Tutorium 2

26. Oktober 2009

Organisatorisches

Hausaufgaben:

- Abgabe der 1. Hausaufgabe (online UND offline)
- Abgabe der schriftlichen Anmeldung
- Anmeldung bei QISPOS (30.11.)
- Abgabe ab Blatt 2 in Gruppen (onlne UND offline)
- Abgabefrist immer am Ende des jeweiligen Tutoriums
- Ab jetzt Programme kommentieren

Organisatorisches

Sonstiges:

- Gruppeneinteilung
- Betreute Rechnerzeiten: Mittwoch 10-19 Uhr im TEL 206 (Tutoren im TEL 207)
- Interaktives Java Tutorial von Bradley Kjell:
  `http://gailer-net.de/`
- Heute: Kapitel 8, 9A, 9B, 46

Outline

Primitive Datentypen - Wozu braucht man Datentypen?

Datenverarbeitung beim Menschen:

1. Bedeutung der Daten muss zur Verarbeitung bekannt sein.
2. *Beispiel*: Was bedeutet `X`?
   - Römische Zahl 10?
   - Mathematische Variable?
   - Mathematische Multiplikation?
   - Buchstabe `X`?
   - Kreuz?

$\Rightarrow$ Ohne Kontext ist Bedeutung von `X` unklar.

Primitive Datentypen - Wozu braucht man Datentypen?

Datenverarbeitung beim Rechner:

- Rechner verarbeitet Bitmuster der Form `0100 0001 ...`
- Bitmister codieren Buchstaben, Zahlen, Audio- und Videodaten, usw
- *Beispiel*: Was bedeutet `0100 0001 ...` für den Rechner?
  1. Buchstabe `A`?
  2. Zahl 65?
  3. oder etwas anderes?

$\Rightarrow$ Datentyp: Schema, das beschreibt wie eine Folge von Bits verwendet werden soll.

Primitive Datentypen

- Datentyp (Schema) wird vom Prorgammierer festgelegt.
- Wichtige, elementare Datentypen sind in Java eingebaut:

Typ	**Bedeutung**	**Beispielinitialisierung**
byte	Ganze Zahlen (8 Bit)	`byte x = (byte)2;`
short	Ganze Zahlen (16 Bit)	`short x = (short)2;`
int	Ganze Zahlen (32 Bit)	`int x = 2;`
long	Ganze Zahlen (64 Bit)	`long x = 2L;`
float	Gleitkommazahlen (32 Bit)	`float x = 1.2f;`
double	Gleitkommazahlen (64 Bit)	`double x = 2.4;`
char	Einzelnes Zeichen	`char x = 'i';`
boolean	Wahrheitswert	`boolean x = true;`

Primitive Datentypen

Bem.:

- Primitive Datentypen sind elementare Komponenten, um komplexere Datentypen zu erstellen.
- Groß- und Kleinschreibung beachten (`Int` $\neq$ `int`)
- Wir verwenden meistens `int`, `double`, `boolean`.

Beispiel:

- Ist Bitmuster 0000000001100111 vom Datentyp `short`, dann wird die Zahl 113 dagesetellt.
- Ist Bitmuster 0000000001100111 vom Datentyp `char`, dann wird der Buchstabe g dargestellt.

Was ist eine Variable?

Eine Variable ist ein Name für eine Speicherstelle, die einen bestimmten Datentyp verwendet, um einen Wert zu halten.

- Name für eine Speicherstelle = leicht zu merkende Addresse
- Datentype legt Größe und Interpretation der Speicherstelle fest
- Speicherstelle hält zu verarbeitenden Wert

Deklaration einer Variablen

1. Variante 1:

```
typ variablenName;
```

   1. Name und Typ der Variable wird Compiler bekannt gemacht.
   2. Es wird Speicherplatz zur Verfügung gestellt.
   3. Wertzuweisung mit Anweisung `variablenName = wert;`

2. Variante 1:

```
typ variablenName = wert;
```

   1. Wie Variante 1.1-1.2
   2. Der Variablen wird der Anfangswert `wert` zugewiesen.

Deklaration einer Variablen

```
1  class Lohn {
2      public static void main(String[] args) {
3          int arbeitsstunden;
4          double stundenlohn = 10.5;
5          arbeitsstunden = 8;
6          System.out.println("Arbeitsstunden : " + arbeitsstunden);
7          System.out.println("Stundenlohn : " + stundenlohn);
8      }
9  }
```

Bem.:

- Auf Wert einer Variable wird mit Variablennamen zugegriffen
- Ausgabe: + konkateniert Zeichenketten

Zuweisungsanweisungen

- Zuweisungsanweisung:

```
x = ausdruck;
```

  - `x` ist der Name einer bereits deklarierten Variable
  - `ausdruck` ist ein auswertbares syntaktisches Konstrukt
  - = bedeutet Zuweisung

- Beispiel:

```
double x;
x = (10.0 - 2.0)/4.0;
```

- Eine Zuweisungsanweisung erfolgt in zwei Schritten
  1. Auswerten des Ausdrucks (d.h.: berechnen des Werts).
  2. Speichern des Werts in der Variablen.

Arithmetische Operatoren

Operator	Bedeutung	Präzedenz	Ausdruck	Auswertung
-	unäres Minus	höchste	-7	-7
$+$	unäres Plus	höchste	$+2$	2
*	Multiplikation	mittel	$2 * 4$	8
/	Division	mittel	$6/3$	2
%	Modulo	mittel	$11\%3$	2
$+$	Addition	niedrig	$2 + 5$	7
-	Subtraktion	niedrig	$3 - 1$	2

Auswertung und Zuweisung

Beispiel:

```
double x = 2.0;
double y = 4.0;
y = ((1.0 + 2.0)*x - (3.0-1.0)*y)/2.0;
```

- Ersetze Variablen `x` und `y` durch ihre Werte
- Werte Ausdruck aus
- Weise Ergebnis der Variablen `y` zu.

Auswertung und Zuweisung

Beispiel: Berechne $y = x^3 + 2x + 1$

```
class Cubic {
    public static void main(String[] args) {
        double x = 2.0;
        double y = x*x*x + 2*x + 1;
        System.out.println("y(" + x +")  = " + y);
    }
}
```

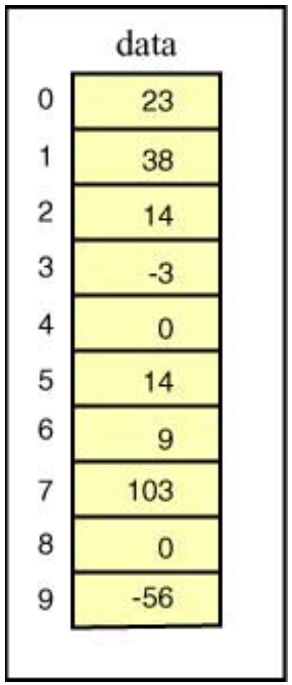
Was ist ein Array?

Ein Array ist ein Objekt mit folgenden Eigenschaften:

1. Arrayobjekt bildet zusammenhängenden Speicherblock
2. Speicherblock ist unterteilt in n Slots
3. Anzahl n der Slots ist die *Länge* eines Arrays.
4. Slots sind sequentiell von 0 bis $n-1$ nummeriert.
5. Jeder Slot enthält einen Wert
6. Alle Werte sind vom gleichen Datentyp.
7. Zugriff auf Slot erfolgt über seinen Index (aus $\{0, \ldots, n-1\}$).

Was ist ein Array?

Beispiel:



Deklaration von Arrays

1. Deklaration
   - Variante 1: `typ[] name;`
     1. Variablenname wird Compiler als Array mit Elementen vom Typ `typ` bekannt gemacht.
     2. Es wird kein Array konstruiert (kein Speicherblock zugewiesen)
   - Variante 2: `typ[] name = new typ[n];`
     1. Variablenname wird Compiler als Array mit Elementen vom Typ `typ` bekannt gemacht.
     2. Es wird ein Arrayobjekt konstruiert, das `n` Slots enthält.
     3. Slots werden mit einem Defaultwert belegt.
   - Variante 3: `typ[] name = {wert_1, ..., wert_n};`
     1. Wie Variante 2.
     2. Slots werden mit den Werten `wert_1, ..., wert_n` belegt.

Wertzuweisungen

- Deklaration und Erzeugung:

```
typ[] name  = new typ[n];
```

- Zuweisung:

```
name[index] = wert;
```

1. Array wurde vorher mit `new typ[n]` erzeugt
2. `index` $\in \{0, \ldots, n-1\}$
3. `wert` ist vom Typ `typ`

Beispielprogramm

Beispielprogramm: `Umsatz.java`

```
1  class Umsatz {
2
3      public static void main(String[] args) {
4          double[] umsatz = {100.34, 98.23, 45.20, 110.76};
5          System.out.println("Laenge d. Arrays " + umsatz.length);
6          System.out.println("1. Quartal: " + umsatz[0]);
7          System.out.println("2. Quartal: " + umsatz[1]);
8          System.out.println("3. Quartal: " + umsatz[2]);
9          System.out.println("4. Quartal: " + umsatz[3]);
10     }
11 }
```