

Felder

1. Bezug zum Unterricht: Anwendungen

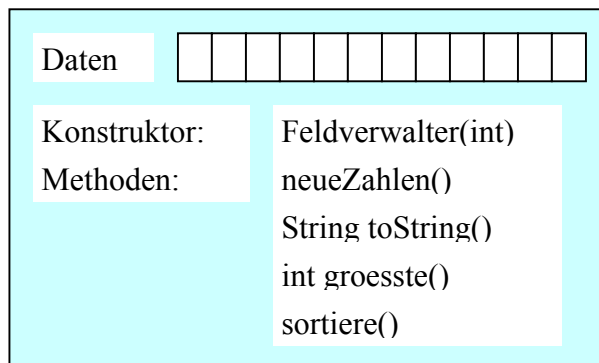
Die im Unterricht benutzten Elemente einer Programmiersprache sollten sich nicht aus Vollständigkeits- oder anderen fachimmanenten Überlegungen ergeben, sondern möglichst aus einer Problemstellung abgeleitet werden, die Bezug zu den Möglichkeiten und Folgen der *Anwendung von Informatiksystemen* hat. Für Felder (*Arrays*) ergeben sich viele Anwendungsmöglichkeiten, da schon im Anfangsunterricht größere Datenmengen verarbeitet werden können, ohne eine echte Datei- oder Datenbankorganisation zu benötigen:

- Reine Datenverarbeitung (*Daten verwalten, sortieren, in Daten suchen, ...*)
- Bildbearbeitung (*Bilder erzeugen, verändern, konvertieren, darstellen, ...*)
- mit Zufallszahlen arbeiten (*Bereiche festlegen, ordnen, Verteilung, doppelte Z., ...*)
- Spiele (, ...)
- in Koordinatensystemen arbeiten (*zelluläre Automaten, Simulationen, ...*)

Da die Daten oft mühsam eingegeben werden müssen, sollte eine Hilfsklasse bereitgestellt werden, mit deren Hilfe Felddaten gespeichert und wieder geladen werden können.

2. Eine eigene Klasse *Feldverwalter*

Wir sollten die Verwaltung von Felddaten von der Benutzung trennen – schon um der Übersichtlichkeit willen. Deshalb führen wir eine eigene Klasse *Feldverwalter* ein, deren Instanzen ein Feld verwalten und den Zugriff regeln. Ein Feldverwalter sieht dann so aus:



Felder in Java sind semidynamisch, d. h. sie können zur Laufzeit mit (fast) beliebiger Größe instanziiert werden, sind danach aber statisch. Felder ganzer Zahlen erhält man z. B. durch

```
int daten[] = new int[6]; oder int daten[] = new int[i];
```

Hat ein Feld die Größe n , dann läuft sein Index von 0 bis $(n-1)$. Die aktuelle Größe des Feldes erhält man über die Instanzvariable `length` (OHNE KLAMMERN!).

Eigene Klassen benötigen einen Konstruktor zur Erzeugung von Klasseninstanzen, der so heißt wie die Klasse. In unserem Fall wollen wir dem Konstruktor die Größe des benötigten Feldes mit übergeben. Dieser legt das Feld dann an.

```
Feldverwalter(int i){daten = new int[i];}
```

Um „Lottozahlen“ zu erzeugen, können wir die mit Hilfe des Zufallsgenerators bestimmen und mit Hilfe des *Math*-Objekts in die gewünschte Form bringen. Dazu müssen wir noch ein Typecasting durchführen, da durch Rundungen Werte des Datentyps *long* entstehen,

```
(int)Math.round(Math.random()*49+0.5);
```

Das Feld *daten* wird dann innerhalb einer Schleife gefüllt, die die Feldgröße abfragt.

```
public void neueZahlen()
{
    for(int i=0;i<daten.length;i++)
        daten[i] = (int)Math.round(Math.random()*49+0.5);
}
```

Entsprechend kann die größte vorhandene Zahl abgefragt werden:

```
public int groesste()
{
    int max=0;
    for (int i=0;i<daten.length;i++)
        if (daten[i]>max) max = daten[i];
    return max;
}
```

oder der Feldinhalt wird in einen String umgewandelt:

```
public String toString()
{
    String h="[ ";
    for(int i=0;i<daten.length-1;i++) h = h + daten[i]+", ";
    h = h+daten[daten.length-1]+" ]";
    return h;
}
```

Natürlich kann man die Felddaten auch sortieren:

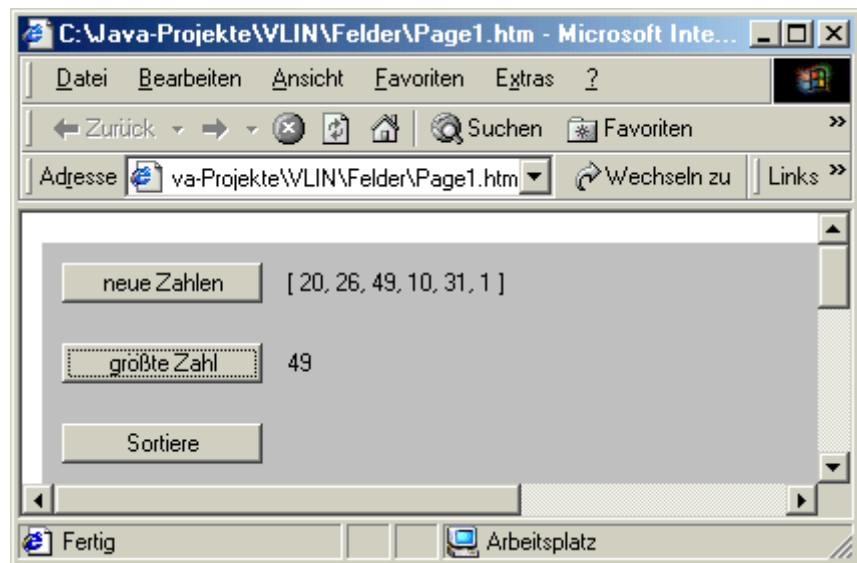
```
public void sortiere()
{
    int h;
    for(int i=0;i<daten.length-1;i++)
        for(int j=0;j<daten.length-i-1;j++)
            if(daten[j]>daten[j+1])
            {
                h = daten[j];
                daten[j] = daten[j+1];
                daten[j+1] = h;
            }
}
```

Insgesamt erhält man (die Klasse darf nicht als *public* deklariert werden!):

```
class Feldverwalter{
    int daten[];
    Feldverwalter(int i){...}
    public String toString(){...}
    public void sortiere()
    public void neueZahlen(){...}
    public int groesste()
}
```

3. Den Feldverwalter benutzen

Wir wollen in einem Applet den Feldverwalter zur Ziehung von Lottozahlen einsetzen. Dazu erzeugen wir in der bekannten Art GUI-Komponenten und verknüpfen sie mit Hilfe eines Action-Listeners (dem Applet selbst) mit den Methoden des Feldverwalters.



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

```
public class Applet1 extends Applet implements ActionListener
{
    TextField eingabe = new TextField("Eingabe");
    Button neu = new Button("neue Zahlen");
    Button groesste = new Button("größte Zahl");
    Button sortiere = new Button("Sortiere");
    Label anzeige = new Label("--");
    Label ganzeige = new Label("--");
    Feldverwalter lottozahlen = new Feldverwalter(6);

    public void init()
    {
        setLayout(null);
        neu.setBounds(10,10,100,20); neu.addActionListener(this); add(neu);
        anzeige.setBounds(120,10,200,20); add(anzeige);
        groesste.setBounds(10,50,100,20);
        groesste.addActionListener(this); add(groesste);
        ganzeige.setBounds(120,50,50,20); add(ganzeige);
        sortiere.setBounds(10,90,100,20);
        sortiere.addActionListener(this); add(sortiere);
        anzeige.setBounds(120,10,200,20); add(anzeige);
    }

    public void actionPerformed(ActionEvent e)
    {
        String aufschrift = e.getActionCommand();
        if(aufschrift == "neue Zahlen")
        {lottozahlen.neueZahlen(); anzeige.setText(lottozahlen.toString());}
        if(aufschrift == "größte Zahl")
        {ganzeige.setText(new Integer(lottozahlen.groesste()).toString());}
        if(aufschrift == "Sortiere")
        {lottozahlen.sortiere(); anzeige.setText(lottozahlen.toString());}
    }
}
```

4. Aufgaben

1. a: Schreiben Sie eine Methode *kleinstes()* des Feldverwalters.
b: Sorgen Sie dafür, dass keine doppelten Zahlen auftreten.
c: Ändern Sie den Feldverwalter so, dass der Bereich, aus dem die Zufallszahlen stammen, mit dem Konstruktor festgelegt werden können.
d: Ermöglichen Sie es, die Zahlen absteigend zu sortieren.
e: Implementieren Sie eine bessere Sortiermethode.
f: Implementieren Sie eine Methode *mittelwert()* des Feldverwalters.
2. a: Informieren Sie sich im Netz über verschiedene Sortierverfahren.
b: Schreiben Sie Applets, die die Arbeitsweise jeweils eines Verfahrens grafisch ansprechend darstellen.
3. a: Schreiben Sie ein Applet, mit dessen Hilfe sich die Felder eines 7x8-Arrays durch Anklicken schwarz bzw. weiß färben lassen.
b: Versuchen Sie eine Methode zu schreiben, die ein Muster in diesem Array finden kann, z. B. ein Kreuz.
c: Machen Sie sich Gedanken darüber, wie Mustererkennung z. B. mit Hilfe einer Quick-Cam im Unterricht eingesetzt werden kann. Leiten Sie unterschiedliche Aufgabenstellungen und deren Bezug zum Unterricht ab.
4. a: Gegeben seien zwei Felder mit Zufallszahlen. Diese sollen so aneinander gehängt werden, dass ein einziges Feld entsteht.
b: Sortieren Sie das ganze Feld unter der Annahme, dass die beiden Felder schon sortiert waren.
c: Sortieren Sie in das Feld einzelne weitere Zahlen ein.
5. a: Analysieren Sie das Spiel „Schiffeversenken“ auf Möglichkeiten, es projektartig im Unterricht zu realisieren.
b: Schreiben Sie Teile des Spiels als Applet.