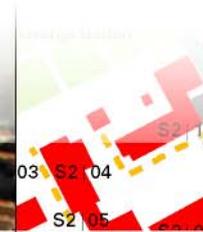


# Anregungen zu Übung 2

Allgemeine Informatik II - SS 2007



Fachgebiet Knowledge Engineering  
Prof. Dr. Johannes Fürnkranz



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



# Informationen zu Klausur und Programmierprojekt

## Klausur AI II

- Die Anmeldung ist ab sofort über die Homepage möglich.
- Anmeldeschluss: 19. August 2007

## Programmierprojekt

- Für das Programmierprojekt gibt es eine separate Anmeldung auf der Homepage. Ebenfalls ab sofort.
- Anmeldeschluss: 17. Juni 2007, bis zu drei Studenten pro Gruppe.
- Abgabeschluss: 13. Juli 2007, per Mail.
- Testate finden in der letzten Vorlesungswoche, also 16. bis 20. Juli 2007, zu den üblichen Tutorstunden statt.



# Fragen aus den Übungen

Ist es möglich, innerhalb von Methoden `static`-Variablen zu definieren?

- Nein, dies ist nicht möglich.
- Variablen innerhalb von Methoden sind immer nur im Methodenrumpf gültig. Nach Abarbeiten einer Methode, werden diese vernichtet.
- Variablen, die als `static` deklariert sind, können aber von überall als Klassenvariable aufgerufen werden.
- Dies führte zu einem Widerspruch.
- Möglich sind aber Konstanten (`final`) innerhalb von Methoden.



# Fragen aus den Übungen

Die Methode `int count(char c)` der Klasse `LetterCounter` ist case-sensitive. Wie ist es möglich, sie case-insensitive zu gestalten?

Es gibt zwei einfache Wege,

- über einen case-insensitiven Vergleich oder
- mittels eines einfachen Workarounds.
- Lösungen auf den nächsten Folien.



# Fragen aus den Übungen

## Ein case-insensitiver Vergleich

```
01 private int count(char c) {
02     // Zähler
03     int number = 0;
04     // Der Char c wird nach String umgewandelt
05     String vorgabe = Character.toString(c);
06     for (int i = 0; i < text.length(); i++) {
07         // Der Char an der Stelle i wird nach String umgewandelt
08         String vergleich = Character.toString(text.charAt(i));
09         // Die Klasse String stellt einen
10         // case-insensitiven Vergleich zur Verfügung.
11         if(vorgabe.equalsIgnoreCase(vergleich))
12             number++;
13     }
14     return number;
15 }
```

Hinweis: Die Methode verlangt den String text und kann z. B. in die Klasse LetterCounter eingefügt und dort verwendet werden.



# Fragen aus den Übungen

## Ein einfacher Workaround

```
01  private int count(char c) {
02      // Zähler
03      int number = 0;
04      // Der Char c wird standardmäßig
05      // in einen Kleinbuchstaben umgewandelt
06      c = Character.toLowerCase(c);
07      // Der String text wird ebenfalls
08      // in Kleinbuchstaben umgewandelt
09      String text1 = text.toLowerCase();
10      for (int i = 0; i < text1.length(); i++) {
11          // Der Vergleich wie im case-sensitiven Fall
12          if(text1.charAt(i) == c)
13              number++;
14      }
15      return number;
16  }
```

Hinweis: Die Methode verlangt den String text und kann z. B. in die Klasse LetterCounter eingefügt und dort verwendet werden.



# Fragen aus den Übungen

Der Konstruktor der Klasse `LetterCounter` initialisiert einen String `text`. Vorsicht mit dem Scope der Variablen.

Einige Lösungen entsprachen dem Beispiel auf dieser Seite:

- Was ist daran falsch?
- Wie kann man es korrigieren?

```
1 public class LetterCounter {
2     // Objektvariable text
3     private String text;
4     // Konstruktor initialisiert String text
5     public LetterCounter(String text) {
6         text = text;
7     }
8 } // Vorsicht, diese Lösung führt zu einem Fehler!!!
```



# Fragen aus den Übungen

Der Konstruktor der Klasse `LetterCounter` initialisiert einen String `text`. Vorsicht mit dem Scope der Variablen.

Lösung 1:

- Verwenden eines anders lautenden Parameters.

```
1 public class LetterCounter {
2     // Objektvariable text
3     private String text;
4     // Konstruktor initialisiert String text
5     public LetterCounter(String text1) {
6         text = text1; // text ist die Objektvariable
7     }
8 }
```



# Fragen aus den Übungen

Der Konstruktor der Klasse `LetterCounter` initialisiert einen String `text`. Vorsicht mit dem Scope der Variablen.

Lösung 2:

- Zugriff auf die Objektvariable mittels des Schlüsselwortes `this`.
- Das Schlüsselwort `this` ist ein Verweis auf ein Datenfeld.

```
1 public class LetterCounter {
2     // Objektvariable text
3     private String text;
4     // Konstruktor initialisiert String text
5     public LetterCounter(String text) {
6         this.text = text; // this.text ist die Objektvariable
7     }
8 }
```

Hinweis Zeile 6: (Verweis auf Objektvariable).text = (Verweis auf Variable).text;