

# Praktische Informatik II - FSS 2012

## Übungsblatt 0

Lehrstuhl Prof. Dr. Heiner Stuckenschmidt

13. Februar 2012

### Organisatorisches

- Abgaben, welche nach der Frist eingereicht werden, werden prinzipiell nicht bepunktet. Abgaben werden nur akzeptiert, wenn auf dem ersten Blatt Name und Matrikelnummer jedes Gruppenmitglieds angegeben ist.
- Alle bepunkteten Aufgaben müssen zwingend in einer einzelnen PDF-Datei eingereicht werden. Abgaben in anderen Formaten (wie doc, docx usw.) werden nicht bepunktet. Die Abgabe erfolgt über Ilias.
- Die auf den Blättern angegebenen Programmieraufgaben werden nicht korrigiert und in der Regel nicht in den Tutorien oder der großen Übung besprochen. Eine detaillierte Erklärung hierzu, siehe unten.
- Das Abschreiben von Quellen (bspw. Lehrbuch oder Internetseiten) zählt als Plagiat. Lösungen sind stets in eigenen Worten zu formulieren!
- Rechenwege und Erklärungen sind auch dann notwendig, wenn sie nicht explizit in der Aufgabenstellung verlangt werden.

### Aufgabe 1: Formalia (1 Punkt)

Für das 0-te Übungsblatt gibt es ausnahmsweise einen Punkt für eine fristgerechte Abgabe mit Angabe von Name und Matrikelnummer aller Gruppenmitglieder auf der ersten Seite des Dokuments. Alle folgenden Abgaben können nur als gültige Abgaben gewertet werden, wenn diese Angaben vorliegen!

### Aufgabe 2: Von-Neumann Architektur (6 Punkte)

- a) (2 Punkte) Beschreiben Sie die grundlegenden Prinzipien der Von-Neumann-Architektur!

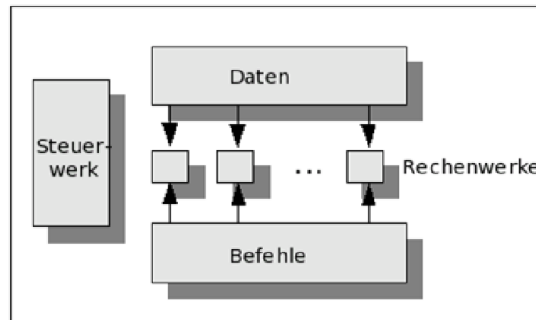


Abbildung 1: Die Harvard Architektur

- b) (4 Punkte) Abbildung 1 zeigt die sogenannte Harvard Architektur, die als Alternative zur Von-Neumann-Architektur vorgeschlagen wurde. Was ist der Hauptunterschied zwischen den beiden Architekturen? Welchen Vorteil bietet die Harvard Architektur gegenüber der Von-Neumann-Architektur? Erläutern Sie in diesem Zusammenhang den Begriff *Neumann Bottleneck*!

### Aufgabe 3: Speicherhierarchie (3 Punkte)

Erläutern Sie die im Kontext der Speicherhierarchie eingeführten Begriffe *Register*, *Cache*, *Hauptspeicher*, *Magnetplatte*, *Band*, und *optische Platte* jeweils in einigen Sätzen (Funktionsweise, Sinn und Zweck)! Geben Sie jeweils auch an, ob eine entsprechende Speichereinheit in einem Standard PC in der Regel vorhanden ist, bzw. ob ein Mechanismus vorhanden ist, der einen Zugriff auf eine solche Speichereinheit erlaubt!

### Hinweis zur Bearbeitung der Programmieraufgaben

Die erfolgreiche Bearbeitung der Programmieraufgaben unterliegt der Selbstkontrolle. Zu jeder Programmieraufgabe gibt es ein Eclipse Projekt zusammen mit den entsprechenden Testklassen, so dass eine selbstständige Überprüfung ohne Kenntnis der Musterlösung möglich ist. Eine Musterlösung (Code) wird einige Tage nach der Besprechung des jeweiligen Aufgabenblatts in ILIAS zur Verfügung stehen. Die Lösungen zu den Programmieraufgaben sollen jedoch nicht abgegeben werden, werden nicht von den Tutoren korrigiert, und werden (in der Regel) auch nicht in den Tutorien oder der Großen Übung besprochen.

## Aufgabe A: Fibonacci-Zahlen

Die so genannten Fibonacci-Zahlen sind folgendermaßen rekursiv definiert.

$$f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-2}$$

Basierend auf dieser Definition soll nun die Berechnung der n-ten Fibonacci-Zahl implementiert werden. Hierfür sind ein entsprechendes Interface und eine Test-Klasse beigefügt.

- Vervollständigen Sie die Klasse `RecursiveFibonacciComputation`, welche das Interface `FibonacciComputation` implementiert und die Berechnung der Fibonacci-Zahlen mittels rekursiver Methodenaufrufe vornimmt.
- Vervollständigen Sie die Klasse `IterativeFibonacciComputation`, welche das Interface `FibonacciComputation` implementiert und die Berechnung der Fibonacci-Zahlen unter Nutzung von Schleifen (und ohne rekursive Aufrufe) vornimmt.

## Programmieraufgabe B: Transformation von Strings

Erweitern Sie die Klassen `LowerCaseTransformer` und `BSpeechTransformer`, die das vorgegebene Interface `StringTransformer` implementieren. Dieses Interface gibt eine Methode `transform` vor, die einen String als Parameter erhält und auch einen String als Rückgabetypp besitzt.

- Erweitern Sie die Klasse `LowerCaseTransformer`, so dass ein eingegebener String mittels der Methode `transform` in Kleinbuchstaben umgewandelt wird!
- Erweitern Sie die Klasse `BSpeechTransformer`, so dass ein eingegebener String in einen entsprechenden Ausdruck der B-Sprache transformiert wird! In der B-Sprache werden alle Vokale und Umlaute  $\alpha$  durch  $\alpha b \alpha$  ersetzt. Hier ein Beispiel: *ka**b**annst du**b**u mi**b**ir sa**b**ageben, wo**b**o du**b**u bi**b**ist? = kannst du mir sagen, wo du bist?* Der Einfachheit halber soll der Rückgabewert nur Kleinbuchstaben enthalten. Etwas komplizierter wird es durch einige Sonderregelung in Bezug auf Buchstabenkombinationen wie *au* und *eu* usw. Diesbezüglich soll *au* auf *abau*, *ie* auf *ibie*, *ei* auf *abei*, und *eu* soll auf *obeu* abgebildet werden. Diese Regelungen haben Vorrang vor der Grundregel.

**Unbepunktete Zusatzfrage** *Ibist ebes möböglibich abeineben Sabatz deber b-Sprabachebe abeindobeutibig ibin deben Sabatz zuburübück zubu übübebersebetzeben, abaus debem deber Sabatz ubursprübünglibich eberzobeugt wuburdebe?*