

Allgemeine Informatik I

Prof. J. Fürnkranz

Technische Universität Darmstadt — Wintersemester 2006/07

Termin: 8. 3. 2007

Name:

Vorname:

Matrikelnummer:

Fachrichtung:

Wiederholer: ja nein

Diplom

Master

Bachelor

Punkte:

(1) ...

(2) ...

(3) ...

(4) ...

(5) ...

(6) ...

Summe:

- **Aufgaben:** Diese Klausur enthält auf den folgenden Seiten 6 Aufgaben zu insgesamt 80 Punkten. Jede Aufgabe steht auf einem eigenen Blatt. Kontrollieren Sie *sofort*, ob Sie alle sechs Blätter erhalten haben!
- **Zeiteinteilung:** Die Zeit ist knapp bemessen. Wir empfehlen Ihnen, sich zuerst einen kurzen Überblick über die Aufgabenstellungen zu verschaffen, und dann mit den Aufgaben zu beginnen, die Ihnen am besten liegen.
- **Papier:** Verwenden Sie nur Papier, das Sie von uns ausgeteilt bekommen. Bitte lösen Sie die Aufgaben auf den dafür vorgesehenen Seiten (auch auf den Rückseiten). Falls der Platz nicht ausreicht, vermerken Sie dies bitte und setzen die Lösung auf einem Zusatzblatt fort. Brauchen Sie zusätzlich Papier (auch Schmierpapier), bitte melden.
- **Fragen:** Sollten Sie Teile der Aufgabenstellung nicht verstehen, bitte fragen Sie!
- **Abschreiben:** Sollte es sich (wie in den letzten Jahren leider immer wieder) herausstellen, daß Ihre Lösung und die eines Kommilitonen über das zu erwartende Maß hinaus übereinstimmen, werden beide Arbeiten negativ beurteilt (ganz egal wer von wem in welchem Umfang abgeschrieben hat).
- **Ausweis:** Legen Sie Ihren *Studentenausweis* und *Lichtbildausweis* sichtbar auf Ihren Platz. Füllen Sie das Deckblatt sofort aus!
- **Hilfsmittel:** Zur Lösung der Aufgaben sind keine Unterlagen erlaubt. Gedruckte Wörterbücher sind für ausländische Studenten erlaubt, elektronische Hilfsmittel (Taschenrechner, elektronische Wörterbücher, Handy, etc.) sind verboten! Sollten Sie etwas anderes verwenden wollen, bitte klären Sie das *bevor* Sie zu arbeiten beginnen.
- **Aufräumen:** Sonst darf außer Schreibgerät, Essbarem, von uns ausgeteiltem Papier und eventuell Wörterbüchern nichts auf Ihrem Platz liegen. Taschen bitte unter den Tisch!

Gutes Gelingen!

Aufgabe 1 (8 Punkte)

IP-Adressen werden zur Adressierung von Rechner im Internet verwendet.

- 1-a Welche allgemeine Form haben IP-Adresse?
- 1-b Wie viele Bytes benötigt man zur Darstellung einer IP-Adresse?
- 1-c Wie lautet die Darstellung der IP-Adresse 130.254.47.128 durch 4 Hexadezimalzahlen?
- 1-d Skizzieren Sie kurz den Vorgang, mit dem ein URL wie `http://www.tu-darmstadt.de` in die zugehörige IP-Adresse übersetzt wird.

Aufgabe 2 (8 Punkte)

Beantworten Sie die folgenden Fragen möglichst kurz und prägnant (1-2 Sätze pro Teilaufgabe).

- 2-a Welche zentrale Idee, die sich bereits in den theoretischen Modellen der Von Neumann-Rechner-Architektur bzw. in der Turing-Maschine findet, verhalf dem Computer zum Durchbruch?
- 2-b Wie kann eine einzige CPU mehrere Prozesse scheinbar gleichzeitig ausführen? Welche Rolle spielt dabei die Priorität eines Prozesses?
- 2-c Was versteht man unter Unicode?
- 2-d Was versteht man unter Maschinensprache im Gegensatz zu Java oder KarelJ?

Aufgabe 3 (12 Punkte)

Gegeben seien folgende Klassendefinitionen mit anschliessendem Task:

```
1  class A extends Robot {
2      void write1() {
3          System.out.print("X");
4      }
5      void write2() {
6          System.out.print("Y");
7      }
8      void write() {
9          write1();
10         write2();
11     }
12 }
13
14 class B extends A {
15     void write1() {
16         super.write1();
17         System.out.print("Z");
18     }
19 }
20
21 class C extends B {
22     void write2() {
23         super.write1();
24         System.out.print("W");
25     }
26 }
27     task {
28         A a;
29         C c;
30
31         a = new A(1,1,0,East);
32         a.write();
33
34         c = new A(1,1,0,East);
35         c.write();
36
37         a = new B(1,1,0,East);
38         a.write();
39
40         c = new B(1,1,0,East);
41         c.write();
42
43         a = new C(1,1,0,East);
44         a.write();
45
46         c = new C(1,1,0,East);
47         c.write();
48     }
```

Geben Sie für jeden der Aufrufe der Methode `write` in den Zeilen 32, 35, 38, 41, 44, 47 an, ob dieser Aufruf erfolgreich durchgeführt werden kann.

- Wenn ja, geben Sie die Ausgabe auf dem Bildschirm an.
- Wenn nein, geben Sie eine kurze Begründung.

Hinweis: `System.out.print("X")` schreibt den Buchstaben X auf den Bildschirm.

je 2 Punkte

Aufgabe 4 (12 Punkte)

Implementieren Sie eine Methode `merge`, die zwei gleich lange Arrays von `int`-Zahlen miteinander vermischt. Die Methode soll einen Array zurückgeben, der mit dem ersten Element des ersten Arrays beginnt und dem letzten Element des zweiten Arrays endet.

Beispiel:

Arrays, die an `merge` übergeben werden:

1	2	3	4
---	---	---	---

21	22	23	24
----	----	----	----

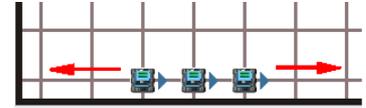
Rückgabe:

1	21	2	22	3	23	4	24
---	----	---	----	---	----	---	----

Hinweise: Die übergebenen Arrays können beliebig lang sein, müssen aber gleich lang sein. Sie müssen *nicht* überprüfen, ob die beiden Arrays wirklich gleich lang sind. Vergessen Sie nicht, einen passenden Methoden-Kopf zu definieren.

Aufgabe 5 (17 Punkte)

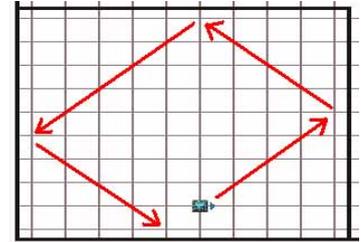
Schreiben Sie eine Klasse **Schlaeger**. Ein Schlaeger besteht aus drei Robotern auf benachbarten Feldern, die gemeinsam auf der ersten Strasse hin und her bewegt werden. Der zentrale Roboter ist für die Steuerung verantwortlich, die beiden Hilfsroboter links und rechts davon bewegen sich synchron dazu.



- 5-a Geben Sie eine Klassendefinition mit geeigneten internen Variablen an, in denen die beiden zusätzlichen Roboter gespeichert werden können.
- 5-b Definieren Sie einen Konstruktor `Schlaeger(int a)`, der den zentralen Roboter auf 1. Strasse und der `a`-ten Avenue, die beiden Hilfsroboter auf der 1. Strasse und den Avenues `a-1` und `a+1`, jeweils mit 0 Bepern und nach Osten blickend, platziert.
- 5-c Überschreiben Sie die `move`-Methode, sodaß sich der Schlaeger (also alle drei Roboter) einen Schritt bewegt.
- 5-d Überschreiben Sie die `nextToABeeper`-Methode, sodaß sie `true` zurückgibt, wenn zumindest einer der drei Roboter auf einem Feld mit einem Beeper steht, und `false` sonst.

Aufgabe 6 (23 Punkte)

Implementieren Sie eine Roboter-Klasse `Ball`. Ein Ball bewegt sich immer diagonal über die Gitter-Linien. Immer wenn der Ball im nächsten Schritt auf eine Wand treffen würde, prallt er von dieser Wand ab, und ändert dementsprechend seine Richtung. Während er sich über das Spielfeld bewegt, sammelt er Beeper.



- 6-a Geben Sie die Klassendefinition an, in die die in der Folge zu implementierenden Methoden integriert werden können
- 6-b Überschreiben Sie die `move`-Methode, sodaß Roboter vom Typ `Ball` einen Diagonal-Schritt links von der momentanen Richtung durchführen (also wenn der Roboter nach Osten blickt, ein Schritt nach Nordosten). Die Richtung des Roboters vor und nach Ausführung des Schrittes muß gleich bleiben.
- 6-c Schreiben Sie eine Methode `pickUpBeepers`, die alle Beeper aufhebt, die sich auf dem momentanen Feld des Roboters befinden.
- 6-d Überschreiben Sie die `frontIsClear`-Methode, sodaß der Roboter überprüft, ob ein wie oben definierter `move`-Schritt durchgeführt werden kann, ohne gegen eine Wand zu laufen. Die Welt soll nach Ausführung der Methode unverändert sein.

Hinweis: Sie müssen dabei bei jedem Teilschritt überprüfen, ob er erfolgreich durchgeführt werden kann, und danach wieder aufs Ausgangsfeld zurückkehren.

- 6-e Schreiben Sie eine Methode `fly`, die bewirkt, daß sich der Ball solange bewegt, bis er auf der 1. Strasse landet. Vor jedem Schritt soll er
- alle auf dem momentanen Feld befindlichen Beeper aufheben
 - überprüfen, ob er nicht gegen eine Wand laufen würde. Wenn ja, dann wird zuerst versucht mit einer Linksdrehung auszuweichen. Wenn er dann ebenfalls gegen eine Wand laufen würde, wird eine (von der ursprünglichen Position aus gesehen) Rechtsdrehung durchgeführt.

Danach wird ein Diagonal-Schritt ausgeführt.

Hinweis: Verwenden Sie bei der Implementierung die Methoden der Aufgaben b-d (auch wenn Sie diese nicht vollständig lösen konnten).