| Syntax | Kommentar | Beispiel |
| --- | --- | --- |
|  |  |  |
| Variablen | | |
| Datentyp *variablenName*; | Deklaration | int *x*; |
| *variablenName* = wert; | Initialisierung | *x* = 134; |
| Datentyp *variablenName* = wert; | Definition | boolean *y* = true; |
|  |  |  |
| Wichtige Datentypen | | |
| int | Ganzzahlen | 0, -1, 1, 42, 69 |
| long | Ganzzahlen | 2, 5, 3456895321983 |
| float | Kommazahlen | 1.4f, 4.9f, .3f |
| double | Kommazahlen | 42.6, 33.2, .7 |
| boolean | Wahrheitswerte | true, false |
| char | Zeichen | 'a','+','$','<','0' |
| String | Zeichenketten | "Hi", "fp8$er#", "5" |
|  |  |  |
| Wichtige Operatoren | | |
| Arithmetische Operatoren |  |  |
| + | Addition | 3 + 5 |
| - | Subtraktion | 9 – 1 |
| \* | Multiplikation | 4.4 \* 3.0 |
| / | Division | 2.8f / 1.4f |
| ++ | Inkrement  Abk. für: x = x + 1; | *x*++;  ++*x*; |
| -- | Dekrement  Abk. für: x = x – 1; | *x*--;  --*x*; |
| Vergleichende Operatoren |  |  |
| == | Gleich | 3 + 5 == 2 + 4 |
| != | Ungleich | 1.3f != 3.0f |
| > | Größer | 44 > 10 |
| < | Kleiner | 91.0 < 133.43 |
| >= | Größer gleich | 30 >= 22 + 8 |
| <= | Kleiner gleich | 12 <= 56.5f |
| Logische Operatoren |  |  |
| ! | Nicht | !false |
| && | Und | true && true |
| || | Oder | true || false |

|  |  |
| --- | --- |
| Syntax | Beispiel |
|  |  |
| Abfragen | |
| if (Bedingung) {  Code;  } | if (true) {  System.*out*.*println*("Hi!");  } |
| if (Bedingung) {  Code;  } else {  Code;  } | double *area*;  if (*r* > 0) {  *area* = 3.14159 \* *r* \* *r*;  } else {  *area* = 0.0;  } |
|  |  |
| Schleifen | |
| for-Schleife |  |
| for (Initialisierung; Bedingung; Iteration) {  Code;  } | for (int *i*=1; *i*<=10; *i*=*i*+1) {  System.*out*.*println*(*i*);  } |
| while-Schleife |  |
| while (Bedingung) {  Code;  } | int *x* = 2;  int *y* = 1;  int *sum* = *y*;  while (*x* > 0) {  *sum* = *sum* + 1;  *x* = *x* – 1;  } |
| do-while-Schleife |  |
| do {  Code;  } while (Bedingung); | double *age* = 0.0;  do {  *age* = *age* + (18 - *age*) / 2;  } while (*age* < 18); |
|  |  |
| Arrays | |
| Datentyp[] *variablenName*; | char[] *characters*; |
| *variablenName* = new Datentyp[länge]; | *characters* = new char[3]; |
| Datentyp[] *variablenName* = new Datentyp[länge]; | int[] *numbers* = new int[2]; |
| Datentyp[] *variablenName* = {wert, wert, wert}; | String[] *words* = {"Yo", "Hi"}; |
| *variablenName*[index] = wert; | *numbers*[0] = 2;  *numbers*[1] = 3;  System.*out*.*println*(*words*[1]); |
| Syntax | |
| Beispiel | |
|  | |
| Klassen | |
| public class KlassenName {  public Datentyp *attributName*;  public Datentyp *attributName* = wert;  ...  public *KlassenName*(...) {  KonstruktorCode;  }  public Datentyp *methodenName*(...) {  Code;  }  ...  } | |
| public class Profile {  public String *name*;  public int *age* = 0;  public *Profile*(String *name*, int *age*) {  this.*name* = *name*;  this.*age* = *age*;  }  public void *birthday*() {  this.*age*++;  }  } | |
|  | |
| Methoden | |
| public Datentyp *methodenName*(Datentyp *parameterName*, Datentyp *parameterName*) {  Code;  } | |
| public void *outputToConsole*(String *line*) {  System.*out*.*println*(*line*);  } | |
| public boolean *equalNumbers*(int *x*, int *y*) {  return (*x* == *y*);  } | |
| Beispielaufruf:boolean resultat = *equalNumbers*(4, 3 + 1); | |