

# Übung 6



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Aufgabe 1 (Wissensfrage)

- Erklären Sie den Unterschied zwischen Klassen- und Objektattributen.
  - Wie äußert sich der Wert des Attributs zwischen verschiedenen Objekten?
  - Wie werden diese deklariert?
  - Wie wird auf diese zugegriffen?
- Ist die Methode `double sqrt(double x)`, wie sie innerhalb der Klasse `Math` deklariert ist, eine Klassenmethode oder eine Objektmethode?
- Wenn eine Objektvariable durch eine lokale Variable in einem Anweisungsblock überdeckt wird, ist es dann trotzdem möglich auf die überdeckte Variable zuzugreifen? Wenn ja, wie? Wenn nein, weshalb nicht?
- Trifft dies auch auf überdeckte Klassenvariablen zu?

*static* 作为修饰符, 被声明为 *static* 的方法和变量  
可以直接用类名进行调用 (不用实例化!)

## Aufgabe 2 (Freunde)

Gegeben ist ein Programm mit den folgenden beiden Klassen `Info` und `Friend`:

```
1 public class Info {
2     public static String name = "WazzUp";
3     public static int versionNr = 5;
4
5     public static String getName() {
6         return name;
7     }
8
9     public static int getVersionNr() {
10        return versionNr;
11    }
12 }
13
```

*有 static*

```
1 public class Friend {
2     public String name;
3
4     public Friend(String name) {
5         setName(name);
6     }
7     public String getName() {
8         return name;
9     }
10    public void setName(String
11    name) {
12        this.name = name;
13    }
14    public String toString() {
15        return getName();
16    }
17 }
```

*Klassen Variable*  
*Klassennmethode*  
*无 static*

Machen Sie sich zunächst klar, bei welchen Attributen es sich um Klassen- bzw. Objektattribute sowie Klassen- oder Objektmethoden handelt.

Geben Sie dann zu den folgenden Code-Abschnitten an, ob sie kompilieren. Wenn ein Code-Abschnitt nicht kompiliert erläutern Sie kurz, warum das der Fall ist. Wenn ein Code-Abschnitt kompiliert, dann geben Sie für jede Zeile an:

- Bei Zuweisungen: Was wird nach der Zuweisung in der Zielvariablen gespeichert?
- Bei Ausgaben: Was wird nach der Ausgabe auf dem Bildschirm ausgegeben?

	Code-Abschnitt	Lösungsvorschlag
a)	<pre>int v = Info.getVersionNr(); System.out.println(v);</pre> <p>静态变量, 静态方法不需要初始化即能使用!</p>	<p><del>X</del></p> <p>Es gibt keine Initialisierung.</p> <p><math>v = 5</math>; 5</p>
b)	<pre>Info.setName("NEW");</pre>	Es gibt keine Methode.
c)	<pre>String f2Name = Friend.getName(); System.out.println(f2Name);</pre>	Es gibt keine Initialisierung.
d)	<pre>Friend f1 = new Friend("James Gosling"); System.out.println(f1.toString()); System.out.println(new Friend("Andrew Tanenbaum").getName());</pre>	<p>James Gosling</p> <p>Andrew Tanenbaum</p>
e)	<pre>Friend f3 = new Friend("Bill Gates"); Friend f4 = new Friend("Linus Torvalds"); f3 = f4;  String f3Name = f3.getName(); System.out.println(f3Name);</pre>	Linus Torvalds
f)	<pre>Friend f5 = new Friend(); System.out.println(f5.toString());</pre> <p>未定义了新的构造器, 默认构造器失效!</p>	<p>None <del>X</del></p> <p>Geht nicht, die Klasse "Friend" hat keine leeren Konstruktor.</p>

---

## Aufgabe 3 (Verwaltung von Sparkonten)

---

In der vorherigen Übung haben Sie bereits ein Programm zur Verwaltung von Konten implementiert. Die Bank hat nun **neue Anforderungen**, die Sie in ihrer Software umsetzen sollen. Neben einem normalen Konto bietet die Bank neuerdings auch **Sparkonten** an. Sparkonten unterscheiden sich von normalen Konten dadurch, dass das **Guthaben auf Sparkonten jährlich verzinst** wird. Die Bank bietet hierzu allen Kunden die gleichen **Konditionen bezüglich des Zinssatzes** an. Ändern Sie Ihr Programm aus Übung 6 wie folgt ab:

**Hinweis:** Sie können den Programmcode der Musterlösung von Übung 6 in Moodle herunterladen.

- Erstellen Sie eine neue Klasse `Sparkonto`. Diese soll die gleiche Funktionalität aufweisen, wie normale Konten. Kopieren Sie also einfach den Programmcode der Klasse `Konto` und ändern Sie den Namen der Klasse.
- Zu jedem `Sparkonto` soll nun der aktuell gültige Zinssatz gespeichert werden. Der Zinssatz kann sich jedoch von Jahr zu Jahr ändern. Fügen Sie der Klasse `Sparkonto` ein Attribut `zinssatz` vom Typ `double` hinzu. Überlegen Sie dabei, ob hier ein Objekt- oder ein Klassenattribut sinnvoller ist.
- Implementieren Sie in der Klasse `Sparkonto` die Methode `verzinsen()`. Der Methode werden keine Parameter übergeben und sie gibt keinen Wert zurück. Die Methode soll das aktuell auf dem `Sparkonto` vorhandene Guthaben entsprechend dem **zinssatz verzinsen**, d.h. das Guthaben entsprechend erhöhen. Für die Zahlung der Zinsen auf das Konto (und damit die Erhöhung des Kontostands) soll **die Methode `einzahlen(double betrag)`** genutzt werden.
- Implementieren Sie nun in der Klasse `Sparkonto` die folgende Klassenmethode: `jahresabschluss(Sparkonto[] konten)`. Diese Methode soll für jedes im übergebenen Array `konten` enthaltene `Sparkonto` die Methode `verzinsen()` aufrufen.
- Ändern Sie nun die Klasse `Kontoverwaltung` so ab, dass im Hauptprogramm (main-Methode) nun fünf `Sparkonto`-Objekte (und nicht mehr vom Typ `Konto`) erstellt werden. Deklarieren Sie nun ein Array, das `Sparkonto`-Objekte aufnimmt, und weisen alle fünf `Sparkonto`-Objekte diesem Array zu. Auf den Elementen des Arrays sollen nun die gleichen Ein- und Auszahlungen wie in der vorherigen Übung ausgeführt werden und danach alle Kontostände ausgegeben werden. Danach soll der Zinssatz für Sparkonten auf 0.04 gesetzt und ein Jahresabschluss durchgeführt werden, d.h. die Methode `jahresabschluss(Sparkonto[] konten)` soll mit dem erstellten Array von `Sparkonto`-Objekten aufgerufen werden. Geben Sie danach erneut alle Kontostände aus. Führen Sie danach noch einen weiteren Jahresabschluss mit einem Zinssatz von 0.1 aus und geben Sie wiederum die Kontostände aus.

**Hinweis:** Die Ausgabe der Konten können Sie in eine statische Methode innerhalb der Klasse `Kontoverwaltung` auslagern, damit Sie nicht jedes Mal eine neue Schleife schreiben müssen.

---

## Aufgabe 4 (Suchen von Zeichenketten)

---

- a) Erstellen Sie für die Klasse `MyString` eine statische Methode `occurrences`, die zwei Parameter vom Typ `String` übergeben bekommt und zurückgibt, wie häufig die erste der übergebenen Zeichenketten in der zweiten vorkommt. Nutzen Sie dazu die unten angegebenen Objektmethoden der Klasse `String`.

**Beispiel:** der folgende Aufruf soll den ganzzahligen Wert 1 ergeben:

```
MyString.occurrences(„ei“, „JaVa iSt eIn KafFeE, eiNe InSEl unD Eine  
ProgRamMiersprAche!“)
```

- b) Schreiben Sie eine weitere statische Methode `caseInsensitiveOccurrences`, die das gleiche macht wie die Methode aus Teilaufgabe (a), aber dabei nicht zwischen Groß-/Kleinschreibung unterscheidet. Z.B. kommt dann „a“ drei Mal in der Zeichenkette „Auf nach DA!“ vor. Überlegen Sie auch, ob Sie die schon bestehende Methode aus Teilaufgabe (a) hierzu wiederverwenden können.

**Beispiel:** der folgende Aufruf soll den ganzzahligen Wert 3 ergeben:

```
MyString.caseInsensitiveOccurrences(„ei“, „JaVa iSt eIn KafFeE, eiNe InSEl unD  
Eine ProgRamMiersprAche!“)
```

Nützliche Objektmethoden für Objekte der Klasse `String` (z.B. für `String s = „Haus“`):

- `s.indexOf(String searchValue, int startIndex);`  
searchValue: die zu suchende Zeichenkette  
startIndex (optional): der Index, an welchem die Suche beginnen soll  
Rückgabewert: Index, an dem die zu suchende Zeichenkette in s vorkommt. Wenn diese nicht vorkommt, wird -1 zurückgegeben.  
Bsp: `s.indexOf(„ei“, 0);` // ergibt -1
- `s.toLowerCase();`  
Rückgabewert: `String`, welcher s entspricht, jedoch nur Kleinbuchstaben enthält  
`s.toLowerCase();` // ergibt „haus“
- `s.toUpperCase();`  
Rückgabewert: `String`, welcher s entspricht, jedoch nur Großbuchstaben enthält  
`s.toUpperCase();` // ergibt „HAUS“

# Aufgabe 1:

a). Klassenattribut gehört zu **jeweiliger Klasse**. Alle Instanzen der Klasse teilen sich dieselbe Variable für das Attribut.

- Deklaration: `public static int xxx = 10;`

- Zugriff: `Klassenname.xxx`

Objektattribut gehört zu **konkretem Objekt (Instanz)**.

Jedes Objekt weist eigene Attribute aus, d.h. die Werte der Attribute könnten unterschiedlich sein.

- Deklaration: `public int yyy = 10;`

- Zugriff: `Objekt o = new objekt();`  
`o.yyy;`

b) Das ist eine Klassenmethode, weil es innerhalb der Klasse „Math“ definiert wird.

c) Ja, mittel des Schlüsselworts „this“ könnte Objektvariable zugegriffen werden:

this . xxx = xxx

This - 一般在 **Objekt** 中使用! (成员变量(对象))  
或者在成员方法中作为 **对构造器的指代** 使用。

d) Auf überdeckte Klassenvariablen kann ebenfalls zugegriffen werden. Hierfür muss über Klassennamen zugegriffen werden<sup>en</sup>.

(Class. Klassenvariable). Grund dafür ist: Es ist jedoch unabhängig von eigentlichem Objekt, deswegen wird „this“ nicht verwendet.