

Handout zum Programm "Rechenprogramm.java"

- Einzeilige Kommentare beginnen mit: `"/"`
- Mehrzeilige Kommentare beginnen mit `"/"` und enden mit `"/"`
- Zusätzliche Funktionalität können wir mit `"import"` laden. Hier wird die Klasse `"java.math.*;"` hinzugeladen, die viele Funktionen enthält.
- `"public class Rechenprogramm"` bezeichnet den Anfang einer Klasse
- `"{"` bezeichnet den Anfang eines Blockes, zum Beispiel den Anfang einer Methode. Jeder Block der angefangen wird, muss am Ende auch wieder mit `"}"` geschlossen werden! Wird dies vergessen erzeugt dies einen Fehler.
- `"public static void main(String [] argumente)"` bezeichnet den Punkt, an dem unser Programm startet. Java Programme starten immer mit der `"main"` Methode!
- `"double a, b, c ;"` hier werden drei Variablen erzeugt (deklariert). Es werden drei Begriffe (Variablen) geschaffen (deklariert) unter denen Daten abgelegt werden können. Daten können verschiedene Typen haben. So gibt es ganze Zahlen (`int`), Kommazahlen (`double`), Text (`String`) und noch viele weitere Datentypen.
- `"int"` steht für ganze Zahlen, Nachkommastellen werden verworfen
- `"double"` steht für Kommazahlen
- `"String"` steht für Text
- Wie ich den Namen der Begriffe wähle bleibt mir überlassen! Ich darf nur keine reservierten Begriffe wählen, d.h. Begriffe die bereits eine andere Funktion haben.
- Jede Anweisung wird mit einem `";"` abgeschlossen
- `"a = 0;"` hier wird die Zahl 0 unter dem Begriff (Variable) `"a"` abgespeichert.
- `"System.out.println("Die erste Zahl lautet: " + a);"` Hier gebe ich den Wert aus, den ich unter dem Begriff (Variable) `"a"` gespeichert habe. Texte werden immer mit Hochkomma `""` umschlossen, während Zahlen nicht mit Hochkomma umschlossen werden! Das Plus Zeichen `+` fügt in diesem Fall einen Text `"Die erste Zahl lautet: "` und den Wert der Variable `"a"` zu einem Text zusammen. Dieser zusammen gefügte Text wird dann an die Methode `"println"` der Klasse `"out"` geliefert (übergeben). Die Methode `"println"` sorgt dann dafür, dass der Text (`String`) in der Kommandozeile ausgegeben wird.
- `"a = b+c;"` hier fügt das Plus Zeichen `+` keine Texte zusammen, sondern zwei Zahlen.
- Die Begriffe (Variablen) `"a"`, `"b"` und `"c"` haben hier in diesem Programm als Datentyp `int` bzw. `double` und sind daher Zahlen.

Beispiel:

`"int b = 10;"` die Variable `b` erhält den Wert 10

`"int c = 5;"` die Variable `c` erhält den Wert 5

`"a = b+c;"` `b` plus `c` ergibt als Ergebnis 15

Dieses Ergebnis speichern wir hier unter dem Begriff (Variable) `"a"`
`a` hat also den Wert 15

`"a = b/c;"` für eine Division wird `"/"` verwendet.

Unter `a` wird 2 gespeichert (Berechnung `10:5 = 2`)

`"a = b*c;"` für eine Multiplikation wird `"*"` verwendet.

unter `a` wird 50 gespeichert (Berechnung `10x5 = 50`)

Folgende Tastenkombinationen vereinfachen das Arbeiten mit Texten:

`"Strg" + "c"` kopiert den markierten Text in einen Zwischenspeicher

`"Strg" + "v"` fügt den Inhalt aus dem Zwischenspeicher an anderer Stelle wieder ein

`"Strg" + "a"` markiert den gesamten Text

`"Strg" + "f"` öffnet ein Suchfenster zum Suchen nach Texten

Klasse:

Eine Klasse ist im wesentlichen eine Sammlung von Methoden. So enthält zum Beispiel die Klasse "Math" mathematische Methoden.

Klassennamen und Methodennamen sollten so gewählt werden, so dass man die grundsätzliche Funktion am Namen ablesen kann.

Dies hat den Vorteil, dass Programme dadurch übersichtlicher werden.

Methoden:

Methoden und Funktionen meinen das gleiche. Eine Methode funktioniert so, dass man ihr beim starten (aufrufen) Daten geben kann, die dann von der Methode verarbeitet werden. Nach der Verarbeitung wird ein Wert zurückgegeben.

Wichtig zu merken ist, dass man einer Methode immer beliebig viele Daten geben kann, die Methode aber immer nur ein oder kein Ergebnis zurückliefert.

```
// Dies ist eine Methode, die einen Wert zurückliefert
```

```
private static int addiere(int a, int b)  
{  
    int c = a+b;  
    return c;  
}
```

"private" sagt aus, dass diese Methode nur innerhalb dieser Klasse verwendet (aufgerufen) werden kann.

"static" sagt aus, dass ich diese Methode direkt aufrufen kann (wir lernen später noch andere Formen kennen)

"int" sagt aus, dass die Methode "addiere" ein Ergebnis mit dem Datentyp Integer (Ganzzahl) zurückliefert.

"void" sagt aus, dass die Methode kein Ergebnis zurückliefert.

"addiere" ist der Name der Methode, über den ich die Methode aufrufen kann.

"(int a, int b)" bezeichnet die Daten (Parameter), die der Methode zum verarbeiten gebe (übergeben).

"int c = a+b;" hier addieren wir die Daten, die unter "a" und "b" abgespeichert sind. Das Ergebnis wird unter "c" abgespeichert.

"return c;" mit return beenden wir die Methode und liefern als Ergebnis den Wert der Variablen "c" zurück.

```
// hier wird die Methode gestartet (aufgerufen)
```

```
int ergebnis3 = addiere(2, 3);
```

"int ergebnis3" hier erzeugen (deklarieren) wir eine Variable unter der wir einen Wert vom Datentyp Integer abspeichern können. Unter dem Begriff (Variable) "ergebnis3" wird das Ergebnis (Rückgabewert) der Methode "addiere" gespeichert.

"addiere(2, 3)" hier wird die Methode "addiere" aufgerufen und es werden zwei Werte (Parameter) übergeben.

Variable:

Statt mit Zahlen direkt zu arbeiten, verwendet man beim Programmieren Variable, d.h. Begriffe unter denen Daten abgelegt werden.

Namen von Variablen sollten so gewählt werden, so dass man den grundsätzlichen Inhalt und eventuell auch den Datentyp am Namen ablesen kann.

Datentyp:

Daten können verschiedene Datentypen haben. So gibt es ganze Zahlen, Kommazahlen, Texte und noch viele weitere Datentypen

"int" erzeugt eine Variable vom Typ Integer (Ganzzahl)

"double" erzeugt eine Variable vom Typ Double (Kommazahl)

"String" erzeugt eine Variable vom Typ String (Text)