObservers();
    }
}
```

KnopfKontrollierer.java

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

public class KnopfKontrollierer implements ActionListener {

    private Zustand z;
    private int delta;

    public KnopfKontrollierer(Zustand z, int delta) {
        this.z      = z;
        this.delta  = delta;
    }

    public void actionPerformed(ActionEvent e) {
        z.aendern(delta);
    }
}
```

RaufRunter.java, Teil 1

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

public class RaufRunter extends JPanel implements Observer {

    private JButton rauf;
    private JButton runter;
    private Zustand z;
    private JLabel  ergebnis;
    private JSlider schieber;
    private Font    font;

    public void update(Observable z, Object dummy){
        ergebnis.setText(((Zustand)z).get() + " ");
        schieber.setValue(((Zustand)z).get());
    }
}
```

RaufRunter.java, Teil 2

```
public RaufRunter() {
    setLayout(new GridLayout(0,1));
    rauf    = new JButton("Addiere");
    runter  = new JButton("Subtrahiere");
    schieber = new JSlider(0,100,42);
    ergebnis = new JLabel("42",JLabel.CENTER);
    font = new Font("SansSerif",Font.BOLD,30);
    ergebnis.setFont(font);
    add(rauf);
    add(ergebnis);
    add(schieber);
    add(runter);
    z = new Zustand(42);
    z.addObserver(this);
    KnopfKontrollierer raufK;
    raufK = new KnopfKontrollierer(z,+1);
    rauf.addActionListener(raufK);
    KnopfKontrollierer runterK;
    runterK = new KnopfKontrollierer(z,-1);
    runter.addActionListener(runterK);
}
}
```

RaufRunterApp.java

```
import java.awt.BorderLayout;
import javax.swing.JFrame;

public class RaufRunterApp {

    public static void main(String args[]) {
        JFrame rahmen = new JFrame("RaufRunter-Applikation");
        rahmen.add(new RaufRunter(),BorderLayout.CENTER);
        rahmen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        rahmen.pack();
        rahmen.setVisible(true);
    }
}
```

RaufRunterApplet.java

```
import java.awt.BorderLayout;
import javax.swing.JApplet;

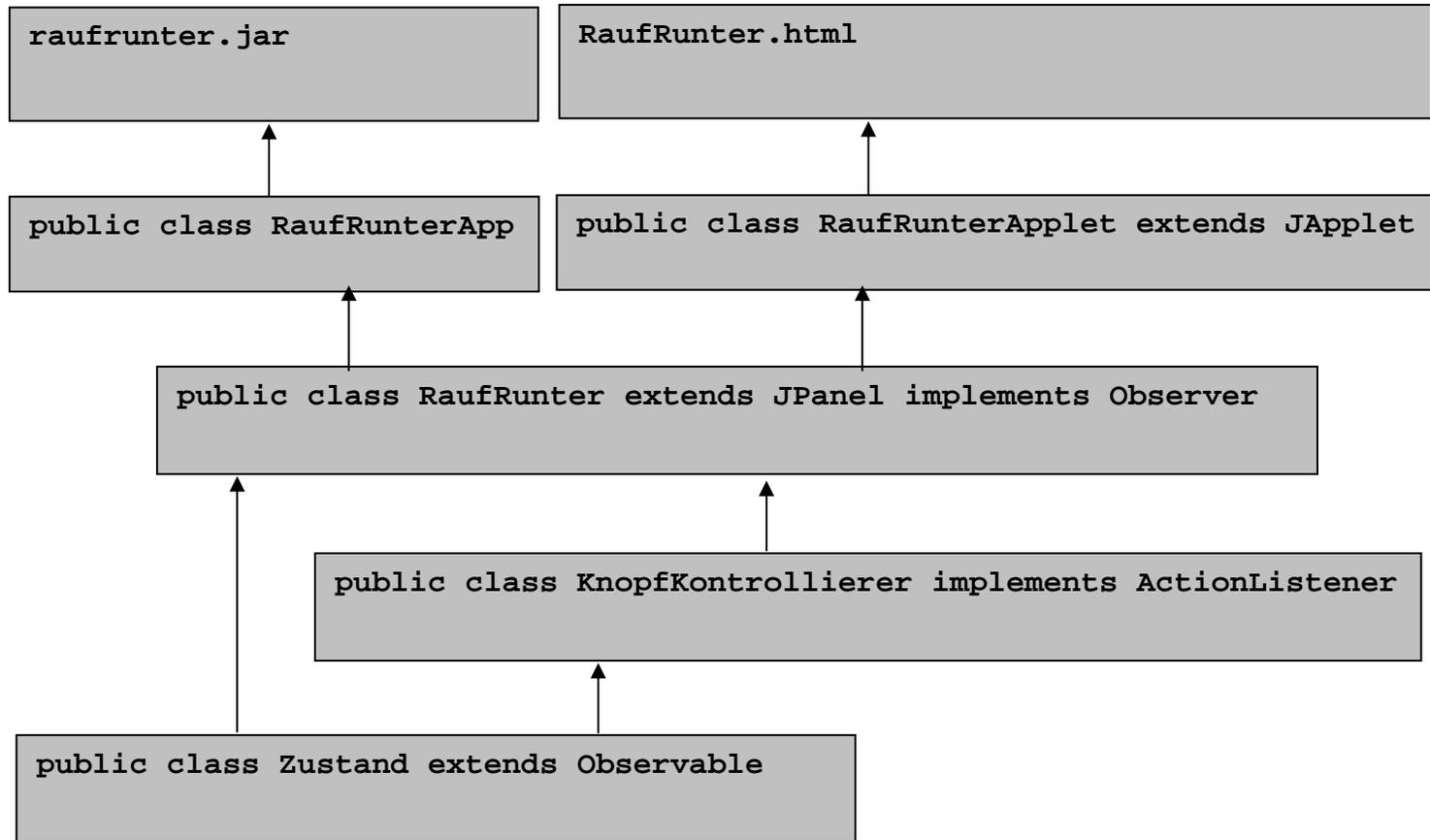
public class RaufRunterApplet extends JApplet {

    public void init() {
        add(new RaufRunter(), BorderLayout.CENTER);
    }
}
```

RaufRunter.html

```
<HTML>
  <HEAD>
    <TITLE>RaufRunter-Applet</TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1>RaufRunter-Applet</H1>
      <APPLET
        width    = 200
        height   = 150
        code     = "RaufRunterApplet.class"
        archive  = "raufRunter.jar">
      </APPLET>
    </CENTER>
  </BODY>
</HTML>
```

Dateisystem



makefile

```
TARGETS = Zustand.class KnopfKontrollierer.class RaufRunter.class
          RaufRunterApp.class RaufRunterApplet.class
all:
    make $(TARGETS)
jar: $(TARGETS)
    make doc
    -(jar cmvf MainClass.txt raufRunter.jar Zustand*
    KnopfKontrollierer* RaufRunter* makefile Doc MainClass.txt
    App.html)
doc:
    rm -rf Doc
    mkdir Doc
    javadoc -link http://java.sun.com/j2se/1.5/docs/api -version -
    author -private -d Doc *.java
app:
    $(TARGETS)
    java RaufRunterApp
%.class:      %.java
              javac $*.java
clean:
    rm -f *.class
    rm raufRunter.jar
    rm -rf Doc
```

[MainClass.txt](#)

[Doc](#)