Objekt

Objekt erstellen

Ein Objekt ist ein konkretes Exemplar, das erstellt wurde.

In Java: new < Klassenname > ();

Bsp: new Viereck();

Klasse

Klassendefinition

Ein Bauplan für Objekte. Gleiche Attribute, gleiche Methoden.

In Java: class < Klassenname > { /... }

Bsp: class Viereck { /... }

Methode

Methodendefinition

Eine schlummernde Fähigkeit. Das Objekt kann etwas tun.

In Java:

void < Methodenname > () { /... }

Bsp: void Drehen () { /... }

Methodenaufruf

Eine Fähigkeit soll ausgeführt werden. Das Objekt tut etwas.

In Java:

< Objektname > . < Methodenname > () ;

Bsp: viereck1.Drehen();

Attribut

Attributdeklaration

Ein Merkmal eines Objekts. Speichert Informationen zu dem Objekt.
Deklaration = Definition.

In Java:
< Datentyp > < Attributname >;

Bsp: int groesse;

Übergabewert / Parameter

Methodenaufruf

Ein Wert, den eine Methode beim Aufruf übergeben bekommt.

Bsp: GeheSchritte(5) oder SetzeFarbe(Grün)

In Java ist der Übergabewert eine lokale Variable.

In Java: void < Methodenname > (< Datentyp > < Variablenname >){ /... }

Bedingte Anweisung

Etwas wird nur ausgeführt, wenn eine Bedingung wahr ist. Es kann auch eine Alternative angegeben werden, die nur ausgeführt wird, wenn die Bedingung falsch ist.

In Java: if (< Bedingung >) { /... }
else { < Alternative >}

Bsp: if (viereck.farbe==gruen) {Drehen();}

Bedingungen verknüpfen

Zwei Wahrheitswerte können mit UND oder ODER verknüpft werden.

In Java: && für UND ; || für ODER

Bsp: "Liegt x zwischen 3 und 10?"

x < 10 && 3<x

Bedingte Wiederholung

Etwas wird wiederholt ausgeführt, solange eine Bedingung erfüllt ist. Nach jedem Durchlauf wird die Bedingung überprüft.

In Java: while (< Bedingung >) { /... }

Bsp: while (NichtVorWand()) {
SchrittGehen();}

Bedingung

Vergleiche

Eine Bedingung ist ein Wahrheitswert,
also true oder false.

Ein Vergleich ist die einfachste Bedingung.

Bsp: $5 < 4$
$90 < x$

In Java: Ungleich (!=), Istgleich (==), Kleiner (<),
Größer (>), Kleinergleich (<=), Größergleich (>=)

Datentyp

Eine Klassifizierung von Werten.

Es gibt Text (z.B. „Hallo!“), Zahl (z.B. 42), Wahrheitswert (z.B. true) und Zeichen (z.B. ‚c‘).

In Java: int (Ganzzahl), double (Kommazahl), String (Text), char (Zeichen), boolean (Wahrheitswert).

Funktion /

Methode mit Rückgabewert

Eine Methode, die einen Rückgabewert hat.

In Java: Schreibe den Datentyp des Rückgabewerts statt void!
UND return < Rückgabewert >;

Bsp: int EinsGeben(){
return 1;}

Wertzuweisung

Zuweisung

Eine Variable/Ein Attribut erhält einen Wert. Der Wert muss zum Datentyp passen.

In Java: < Variable > = < Wert >;

Bsp: groesse = 200;

lokale Variable

Deklaration einer Variablen

Kann Werte speichern. Eine lokale Variable wird innerhalb einer Methode definiert und ist danach nicht mehr verfügbar.

In Java: < Datentyp > < Attributname >;

Bsp: int groesse;

Konstruktor

Eine Methode, die Objekte der Klasse erstellt. Wird für Erstzuweisungen verwendet.

In Java: < Klassenname > () {/...}

Bsp: Viereck(){
groesse=100;}

Unterklasse

Eine Klasse, die Zugriff auf alle Attribute und Methoden einer anderen Klasse, der Oberklasse, hat.

In Java: class < Klassenname > extends < Name der Oberklasse > { /... }

Bsp: class Papagei extends Vogel { /... }

Methodenaufruf auf der Oberklasse

Konstruktoraufruf auf der Oberklasse

Eine überschriebene Methode kann auf die passende Methode der Oberklasse zugreifen.

In Java: super.< Methodenname > ();

super(); ruft den Konstruktor der Oberklasse auf.

Überschreiben

Definiert man eine Methode, die es in der Oberklasse schon gibt, neu, wird bei jedem Aufruf die Methode der Unterklasse aufgerufen.

Das nennt man Überschreiben.

ArrayList

Feld

Objekte gleicher Klasse kann man in einer Liste der Klasse ArrayList verwalten.

In Java: new ArrayList < Klassenname >();

folgende Methoden sind möglich:

size() – remove(i) – contains(e) – remove(e) – get(i) – add(e) – set(i,e)

ArrayList

durchlaufen

Idee: Zähle von 0 bis zur Länge der ArrayList - 1

Bsp in Java:

```
for (int i = 0; i < Autos.size(); i++){
 Autos.get(i).FahrernamenGeben() ==
 „Günther“;}
```

Kapselung

Der Zugriff auf Attribute und Hilfsmethoden einer Klasse wird eingeschränkt.

In Java: public < Methode >

private < Attribut >

Bsp: public void Drehen() { /...}

Referenzen

Eine Referenz speichert einen Verweis auf ein Objekt. Die Referenz kann leer sein.

In Java: Statt dem Datentyp steht die Klasse des referenzierten Objekts.

Bsp: Viereck viereck1;
viereck1 = new Viereck();

for each Schleife

Vereinfachte Iteration über Listen.

Idee: Definiere eine lokale Variable als Referenzattribut mit dem aktuellen Objekt der Liste. Die Schleife nimmt dann für jeden Durchlauf das nächste Objekt in der Liste für das Referenzattribut.

In Java: for(< Klasse > < Name RefAtt > :
< Name ArrayList >){ /... }

Bsp: for (Auto a : Autos){
a.FahrernamenGeben() == „Günther“;}

Methodenaufrufe

auf Referenzen

Eine Referenz verwaltet ein Objekt. Gibt eine Methode eine Referenz zurück, ist das wieder ein Objekt -> Objekt.Methode.Methode.

Bsp in Java: Haus haus;
(mit der Methode Fenster Fenster(){/...})

haus.Fenster().Aufmachen()
haus.Klingeln();