GYMNASIUM MIT SCHÜLERHEIM PEGNITZ
Wilhelm-von-Humboldt-Str. 7
91257 Pegnitz
Tel. 09241/48333 Fax. 09241/2564
sek@gympeg.de www.gympeg.de

Grundwissen Informatik JS 10 — 13. November 2018

Grundlagen der Objektorientierung

1. a) Wozu dienen Klassen- und Objektkarten?

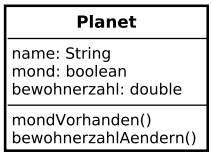
   b) Beschreibe das Aussehen, den Inhalt und Schreibkonventionen bei Klassenkarten.

   c) Beschreibe das Aussehen, den Inhalt und Schreibkonventionen bei Objektkarten.

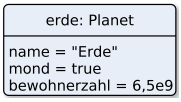
Lösung

a) Um Klassen und Objekte möglichst übersichtlich zu beschreiben nutzt man Klassen- und Objektkarten.

b) Scharfe Ecken, Klassenname groß, Attribute klein geschrieben mit Datentyp (Doppelpunkt dazwischen); Methoden (klein geschrieben) stets mit runden Klammern.



c) Runde Ecken, Objektname klein mit dazugehöriger Klasse (Doppelpunkt dazwischen), den Attributen werden durch „=“ konkrete Attributwerte zugewiesen; keine Methoden.



2. Erkläre die Begriffe „Klasse“, „Attribut“, „Attributwert“, „Objekt“, „Methode“ und „Dienst“. Beispiel!

Lösung

Allgemein	Beispiel
Eine **Klasse** ist eine **Bauanleitung** (Oberbegriff) **für gleichartige Objekte mit gleichen Attributen**.	Klasse `Konto` mit den Attributen `KontoNummer`, `Guthaben`, ...
Objekte haben einen eindeutigen Namen und haben ggf. verschiedene **Attributswerte**.	Das Konto `konto07` mit `KontoNummer=4711` und `Guthaben=1000`.
Attributswerte werden mit **Methoden** (oder **Diensten**) verändert.	Die Methode `abheben()` verändert den Wert des Attributs `Guthaben`.

3. a) Gib einige grundlegende Datentypen von Java und ihre Bedeutung an.

   b) Welche Syntaxregeln müssen bei Datentypen eingehalten werden?

   c) Warum sind Datentypen in Programmiersprachen nötig?

Lösung

a)

Typ	Beschreibung	Literale
int	ganze Zahl	Bereich ca. ±2 Mrd
boolean	boolescher Wert	true, false
double	Gleitkommazahl	ca. $-10^{308} \ldots 10^{308}$
char	Zeichen	'a','b','c',')'
String	Zeichenkette	“Hallo Welt“
...		

b)
- primitive Datentypen werden klein geschrieben, Objekttypen (z.B. `String`) groß.
- Attributwerte vom Datentyp `char` / `String` müssen in einfache / doppelte Hochkommata einschlossen werden.

c) Abhängig vom Datentyp einer Variable muss entsprechend Platz im Datenspeicher reserviert werden.

4. a) Erkläre an einem Beispiel, wie der Aufbau einer Java-Klasse aussieht. Welche Notationskonventionen gibt es in Java?

   b) Was versteht man unter Datenkapselung, wozu ist sie gut?

Lösung

a)
```
1 public class Eintrittskarte
2 {   // erst Attribute festlegen
3     private String opername;
4     ...
5     //dann Methoden festlegen
6 }
```

- Klassennamen werden groß geschrieben, Attribut- und Methodennamen klein.
- Nach jeder { wird der Programmcode weiter eingerückt, nach jeder } wieder ausgerückt.
- Kommentare nach // oder innerhalb von `/**` und `*/` erleichtert die Lesbarbeit des Codes.

b) Attribute werden stets `private` deklariert um sie vor unerlaubtem Zugriff von außen zu schützen. Der Zugriff auf die Attribute erfolgt ausschließlich über Getter- und Settermethoden.

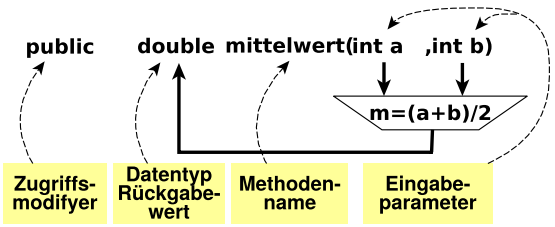
5. a) Erkläre genauer was man unter einer Methode versteht.

   b) Erkläre an einem Beispiel, welchen formalen Aufbau eine Methodensignatur in Java hat.

Lösung

a) Eine Methode führt gewisse Dienste einer Klasse aus. Genauso wie eine mathematische Funktion kann sie aus bestimmten Eingabeparametern bestimmte Werte bestimmen oder Tätigkeiten ausführen.

b) Beispiel public double Mittelwert(int a, int b)



6. a) Was versteht man unter einer Wertzuweisung? Beispiele!

   b) Worin besteht der Unterschied zwischen einer Wertzuweisung und einem mathematischen „=“-Zeichen?

Lösung

a) Bei einer Wertzuweisung der Form `A = B` wir der Variable mit der Bezeichnung A der Inhalt der Variable mit der Bezeichnung B zugeordnet.

```
name = ‘‘Sepp‘‘; // name erhält den Wert Sepp
x = 3 + 4; // x erhält den Wert 7
x = x + 1; // x wird um 1 erhöht
```

b)
- Auf der linken Seite darf nur ein Attribut stehen,
- vertauscht man die beiden Seiten, erhält man ggf. ein anderes Ergebnis (oder eine Fehlermeldung)
- Gleichungsketten sind nicht zugelassen.

7. a) Was versteht man unter einem Konstruktor, welche Aufgaben hat er?

   b) Welchen Namen besitzt er, wie wir er aufgerufen, wie viele Konstruktoren darf eine Klasse besitzen, wo sind sie im Programmcode zu finden?

Lösung

a) Der Konstruktor einer Klasse ist eine **Methode zur Erzeugung eines Objektes** und wird mit dem Befehl new(...) aufgerufen. Dabei werden **alle Attribute initialisiert** bestimmte Anfangswerte zugewiesen.

b)
- Jede Klasse kann beliebig viele (auch keine) Konstruktoren haben, sie **tragen alle den Namen der Klasse** müssen sich aber in der Parameterliste unterscheiden.
- Ist kein selbst geschriebener Konstruktor vorhanden, verwendet Java einen Standardkonstruktor, der alle Attribute auf `null` setzt.
- Die Konstruktoren werden nach der Attributdeklaration als erste Methoden der Klasse aufgeführt.

---

8. Erkläre an einem Beispiel, wie ein Konstruktor mit zwei Übergabeparametern implementiert werden kann.

```
1 public class Konto{
2     private int kontoNummer;
3     private double guthaben;
4     private String kunde;
5     private boolean
          kreditkarte;
6 }
```

Lösung

```
1     public Konto(String neuerKunde
          ,int neueKNr){
2         kunde=neuerKunde;
3         kontoNummer=neueKNr;
4         guthaben=0;
5         kreditkarte=false;
6     }
```

---

9. a) Was versteht man unter lokalen Variablen, was unterscheidet sie von Attributen?

   b) Welchen Vorteil haben lokale Variablen gegenüber den global definierten Attributen?

Lösung Lokale Variablen...

a) werden in einzelnen Methoden ohne Modifizierer deklariert.

b) Sie belegen nur während der Ausführungszeit einer Methode Speicherplatz und dürfen in verschiedenen Methoden den gleichen Namen tragen, erzeugen dadurch übersichtlicheren Programmcode.

```
1 public class Kugel{
2     private double radius;
3     public double oberflaecheGeben(){
4         double oberflaeche;
5         oberflaeche = 4/3*3.1414*
              radius*radius*radius;
6         return oberflaeche;
7     }
8 }
```

10. a) Wie lauten die Befehle zur Ausgabe von Text in einem Textfenster, worin unterschieden sie sich, warum sind sie so lang?

    b) Welche Datentypen dürfen den Befehlen übergeben werden? Nenne einige typische Beispiele für die Verwendung der Bildschirmausgabe.

Lösung

a) `System.out.print(daten)` ohne Zeilenumbruch bzw `System.out.println(daten)` mit Zeilenumbuch am Ende. Viele Java-Befehle sind in sogenannte Bibliotheken ausgelagert. `print(ln)` befindet sich in der Bibliothek `System.out`. Damit der Compiler weiß, in welcher Bibliothek er den Befehl findet, muss der Bibliotheksname mit angegeben werden.

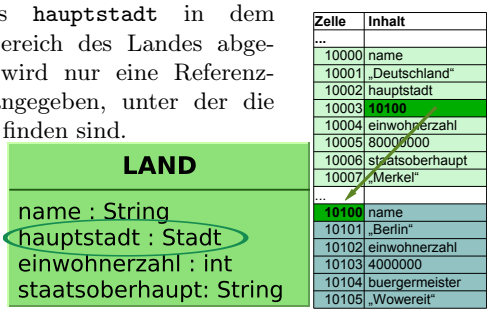
b) Es dürfen nur einfache Datentypen und `Strings` übergeben werden. Objektdatentypen nicht.

- `System.out.println(“Hallo “ + name)`
  ⇒ `Hallo Peter`
- `System.out.println(“Kontostand: “ + betrag + “ Euro“)`
  ⇒ `Kontostand: 500 Euro`

11. Erkläre an einem Beispiel, was man unter einer Aggregation und unter einem Referenzattribut versteht? Wie werden die Methoden eines Referenzatributes aufgerufen?

Lösung

Die Beziehung der Klassen `STADT` und `LAND` ist eine Aggregation, weil `hauptstadt` ein Attribut vom Datentyp `STADT` ist. Dabei werden aber nicht die Daten des Attributes `hauptstadt` in dem Speicherbereich des Landes abgelegt. Es wird nur eine Referenzadresse angegeben, unter der die Daten zu finden sind.

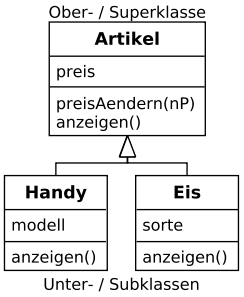


Die Methoden eines Referenzattributes werden mit der Punktschreibweise aufgerufen,
z.B. `hauptstadt.datenAnzeigen()`.

12. Erkläre die Grundprinzipien der Vererbung?

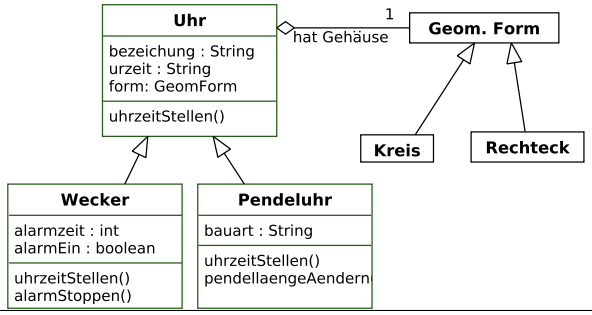
Lösung

- Die Subklassen besitzen (erben) alle Attribute und Methoden der Oberklasse. Die Methoden können aber in den Subklassen überschrieben (neu implementieren) und auf deren Bedürfnisse angepasst werden.
- Will man aus der Unterklasse den Konstruktor der Oberklasse aufrufen, geschieht das mit dem Befehl `super(<P-Liste>)`.
- Will man aus der Unterklasse eine anderen Methode (z.B. `anzeigen()`) der Oberklasse aufrufen, geschieht das mittels `super.anzeigen()`.



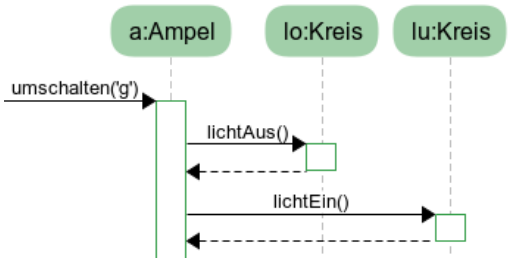
13. Erkläre an einem Beispiel, wie ein Klassendiagramm aufgebaut ist und wozu es nützlich ist. Gehe auch auf die Unterschiede zwischen Aggregation und Vererbung ein.

Lösung Das Klassendiagramm stellt die Beziehungen (Aggregation, Vererbung) zwischen den Klassen dar. Die Aggregation modelliert dabei eine `hat`-Beziehung (Eine Uhr **hat** einen Antrieb) und wird durch eine Raute samt Kardinalität dargestellt, die Vererbung modelliert eine `ist`-Beziehung (Ein Wecker **ist** eine Uhr) und wird durch einen Dreieckspfeil dargestellt.



14. Erkläre an einem Beispiel, wie ein Sequenzdiagramm aufgebaut ist und wozu es nützlich ist.

Lösung Im Sequenzdiagramm verlaufen vertikal die Lebenslinien der Objekte. Horizontal werden die ausgetauschten Anfangsbotschaften (Methodenaufrufe) und Antwortbotschaften (Rückgabewerte) dargestellt. Aktivitätsbalken auf den Lebenslinien zeigen, wenn eine Methode gerade ausgeführt wird.



15. Was versteht man unter einem Algorithmus? Beispiele!

Lösung Ein Algorithmus ist eine Verarbeitungsvorschrift, die aus **endlich vielen, eindeutig ausführbaren** Anweisungen besteht.
Beispiel:

- schriftlicher Multiplikations-Algorithmus
- Computerprogramme
- Kochrezepte und Gebrauchsanweisungen mit Einschränkungen, weil die Anweisungen oft nicht eindeutig sind.

16. Wie werden einfache Programmverzweigungen in Java implementiert? Welche Schreibkonventionen werden eingehalten?

Lösung

```
1 if (Bedingung) {
2     Programmcode falls die
          Bedingung zutrifft
3 }
4 else{
5     Programmcode falls die
          Bedingung nicht zutrifft
6 }
```

- Der `else`-Block kann weggelassen werden.
- Der Programmcode für den `if`- und den `else` - Teil wird jeweils eingerückt, bei verschachtelten Verzweigungen auch mehrfach.
- Das zu einem `if` gehörende `else` beginnt in der gleichen Spalte.

17. a) Was versteht man unter logischen Ausdrücken? Beispiele!

b) Welche Vergleichsoperatoren gibt es bei logischen Ausdrücken für einfache Datentypen und Strings?

c) Mit welchen logischen Grundoperationen könnne logische Ausdrücke zusammengesetzt werden?

Lösung

a) Logische Ausdrücke sind Ausdrücke und Methoden, deren Ergebnis vom Datentyp `boolean` ist, z.B. `x==7`; `x<=6`; `true`; `wasserVorhanden()`.

b)

Name	Term
Gleichheit	A==B
Ungleichheit	A!=B
größer/kleiner	A>B, A<B
größer/kleiner gleich	A>=B, A<=B

c)

Name	Term	wahr wenn
Und	L && M	L und M wahr sind
Oder	L || M	L oder M oder beide wahr
Nicht	!L	L falsch ist.

18. a) Wozu sind `switch-case`-Anweisungen nützlich?

b) Erkläre an einem aussagekräftigem Beispiel, wie eine `switch-case`-Anweisung aufgebaut ist und erläutere ihre Bestandteile

Lösung

a) `switch-case`-Anweisungen werden beim Fallunterscheidungen mit vielen unterschiedlichen Fällen verwendet.

b) Beispiel: Noten in ein Prädikat umwandeln

```
1 switch ( note ) {
2    case 1: nText = "sehr gut"; break;
3    ... //andere Faelle hier ausgelassen
4    case 6: nText = "ungenuegend"; break;
5    default: nText = "Fehler"; break;
6 }
```

- Es sind beliebig viele `case`-Klauseln erlaubt.
- `break` sorgt dafür, dass nicht(!) die nächste `case`-Klausel bearbeitet, sondern ans Ende der `switch`-Anweisung gesprungen wird.
- Die `default`-Klausel ist optional und sammelt alle Fälle, die vorher nicht aufgeführt sind.

19. a) Wie lautet die allgemeine Syntax für eine Bedingte Wiederholung in Java?

b) Gib ein aussagekräftiges Beispiel für eine Bedingte Wiederholung an.

Lösung

a)

```
1 while (<logischer Ausdruck>) {
2         Anweisungen
3 }
```

b)

```
1 // Quadratzahlen ausgeben
2 i=1;
3 while (i<11) {
4   System.out.println(i+" hoch 2 =
        "+(i*i));
5   i=i+1;
6 }
```

20. Gib ein aussagekräftiges Beispiel für eine Wiederholung mittels `for`-Schleife und erkläre die einzelnen Bestandteile.

Lösung

```
1 // Quadratzahlen ausgeben
2 for(int i=1; i<11; i=i+1) {
3   System.out.println(i+" hoch 2 =
        "+(i*i));
4 }
```

`int i = 1;` lokale Zählvariable wird deklariert und ihr Startwert festlegt.

`i<11;` Bedingung, die Zutreffen muss, damit die Scheife „betreten“ wird.

`i=i+1;` Festlegung des Wertes für den nächsten Durchlauf.

21. Was versteht man unter Arrays, wie lauten die dazugehörigen Fachbegriffe? Gib ein typisches Beispiel für die Verwendung von Array an.

Lösung **Arrays** benutzt man um Attribute zu einem Paket mit nummerierten Zellen zusammenzufassen. Die Anzahl der Zellen hießt **Länge des Arrays**. Die Nummer einer Zelle wird auch **Index** genannt. Die erste Zelle besitzt in Java den Index 0.

Z.B. kann man die Schüler einer Klasse zu einem Array `Schuelerliste` zusammfassen, der die Namen enthält.

22. Gib mit Hilfe eines Beispiels an, wie Arrays in Java implementiert werden (zwei Möglichkeiten).

Lösung

- Langform bei unbekannter oder wählbarer Länge

```
1 private double[] zeiten;
2 zeiten = new double[8];
3 zeiten[0] = 13.1;
4 ...
```

- Kurzform bei bekannten Daten und bekannter Länge

```
1 private double[] zeiten = {10.1,
      11.1, 12.3, 12.4, 12.6, 13.0,
       13.5, 15.7};
```

23. Gib mit Hilfe eines Beispiels an, wie Daten, die in einem Array gespeichert sind, ausgegeben werden können.

Lösung

```
1 for ( int i=0 ; i<zeiten.length ;
      i=i+1 ){
2     System.out.println("Platz
          "+ (i+1) +": "+ zeiten
          [i] +"s");
3 }
```