Name: ……

Matrikel-Nr. ……

Prüfung Programmieren II, SS 2014

Studiengang IW

Datum, Uhrzeit: 23.07.2014, 11:00 Uhr

Semester: SS 2014

Prüfungsnummern: 4710060, 2410030

Prüfer: Prof. Dr. Thomas Wölfl, Prof. Dr. Jan Dünnweber

Dauer: 90 Minuten

Hilfsmittel: Keine

Die Prüfung umfasst 9 Seiten (bitte sofort nachprüfen). Die Lösungen für die Aufgaben sind auf den Aufgabenblättern einzutragen. Sie können auch die Rückseiten verwenden. Achten Sie bitte darauf, dass auf jedem Blatt Ihr Name und Ihre Immatrikulationsnummer vermerkt sind. Ist eine Aufgabenstellung Ihrer Meinung nach nicht vollständig oder mehrdeutig, so treffen Sie entsprechende Annahmen.

Prüfungsbewertung:

Aufgabe	1	2	3	4	**Gesamt**
Ergebnis					

Prüfungsergebnis:

Note	

1. Prüfer

2. Prüfer

Aufgabe 1 (Grundlagen)

a) Worin unterscheidet sich die Programmiersprache Java von C? Nennen Sie drei Stichpunkte.

b) Nennen Sie vier Grunddatentypen von Java mit den zugehörigen Wertebereichen.

c) Worin unterscheidet sich UNICODE von ANSI? Nennen Sie einen Aspekt.

d) Rechnen Sie die Dualzahl 1110 in das Dezimalsystem um.

e) Was ist die Aufgabe einer Betriebssystem-Umgebungsvariablen?

Aufgabe 2 (Klassenmodell, Programmierung)

Es soll ein einfaches Projektverwaltungsprogramm entwickelt werden. Die verwalteten Projekte haben eine Laufzeit (Datum von, Datum bis), einen Status (offen, läuft, abgeschlossen), einen Tagessatz (Geldbetrag) und eine Bezeichnung. Weiterhin gibt es zwei Arten von Projekten:

- Öffentliche Projekte. Diese besitzen zusätzlich einen Kostenträger. Ein Kostenträger ist durch eine 18-stellige Nummer (nur Zahlen) und einen Namen gekennzeichnet.
- Private Projekte. Diese besitzen zusätzlich ein Projektbudget (Geldbetrag).

Für jedes Projekt kann berechnet werden, welche Kosten durch das Projekt bereits entstanden sind. Dafür wird der Tagessatz des Projektes mit der Anzahl von Tagen multipliziert, die seit dem Projektbeginn bis zum Tag der Kostenberechnung verstrichen sind. Wenn das Projekt bereits das Laufzeitende erreicht oder überschritten hat, darf nur die Laufzeit des Projektes in die Kostenberechnung eingehen.

Die Kostenberechnung von öffentlichen Projekten gleicht der Berechnung von privaten Projekten, mit dem Unterschied, dass die Kosten von öffentlichen Projekten um den Faktor 1,6 höher sind.

a) Erstellen Sie ein vereinfachtes Klassenmodell mit Attributen, Methoden und Datentypen, das für diese Aufgabenstellung geeignet ist.

b) Programmieren Sie die Methoden, welche Sie für die Kostenberechnungen von privaten und öffentlichen Projekten vorgesehen haben.

c) In der Software gibt es eine Liste *list* von Projekten. Diese Liste soll absteigend nach den Projektkosten sortiert werden. Ergänzen Sie Ihre Klassen entsprechend und implementieren Sie die hierfür nötigen Änderungen bzw. Erweiterungen.

d) In der Software gibt es weiterhin eine Liste *list* von Projekten. Programmieren Sie eine statische Methode mit dem Namen *cleanUp*, welche die Liste *list* als Parameter übergeben bekommt. Diese Methode löscht aus der übergebenen Liste alle Projekte, die den Status „abgeschlossen“ haben, wenn diese älter als 10 Jahre sind.

Aufgabe 3 (Codeverständnis)

a) Beschreiben Sie die Hauptaufgabe der Methode *doIt* des folgenden Java-Programms in ca. 2 – 3 Sätzen. Welche Anforderung bzw. Funktionalität setzt die Methode um?

```
1  public class WasMacheIch {
2
3      private static char[] p = { '.', '!', '?', ':' };
4
5      public static int doIt(String s) {
6          if (s == null){
7              return 0;
8          }
9          int x = 0;
10         char[] a = s.toCharArray();
11         for (char c : a) {
12             if (check(c)){
13                 ++x;
14             }
15         }
16         return x;
17     }
18
19     private static boolean check(char a) {
20         for (char c : p) {
21             if (c == a) {
22                 return true;
23             }
24         }
25         return false;
26     }
27 }
```

b) In dem folgenden Java-Programm sind zwei Fehler enthalten. Nennen Sie die Fehler und begründen Sie Ihre Wahl.

```
1  public class Fehlerhaft {
2
3      public static void main(String[] args) {
4
5          int[] x = new int[5];
6
7          for (int i = 5; i > 0 ; i--){
8              System.out.println("Zahl: " + x[i]);
9              if( x = 5 ){
10                 System.out.println("Zahl 5 ist enthalten");
11             }
12         }
13     }
14 }
```

Aufgabe 4 (Threads)

Gegeben ist die folgende Klasse, die eine Liste von Strings enthält.

```
1 import java.util.ArrayList;
3
4 public class Jobs {
5
6     // Nehmen Sie an: list enthält sehr viele Elemente
7     public static List<String> list = new ArrayList<String>();
8
9     public static String getElement() {
10        if (list != null && !list.isEmpty()) {
11            return list.remove(0);
12        }
13        return null;
14    }
15
16    public static void addElement(String s){
17        list.add(s);
18    }
19 }
```

Nehmen Sie an, dass die String-Liste *list* eine große Anzahl von Elementen enthält. Programmieren Sie einen Thread, der sich mit Hilfe der Methode *getElement* der vorliegenden Klasse ein Element aus der Liste abholt und dieses am Bildschirm ausgibt. Jedes Element der Liste soll genau einmal angezeigt werden. Von diesem Thread sollen in einem Hauptprogramm acht Instanzen angelegt und gestartet werden. Wenn alle Threads fertig sind (weil die String-Liste leer ist), soll der Text „Ende“ am Bildschirm ausgegeben werden. Verändern Sie die vorliegende Klasse Jobs nicht.

(Für die Lösung können Sie auch die Rückseite verwenden)