

Reihungen

Sven Eric Panitz

Reihungen

- Neben den primitiven Typen (char, int, double,...) und Klassen kennt Java noch einen besonderen Typen:
- Reihungen engl. arrays (Deutsch auch als Datenfelder bezeichnet)

Deklaration und Initialisierung von Reihenungen

- In einer Variablen von einem bestimmten Typ, z.B.:

```
String x = "Hallo";
```

kann ein Element (hier ein Stringobjekt) gespeichert werden.

- Mit eckigen Klammern an den Typ, macht man eine Variable zu einer Reihung von diesem Typ und kann so mehrere Elemente in dieser Variablen speichern:

```
String[] xs = { "Freunde", "Römer ", "Landsleute"  
              , "leiht", "mir", "euer", "Ohr"};
```

- In der Variablen xs sind jetzt 7 String-Objekte gespeichert.

Zugriff auf die Länge einer Reihung

- ähnliche wie die Objekte einer Klasse, kann für Reihungen ein Feld angesprochen werden. Dieses Feld heißt `length` und gibt in einem `int`-Wert die Anzahl der Elemente der Reihung an:

```
String[] xs = {"Freunde", "Römer", "Landsleute"};  
int anzahl = xs.length;  
System.out.println(anzahl);
```

Zugriff auf Element über einen Index

- Auf die einzelnen Elemente der Reihung kann über ein Index zugegriffen werden.
- Dieser Index wird in eckigen Klammern einer Reihung angehängt.

```
int[] xs = {17, 4, 25, 42};  
var x1 = xs[1];  
var x3 = xs[3]
```

- Der Index läuft von 0 bis length-1.
- Ein Index außerhalb dieses Bereichs führt zu einem Laufzeitfehler.

Modifizieren von Reihenungen

- Über den Index können auch die Elemente der Reihung einen neuen Wert bekommen.

```
int[] xs = {1, 2, 3, 4};  
xs[3] = 42;
```

Erzeugen von Reihenungen mit uninitialized Elementen

- Reihenungen können auch nur durch Angabe ihrer Länge erzeugt werden.
- Hierzu dient wie bei Objekten das Schlüsselwort `new`.
- Die Elemente sind bei Objekten dann mit `null`, bei primitiven Typen mit `0` initialisiert.
- Beispiel: Erzeugung eines `int`-Arrays für 10 Elemente und eines `String`-Arrays mit 200 Elementen.

```
var xs = new int[10];  
var ys = new String[200];
```

Feste Größe

- Wurde eine Reihung erzeugt, hat dieser die bei der Erzeugung angegebenen Länge. Diese kann nicht mehr verändert werden.

Iteration durch alle Elemente einer Reihung

- Durch alle Elemente einer Reihung kann mit der foreach-Schleife iteriert werden:

```
int [] xs = {1, 32, 31, 321, 321, 321, 1, 21, 45, 4, 436, 45, 245, 9};  
for (var x:xs) {  
    System.out.println(x*x);  
}
```

- Mit der foreach-Schleife können die Elemente nicht neu zugewiesen werden.
- Für die foreach-Schleife gibt es auch `break` und `continue`.

Iteration mit einer Indexvariablen

- Man kann auch von Hand mit einer Indexvariablen über Reihenungen iterieren.
- Dann lassen sich auch neue Elemente in eine Reihung schreiben:

```
var xs = new int[10];  
for (int i=0;i<xs.length;i=i+1) {  
    xs[i] = i*i;  
}
```

- Hier werden die Quadratzahlen 0^2 bis 9^2 in die Reihung gespeichert.

Mehrdimensionale Reihenungen

- Die Elemente einer Reihung können wieder Reihenungen sein. Dann enthält man mehrdimensionale Reihenungen.
- So ist `int [][]` eine Reihung, deren Elemente Reihenungen von Zahlen sind.
- Mehrdimensionale Reihenungen können durch einen Aufruf mit `new` erzeugt werden:

```
var xss = new int[10][20]
```

- Elemente mehrdimensionale Reihenungen werden durch mehere Indizes angesprochen:

```
int i = xss[2][9];
```

Zusammenfassung

- Reihungen sind Aneinanderreihungen von Elementen gleichen Typs.
- Reihungstypen werde durch das Klammernpaar [] deklariert.
- Reihungen haben eine feste Länge.
- Reihungen können durch Aufzählung der Elemente oder durch einen Aufruf mit new initialisiert werden.
- Mit der foreach-Schleife lässt sich bequem durch Reihungen iterieren.
- Per Index lässt sich auf die einzelnen Elemente einer Reihung zugreifen. Index startet bei 0.