



Java-Programmieren (Teil 1)

Themenüberblick

1. Grundlegende Unterschiede zu C

- ◆ Funktionen
- ◆ main()-Funktion
- ◆ printf()-Funktion
- ◆ Datentypen
- ◆ Arrays
- ◆ Eingabe über Konsole
- ◆ Umwandlung Zahl ↔ String

1. Grundlegende Unterschiede zu C

Grundraster-Programm

```
Test1.java ✕
1 public class Test1 {
2
3     static int add (int a, int b)
4     {
5         int d = a+b;
6         /*
7         int x = 42;
8         d = x;
9         */
10        return d;
11    }
12
13    public static void main(String[] args)
14    {
15        int a = 3;
16        int b = 2;
17        int c = add(a,b);
18        System.out.printf("c=%d",c);
19    }
20
21 }
```

Grundraster-Programm

```
1 public class Test1
2
3     static int add (int a, int b)
4     {
5         int d = a+b;
6         /*
7         int x = 42;
8         d = x;
9         */
10        return d;
11    }
12
13    public static void main (String[] args)
14    {
15        int a = 3;
16        int b = 2;
17        int c = add(a,b);
18        System.out.printf("c=%d", c);
19    }
20
21 }
```

Der Dateiname muss mit dem Namen der Klasse übereinstimmen

Klasse Test1

public:
Objekt ist für alle Klassen sichtbar.

static:
Objekt gehört zur Klasse und nicht zu einer Instanz von ihr

Eingabeparameter

Methode add mit Rückgabotyp **int**

int-Variable d mit Wert a+b

Kommentare werden vom Compiler ignoriert.

Rückgabewert der Methode add

Eingabeparameter: Optional können über die Konsole Parameter an main übergeben werden.

Methode main mit Rückgabotyp **void** (kein Rückgabewert)

Variablen a und b erstellen mit a = 3 und b = 2

Variable c mit add (a, b) belegen. Dazu wird die Methode add aufgerufen.

printf-Ausgabe des Programms: c=5

Ausgabe auf Konsole

- `System.out.printf(Formatstring, Variable 1, ... Variable n)`
 - ◆ Wie in C, nur mit `System.out`
 - ◆ `System.out.printf("Ich bin %d Jahre alt", age);`
 - ◆ Teilweise andere Formatzeichen als in C
- `System.out.print(String s)`
 - ◆ Keine Formatierung der Strings möglich
- `System.out.println(String s)`
 - ◆ `print` mit `\n` am Ende des Strings

Datentypen in Java

- Viele Datentypen genauso wie in C:

<i>Datentyp</i>	<i>Name</i>	<i>Formatzeichen</i>
◆ Integer	int	%d
◆ Long	long	%d
◆ Short	short	%d
◆ Double	double	%f
◆ Float	float	%f
◆ Character	char	%c

boolean

- Wahrheitswerte
- Wertebereich : {**true**, **false**}
- Name : **boolean**
- Formatzeichen : **%b**

```
1 public class Test2
2 {
3     public static void main(String[] args)
4     {
5         boolean a = true;
6         boolean b = false;
7         boolean c = a && b;
8         System.out.printf("%b und %b = %b", a, b, c);
9     }
10 }
11
```

String

- Zeichenketten (eigener Datentyp in Java)
- Wertebereich : Beliebig lange Zeichenketten
(nur eine Frage des Arbeitsspeichers)
- Name : `String`
- Formatzeichen : `%s`
 - ◆ Kann auch für andere Datentypen verwendet werden, um sie als String darzustellen, wie z.B. bei `printf`

Konkatenation von Strings

- Strings können in Java leicht miteinander verbunden werden

```
String str = "Hallo";  
str = str + " Peter!\n";  
str = str.concat("Wie geht es dir?");  
String str1 = str;  
System.out.println(str1);
```

Zugriff auf einzelne Zeichen im String

- In C ist jeder String ein char-Array
 - ◆ Vorteil: Direkter Zugriff auf einzelne Zeichen möglich
- Java bietet aber Methoden zur Konvertierung zwischen Strings und char-Arrays
 - ◆ `char[] String.toCharArray()`
 - ◆ `new String(char[] chararray)`

String \Leftrightarrow char-Array (Beispiel)

```
1  public class Test2
2  {
3      public static void main(String[] args)
4      {
5          String str = "Ausgabe 1";
6          char str1[] = str.toCharArray();
7          str1[8] = '2';
8          str = new String(str1);
9          System.out.print(str);
10     }
11 }
12
```

Weitere nützliche String-Operationen

- **int** `String.getLength()`
 - ◆ Gibt die Länge des Strings als int zurück
- **boolean** `String.equals(String str)`
 - ◆ Gibt true zurück, wenn beide Strings gleich sind, sonst false.
 - ◆ Geeignet für if-Abfragen

Danke für die Aufmerksamkeit!