Prof. Dr.-Ing. J. Hornegger
Lehrstuhl für Mustererkennung
Universität Erlangen-Nürnberg

20. Februar 2006

Schriftliche Prüfung zu der Vorlesung Grundlagen der Informatik I Februar 2006

(Prüfungsnummern 27348110 und 11248143)

Angaben zur Person:

Name, Vorname .....

Geburtsdatum .....

Matrikelnummer .....

Nicht von der Kandidatin/dem Kandidaten auszufüllen!

Bewertung:

Aufgabe	1	2	3	4	5	6	$\sum$	Note
Punktezahl								

Organisatorische Hinweise

Die folgenden Hinweise bitte aufmerksam lesen und die Kenntnisnahme durch Unterschrift bestätigen!

- Bitte legen Sie Ihren Studentenausweis und einen Lichtbildausweis zur Personenkontrolle bereit!
- Die Bearbeitungszeit beträgt 90 Minuten. Da Sie maximal 90 Punkte erreichen können, ist aus den möglichen Punkten pro Aufgabe leicht die Zeit zu berechnen, die Sie zum Lösen der jeweiligen Aufgabe investieren sollten. Die angegebene Punkteverteilung gilt unter Vorbehalt.
- Hilfsmittel (außer Schreibmaterial) sind nicht zugelassen.
- Fragen zu den Prüfungsaufgaben werden grundsätzlich nicht beantwortet!
- Auf Ihrem Platz befindet sich 1 Blatt Schmierpapier. **Das Schmierpapier darf nicht mit abgegeben werden.** Die Lösung einer Aufgabe muss auf das Aufgabenblatt geschrieben werden und zwar in den freien Raum, der jeweils der Aufgabe folgt. Sollte der Platz nicht ausreichen, so müssen Sie bei bei der Aufsicht zusätzliche Formblätter anfordern und einheften lassen.
- Wenn Sie die Prüfung aus gesundheitlichen Gründen abbrechen müssen, so muss Ihre Prüfungsunfähigkeit durch eine Untersuchung in der Universitätsklinik nachgewiesen werden. Melden Sie sich in jedem Fall bei der Aufsicht und lassen Sie sich das entsprechende Formular aushändigen.
- **Überprüfen Sie die Prüfungsaufgaben auf Vollständigkeit (13 Seiten inklusive Deckblatt) und einwandfreies Druckbild!** Vergessen Sie nicht, auf dem Deckblatt die Angaben zur Person einzutragen!

Erklärung

Durch meine Unterschrift bestätige ich den Empfang der vollständigen Klausurunterlagen und die Kenntnisnahme der obigen Informationen.

Erlangen, der 20. Februar 2006 .....
(Unterschrift)

Ich bin damit einverstanden, dass mein Prüfungsergebnis unter Angabe der Matrikelnummer anonymisiert veröffentlicht wird:

ja: ☐ nein: ☐

Erlangen, der 20. Februar 2006 .....
(Unterschrift)

Aufgabe 1 (10 Punkte)

Kreuzen Sie die richtigen Lösungen an. Es können mehrere Antworten richtig sein.

1. Welche Aussagen treffen auf die Darstellung von ganzen Zahlen in einem Digitalrechner zu?

   (a) Ganze Zahlen werden im B-Komplement durch ein Vorzeichen-Bit und den Betrag der Zahl dargestellt. ☐

   (b) Im B-1-Komplement ist die Darstellung der 0 eindeutig. ☐

   (c) Im B-Komplement sind alle Zahlen mit einer führenden 1 negativ. ☐

   (d) Im B-Komplement ist der darstellbare Zahlenbereich unsymmetrisch. ☐

2. Für die Darstellung von Gleitkommazahlen im IEEE-Format stimmen folgende Aussagen:

   (a) Die Genauigkeit, mit der eine Zahl gespeichert wird, hängt von der Größe der Zahl ab. ☐

   (b) Abhängig von der Größe des Exponenten einer Zahl werden untschiedlich viele Bits verwendet. ☐

   (c) Im IEEE-Format können beliebig große Zahlen (mit abnehmender Genauigkeit) gespeichert werden. ☐

   (d) Das Vorzeichen der Zahl wird in einem eigenen Bit gespeichert. ☐

3. Folgende Aussagen sind für den klassischen Universalrechner zutreffend:

   (a) Das Steuerwerk im klassischen Universalrechner dient zur Ausführung von Operationen mit einem oder zwei Operanden. ☐

   (b) Durch den Cache lässt sich der Speicherplatz eines Rechners erweitern. ☐

   (c) Cache-Speicher nutzen die Tatsache aus, dass aufeinander folgende Programmbefehle häufig auch physikalisch zusammengehörend gespeichert werden. ☐

   (d) Werden Funktionseinheiten als *Pipeline* realisiert, kann dadurch die Berechnung gleichartiger Operationen beschleunigt werden. ☐

4. Für Typ-2-Grammatiken gilt:

   (a) Auf der rechten Seite einer Regel dürfen nur terminale Symbole stehen. ☐

   (b) Auf der rechten Seite dürfen nicht weniger Symbole stehen als auf der linken Seite. ☐

   (c) Auf der rechten Seite dürfen nur nicht-terminale Symbole stehen. ☐

   (d) Die linke Seite besteht ausschließlich aus einem nicht-terminalen Symbol. ☐

5. Welche der folgenden Aussagen treffen auf Sortierverfahren zu:

   (a) Das Verfahren *Quicksort* ist ein so genanntes *Divide & Conquer*-Verfahren. ☐

   (b) *Quicksort* ist ein effizientes Verfahren, dessen Aufwand zum Sortieren immer bei $O(n \cdot \log n)$ liegt. ☐

   (c) Bei *BubbleSort* wächst der Aufwand exponentiell mit der Anzahl der zu sortierenden Elemente. ☐

   (d) Den Verfahren *QuickSort* und *BubbleSort* ist gemeinsam, dass das zu sortierende Feld nicht kopiert werden muss. ☐

Aufgabe 2 (14 Punkte)

1. Folgendes Programm soll dazu dienen, die Zahlen von 1 bis 10 zu addieren und auszugeben. Verbessern Sie die Fehler, sodass es sich danach um ein fehlerfreies Java-Programm handelt, welches die gestellte Aufgabe erfüllt.

```
public class

    public static void main(String[] args) {
        int byte = 1;

        for (int i; i < 10;) {
            sum += i;
            i++
        }

        System.println("Summe: " + sum);

        return 0;

}
```

2. Geben Sie an, was beim Ablauf des Programmes der Reihe nach auf `stdout` ausgegeben wird.

```
public class Scope {

    private static int a = 10;
    private static int c = 5;

    public static void vertausche(int a, int b) {
        int h = a;

        a = b;
        b = h;
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("c = " + c);
    }

    public static void main(String[] args) {
        int a = 0;
        int z1 = 14;
        int z2 = 7;
        vertausche(z1, z2);
        System.out.println("z1 = " + z1);
        System.out.println("z2 = " + z2);
        System.out.println("a = " + a);
    }
}
```

Auf `stdout` wird ausgegeben:

Aufgabe 3 (10 Punkte)

Die Potenzierungsfunktion für zwei Zahlen $m, n \in \mathbb{N}_0$ kann durch fortgesetzte Multiplikation realisiert werden:

$$m^n := \text{pot}(m, n) = \begin{cases} m \cdot m^{n-1} & \text{falls } n > 0 \\ 1 & \text{sonst} \end{cases}$$

Geben Sie eine iterative und eine rekursive Implementierung der Potenzierungsfunktion in Java an. Verwenden Sie dabei keine Methoden der Klasse `java.lang.Math`.

```
// Iterative Variante der Potenzierungsfunktion
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
```

```
// Rekursive Variante der Potenzierungsfunktion
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
-------------------------------------------------------------
```

Aufgabe 4 (16 Punkte)

Die Elemente einer einfach verketteten Liste sollen mit Hilfe der folgenden Klasse modelliert werden:

```
class ListElement {
    int value;
    ListElement next;
}
```

Die verkettete Liste enthält die eingefügten Zahlen der Größe nach aufsteigend sortiert. Schreiben Sie eine Methode `insert`, die eine Zahl vom Typ `int` in die verkettete Liste einfügt, dabei die Sortierreihenfolge aber erhält. Beachten Sie als Sonderfälle das Einfügen in eine leere Liste sowie das Einfügen am Listenanfang.

```
public class SortedList {
    ListElement head = null;

    // Methode insert
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
}
```

Aufgabe 5 (16 Punkte)

1. Eine Grammatik ist definiert als 4-Tupel (V_N, V_T, S, R). Benennen und definieren Sie die vier Bestandteile einer allgemeinen Grammatik:

2. Gegeben sind folgende Grammatikregeln mit dem Startsymbol `S`:

   `S`→`A`
   `A`→`a b`
   `A`→`a A b`

   (a) Geben Sie die Mengen V_N und V_T an.

   (b) Geben Sie die Sprache L an, die von dieser Grammatik erzeugt wird.

   (c) Zeichnen Sie den Ableitungsbaum für das Wort `aaaabbbb` der Sprache L.

Aufgabe 6 (24 Punkte)

Im Verwaltungssystem eines Unternehmens sollen Informationen über die Kunden und die Mitarbeiter der Firma gespeichert werden.

1. Für jede Person soll der Vorname, der Nachname und die Adresse bestehend aus Straße, Hausnummer, Postleitzahl und Ort gespeichert werden. Die Werte sollen beim Anlegen der Daten einer Person über den Konstruktoraufruf gesetzt werden. Das Lesen der Daten darf ausschließlich über die Schnittstelle nach außen erfolgen, die durch folgende Methoden gegeben ist:

   - `getName()`: gibt den vollständigen Namen der Person in der Form

     `<Vorname> <Nachname>`

     zurück
   - `getAddress()`: liefert die Adresse der Person in der Form

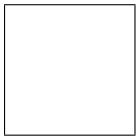
     `<Straße> <Hausnummer>`
     `<PLZ> <Ort>`

     zurück

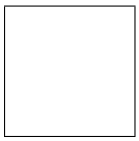
   Implementieren Sie die Klasse `Person` mit der beschriebenen Funktionalität:

```
public class Person {
    // Attribute
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------

    // Konstruktor
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
```

```
    // Methode getName()
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
```

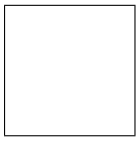


```
    // Methode getAddress()
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
    ------------------------------------------------------------
}
```

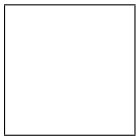


2. Zusätzlich zu den Persondaten soll bei einem Mitarbeiter das Einstiegsjahr gespeichert werden. Das Einstiegsgehalt eines Mitarbeiters beträgt 38 500,- €. Alle zwei Jahre erhöht sich das Gehalt um 139,50 €. Gehen Sie der Einfachheit halber davon aus, dass Mitarbeiter immer nur zum 1.1. eines Jahres eingestellt werden. Modellieren Sie eine Klasse `Mitarbeiter` unter Verwendung der Klasse `Person`. Stellen Sie eine Methode `getSalary(int year)` zur Verfügung, die das Gehalt für das als Parameter übergebene Jahr berechnet.

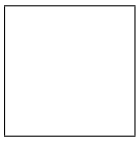
```
// Klassendefinition Mitarbeiter
```



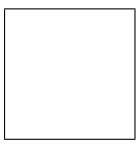
```
-------------------------------------------------------------

    // Attribute
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
```



```
    // Konstruktor
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
```



```
    // Methode getSalary
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
    -------------------------------------------------------------
```



```
-------------------------------------------------------------
```

3. Für einen Kunden soll neben den Personendaten intern gespeichert werden, für wie viel Euro er schon bei dem Unternehmen eingekauft hat. Die Klasse `Kunde` soll demnach eine Methode `buy(double amount)` zur Verfügung stellen, um auf dem Konto des Kunden den aktuellen Kauf über <`amount`> € zu vermerken. Als Premiumkunde zählen Kunden, die bereits für über 4 999 € eingekauft haben. Die Klasse `Kunde` soll eine Methode `isPremium` zur Verfügung stellen, die dies ermittelt.

   Zeichnen Sie ein UML-Klassendiagramm für die Klassen `Person`, `Mitarbeiter` und `Kunde`, das die in der Aufgabenstellung angegebenen Informationen enthält.