

Name: _____

Matrikelnummer: _____

RWTH AACHEN
Lehrstuhl für Informatik 10 (Medieninformatik)

Programmierung für Alle (Java)

Dozent: Prof. Dr. Jan Borchers

Probeklausur
Dezember 2006

Das ist eine **40-Minuten-Klausur** mit insgesamt **40 Punkten**: Sie sollten **für jeden Punkt etwa eine Minute verwenden**.

Die Klausur umfaßt **5 Seiten** mit **7 Fragen**. Bitte überprüfen Sie jetzt, ob ihre Klausur vollständig ist.

Bitte **schreiben Sie jetzt Ihren Namen auf jede Seite** der Klausur, da sich die Seiten während der Korrektur leicht voneinander lösen können.

Schreiben Sie **nicht** mit Bleistift oder mit roter Tinte. Was nicht bewertet werden soll, kennzeichnen Sie bitte durch Durchstreichen.

Verwenden Sie **nur** das von uns zur Verfügung gestellte Papier. Trennen Sie die Blätter nicht voneinander.

Es sind **keine Hilfsmittel** während der Klausur zugelassen.

Punktetabelle für die Korrektur:

	Punkte		Punkte		Punkte
1	(9)	3	(5)	5	(7)
2	(5)	4	(4)	6	(4)
Zw.-Summe		Zw.-Summe		7	(6)
				Zw.-Summe	
Summe (40)					

1. Verständnisaufgaben

- (a) [1 Punkt] Was versteht man unter einem Algorithmus?
- (b) [1 Punkt] Erklären Sie den Unterschied zwischen *Parametern* und *Argumenten*.
- (c) [2 Punkte] Erläutern Sie kurz die Eigenschaften und Bit-Tiefe der 8 elementaren Datentypen.
- (d) [2 Punkte] Beschreiben Sie drei wichtige Techniken der Softwareentwicklungsmethode *Extreme Programming*.
- (e) [3 Punkte] Machen Sie unter einer korrekten Codezeile einen Hacken. Markieren Sie fehlerhafte Codezeilen entweder als *Compilerfehler* oder *Laufzeitfehler* und korrigieren Sie alle vorhandenen Fehler.
- `int zahl = Integer.parseInt ("vier");`
 - `int x = 34.5;`
 - `public static void main (string[] args) { ... }`
 - `if (x > 7) x++; else x = x - 1;`
 - `public double quadrat (int x) return x * x;`
 - `float $dollar$;`

2. [5 Punkte] Schreiben Sie ein vollständiges Programm *Ausgabe.java*, das alle Argumente, die Sie beim Programmaufruf auf der Kommandozeile angeben, in umgekehrter Reihenfolge auf einer Zeile wieder ausgibt. Tipp: Die übergebenen Argumente werden in dem Array *String[] args* gespeichert.

Beispiel:

java Ausgabe Ich mag Java

Java mag Ich

3. [5 Punkte] Schreiben Sie ein Methode *convertSec*, die eine Anzahl an Sekunden erwartet und die Anzahl der entsprechenden Stunden, Minuten und Sekunden auf den Bildschirm ausgibt. Tipp: Benutzen Sie den Modulo-Operator.

Beispiel:

Eingabe 1234

Ausgabe 0:20:34

4. [4 Punkte] Schreiben Sie eine öffentliche Methode mit dem Namen *c2f*, die einen *double-Wert c* als Parameter erwartet und einen *double-Wert* als Ergebnis zurückliefert. Diese Methode soll einen Temperaturwert von Celsius nach Fahrenheit konvertieren. Die Berechnungsvorschrift mit *f* für den Wert in Fahrenheit und *c* für den Wert in Celsius lautet $f = 9 * c / 5 + 32$. Tipp: Achten Sie darauf, daß Ihr Programm beim Rechnen keine Nachkommastellen "verliert".

5. (a) [2 Punkte] Welche mathematische Funktion berechnet die folgende Methode?
Achtung: Sehen Sie genau hin!

```
public int berechne1 (int n) {
    int res = 0;
    for (int y = 0; y < n; y++) {
        res = res + n;
    }
    return res;
}
```

- (b) [3 Punkte] Welchen Wert liefert der folgende Methodenaufruf **berechne2 (3,7)**?

```
public int berechne2 (int x, int y) {
    int i;
    int res;
    if (x < y) {
        i = x;
        res = y - x;
    }
    else {
        i = y;
        res = x - y;
    }
    res = res + 2 * i;
    return res;
}
```

- (c) [2 Punkte] Wie oft wird die innere Schleife der Methode durchlaufen?

```
public int berechne3 (int x, int y) {
    int res = 0;
    for (int a = 0; a < x; a++) {
        for (int b = 0; b < y; b++) {
            res = res + 1;
        }
    }
    return res;
}
```

6. Gegeben sind die folgenden Klassen und Objekte.

```
class Vogel { ... }  
class Adler extends Vogel { ... }  
class Pinguin extends Vogel { ... }
```

```
Vogel vogel = new Vogel ();  
Adler adler = new Adler ();  
Pinguin pinguin = new Pinguin ();
```

(a) [2 Punkte] Welche der folgenden Zuweisungen erzeugen einen *Compilerfehler*? Welche Zuweisungen könnten einen *Laufzeitfehler* erzeugen? (Betrachten Sie jede Zuweisung unabhängig von den anderen Zuweisungen.)

- `vogel = adler;`
- `adler = vogel;`
- `adler = (Adler) vogel;`
- `adler = pinguin;`
- `adler = (Adler) pinguin;`

(b) [2 Punkte] Betten Sie die fehlerhaften Zuweisungen so in Testanweisungen, daß keine *Laufzeitfehler* mehr möglich sind.

7. Ein Student hat einen Vornamen, einen Nachnamen, eine Matrikelnummer, eine Semesterzahl und ein Studienfach.

(a) [4 Punkte] Deklarieren Sie eine öffentliche Klasse für einen Studenten. Das “Wissen” des Studenten soll für alle Unterklassen sichtbar sein aber nicht für alle Klassen überall. Implementieren Sie für alle zugängliche Getter- und Setter-Methoden für den Nachnamen und die Matrikelnummer. Wird keine gültige Matrikelnummer zwischen 100000 und 300000 eingegeben, dann wird die Matrikelnummer auf 0 gesetzt.

(b) [2 Punkte] Erstellen Sie ein Studenten-Objekt und initialisieren Sie sein vollständiges “Wissen”. Benutzen Sie auch die Setter-Methoden, die Sie nicht explizit in Teil (a) implementiert haben.