Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Einführung in die Informatik
Sommersemester 2012

Prof. Dr. Wolfram Burgard
PD Dr. Cyrill Stachniss
Dipl. Ing. Jörg Röwekämper
Dipl. Inf. Markus Kuderer

Übungsblatt 13

Abgabe bis Freitag, 03.08.2012, 12:00 Uhr

Hinweis:

Aufgaben immer per E-Mail (eine E-Mail pro Blatt und Gruppe) an den zuständigen Tutor schicken (Bei Programmieraufgaben Java Quellcode und eventuell benötigte Datendateien).

Aufgabe 13.1

Eine Warteschlange (engl. „Queue“) ist eine Datenstruktur, die Objekte nach dem FIFO-Prinzip (First In – First Out) verwaltet, d.h. dass Objekte genau in der Reihenfolge ausgegeben werden, in der sie in die Warteschlange eingefügt werden. In dieser Aufgabe soll eine Warteschlange mit einer einer einfach verketteten Liste implementiert werden.

```
public class Queue {
  public Queue() {...}
  public void push(Object o) {...}
  public Object pop() {...}
  ...
  protected QueueNode head;
  protected QueueNode tail;
}

public class QueueNode {
  ...
  private Object content;
  ...
}
```

1. Geben Sie die vollständige Klassendefinition eines Speicherelements `QueueNode` mit geeigneten Zugriffsmethoden sowie typischem Konstruktor an und implementieren Sie diese.

2. Implementieren Sie den Konstruktor sowie die Methoden `push()` und `pop()` der Klasse `Queue`. Die Methode `push()` fügt dem Speicher ein Element hinzu. Die Methode `pop()` liefert das nächste abzuarbeitende Element zurück und löscht dieses aus der Warteschlange.

3. Geben Sie den Aufwand für ihre Methoden in Abhängigkeit der Anzahl an gespeicherten Objekten an.

Aufgabe 13.2

Sind die folgenden Aussagen korrekt? Begründen Sie Ihre Antwort.

1. Deterministische Verarbeitungsvorschriften liefern zur gleichen Eingabe immer die gleiche Ausgabe.
2. Die Menge der HTML-Dokumente im Internet ist überabzählbar.
3. Jede Funktion kann durch einen Algorithmus beschrieben werden.
4. Durch den Cantorschen Diagonalschluss kann gezeigt werden, dass die Menge der `while`-Programme überabzählbar ist.
5. Es gibt Java-Programme, für die gezeigt werden kann, dass sie mit beliebigen Eingaben in eine Endlosschleife geraten.
6. Gäbe es ein Programm, das überprüft ob zwei Programme immer die gleiche Ausgabe liefern, wäre das Halteproblem entscheidbar.

Aufgabe 13.3

Betrachten sie folgendes Programm, das die Eingabewerte `a` und `b` addieren soll:

```
public static int add(int a, int b) {
  if(a % 2 == 0) {
    return a + b;
  } else {
    while(true){ }
  }
}
```

1. Die Methode `add(int a, int b)` ist partiell korrekt.
2. Die Methode `add(int a, int b)` ist total korrekt.
3. Die Methode `add(int a, int b)` löst das Halteproblem.
4. Die Methode `add(int a, int b)` ist ein Algorithmus.