

Arrays Einführung

Reihungen – Arrays

Würfeln

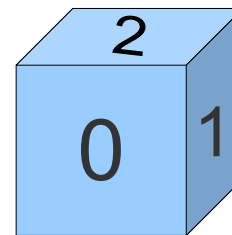
- **Beispiel:** Programm, das 100 Würfe mit einem (sechsseitigen) Würfel simuliert. Es soll gezählt werden, wie oft welche Augenzahl fällt.
- **Achtung:** Der Würfel der Informatiker hat die Augenzahlen 0 - 5.



The screenshot shows a Java Swing window titled "wuerfel1 : Wuerfeln". It contains a table of static variables and their counts, along with several buttons. The table is as follows:

Variable	Count
int anzahl0	16
int anzahl1	22
int anzahl2	16
int anzahl3	14
int anzahl4	18
int anzahl5	14

Buttons in the window include "Inspiziere", "Hole", "Zeige statische Variablen", and "Schließen".



Würfeln



- Man sieht: Die Variablen **anzahl0**, **anzahl1**, ..., **anzahl15** sind redundant:
 - Gleicher Typ: **int**
 - Gleichartige Daten: Anzahl, wie oft eine Seite geworfen wurde
 - Sie werden gleichartig behandelt:
 - Am Anfang werden sie alle auf 0 gesetzt.
 - Wenn die Zahl geworfen wird, wird ihr Wert um 1 erhöht.

Arrays

- Die Variablen `anzahl10`, ..., `anzahl15` werden in eine Datenstruktur namens **Array** ("Feld") zusammengefasst. Deklaration

- `int[] anzahlen;`

"Die Variable `anzahlen` ist ein Array von `int`-Werten."

- `anzahlen = new int[6];`

"Es wird ein neues Array mit Platz für 6 `int`-Zahlen angelegt und der Variablen `anzahlen` zugewiesen."

- Beide Befehle kann man auch zusammenfassen oder den Array Elementen direkt Werte zuweisen:

```
int[] anzahlen = new int[6];
```

```
int[] anzahlen = {1,4,2,6,7,2};
```

Arrays

- Einzelne Elemente des Arrays können über ihren **Index** (Position im Array) gelesen und geschrieben werden:
 - `int[] anzahlen = new int[6];`
`anzahlen[0] = 5;`
`int x = anzahlen[3];`
- Die Anzahl der Elemente im Array (d.h. seine Länge) kann über seine Eigenschaft `length` abgefragt werden:
 - `int laenge = anzahlen.length;`
Wert von `laenge`: **6**
- **Wichtig:** Das erste Element des Arrays `a` hat den Index `0`, das letzte hat den Index `a.length-1`

Arrays

- Begriffsunterscheidung: **Array**, **Elemente** des Arrays und deren **Index**.
- Bildliche Vorstellung:
 - Fussballmannschaft = Array
 - Spieler = Element
 - Trikotnummer = Index



Arrays

- Arrays anderer Datentypen können ebenso angelegt werden:
 - `double[] a = new double[17];`
 - `String[] woerter = new String[100];`
 - u.ä.
- Die Größe eines Arrays kann nachträglich nicht mehr geändert werden. Der Speicher wird bei der Deklaration alloziert.
- Man kann einer Array-Variablen jedoch ein neues Array zuweisen:

```
int[] zahlen = new int[6];
```

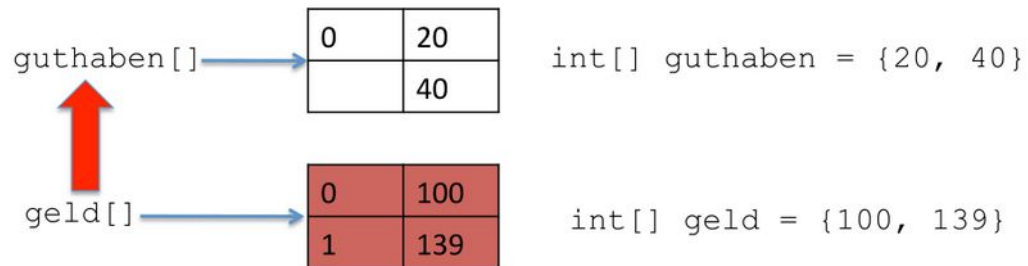
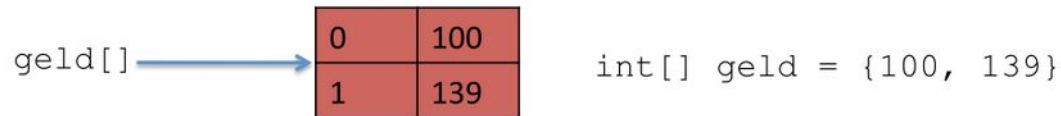
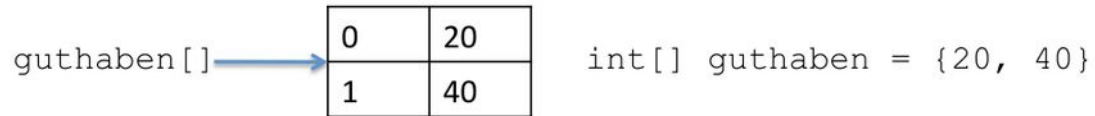
```
...
```

```
zahlen = new int[10];
```

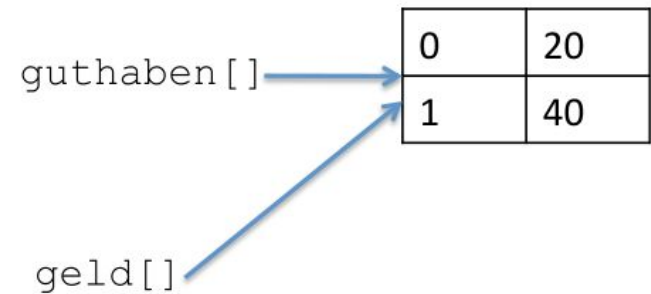
In diesem Fall wird der alte Inhalt verworfen - es wird neuer Speicher alloziert.

Arrays

Array sind Referenztypen - sie verweisen auf Speicherbereiche



geld = guthaben;



Arrays

Ein König will zu seinem Geburtstag alle 100 Gefangenen, die in Einzelzellen (von 1 bis 100 durchnummeriert) sitzen, freilassen. Dazu schickt er einen Boten, der alle Türen aufschließen soll.

Während der Bote den Befehl ausführt, kommen dem König Bedenken. Er schickt einen zweiten Boten hinterher, der jede zweite Tür (d.h. die Türen 2, 4, 6, usw.) wieder schließt. Bei jedem "Schließvorgang" wird eine Tür entweder aufgeschlossen (wenn zu) oder zugeschlossen (wenn auf). Danach einen dritten Boten, der jede dritte Tür (also 3, 6, 9, ...) weiterschließt, usw. bis zum 100. Boten.

- Welche Türen sind am Schluss offen? Erstelle ein Programm, das diese Frage beantwortet. Lässt sich das Ergebnis erklären?
- Überlegen zunächst, wie du die Zellen und ihren Zustand modellieren möchtest.
- Beginne zunächst mit einer Methode, die den ersten Boten repräsentiert, dann einer, die den zweiten Boten repräsentiert. Verallgemeinere dann die Methode, dass sie den i -ten Boten darstellt.

