

### Aufgabe 1: Verständnisfragen – Lösung

- a) Erklären Sie, was man bei der Klasse "StringBuffer" unter den Begriffen "Size" und "Capacity" versteht.

*"Size" = aktuelle Anzahl Zeichen in einem StringBuffer-Objekt.*

*"Capacity" = mögliche Anzahl Zeichen, die ein StringBuffer-Objekt total enthalten kann, ohne dass eine Reorganisation des Speichers erfolgt.*

- b) Erklären Sie, warum es bei der Klasse "String" nicht sinnvoll ist von "Capacity" zu sprechen.

*Weil ein String-Objekt seine Grösse nicht (nie) ändern kann.*

- c) Erklären Sie, 1. was man unter dem Begriff "immutable" versteht und 2. warum diese Eigenschaft so wichtig ist.

*"immutable" = unveränderbar. String-Objekte sind inhaltlich "immutable" (unveränderbar). Das heisst, der Text, den sie beinhalten, kann nicht geändert werden.*

*Vorteil: Referenzen darauf können in einem Programm beliebig "herumgeschoben", dupliziert etc. werden, man kann trotzdem sicher sein, dass der Inhalt eines solchen Objektes nicht verändert wird oder wurde.*

- d) Gegeben ist die folgende Klasse "Person" (Fragment):

```
public class Person
{
    private String name;
    public Person (String name)
    {
        this.name = new String(name);
    }
    // ...
}
```

Erklären Sie, warum der obige Konstruktor unsinnig ist, obwohl das Programm gleichwohl korrekt funktioniert.

*'this.name = name;' genügt; es braucht keine String-Objekt-Kopie (deep-copy) da das String-Objekt 'name' ja immutable ist.*

- e) Erklären Sie, was - im Vergleich - die Vor- und Nachteile der Klassen "String" und "StringBuffer" sind.

*String:*

- Vorteile: einfach in der Anwendung (z.B. +-Operator), immutable;
- Nachteil: unter Umständen werden temporär String-Objekte erzeugt, die nachher vom GC wieder entsorgt werden müssen.

*StringBuffer:*

- Vorteile: vollständige Kontrolle über Speicherverwendung (Capacity);
- Nachteile: nicht immutable, etwas komplizierter in der Anwendung.