

Name	Matrikelnummer

Klausur zur Veranstaltung Software-Entwicklung 3

Sommersemester 2004
Hessische VWA
Dr. Alexandra Roder, Bernd Ulmann

Hinweise:

- Die Klausur besteht aus 6 Aufgaben mit insgesamt 15 Teilaufgaben.
- Insgesamt sind 120 Punkte erreichbar, jede Teilaufgabe entspricht 8 Punkten.
- Als Hilfsmittel sind alle „unbelebten Hilfsmittel“, d.h. insbesondere Ihre Vorlesungsmitschriften sowie das Skript, zugelassen.
- Lesen Sie jede Aufgabe vor Beginn ihrer Bearbeitung gründlich (!) durch!
- Bitte schreiben Sie leserlich! (Dankeschön! :-)
- Verwenden Sie bitte keinen (!) Bleistift!
- Vergessen Sie nicht, Primärschlüsselbezeichnungen zu unterstreichen!

Viel Erfolg!

Aufgabe 1

Sie sind Leiter eines Projektes für eine Dialogfunktion, das voraussichtlich einen Umfang von 24000 Zeilen Code aufweisen wird. Zu dessen Durchführung stehen Ihnen vier Programmierer zur Verfügung, die Monatsleistungen von 600, 500, 300 und 200 Zeilen erbringen.

1. Berechnen Sie den Aufwand in Personenmonaten, wenn man von einer durchschnittlichen Programmierleistung ausgeht.

2. Berechnen Sie die optimale Laufzeit s des Projektes mit Hilfe der Formel

$$s \approx \frac{5}{2} \text{Anzahl-Personenmonate}^{0.36}$$

3. Berechnen Sie die benötigte Anzahl von Programmierern, um das Projekt so schnell wie möglich zum Abschluß zu bringen, wenn die durchschnittliche Programmierleistung gleich bleibt.

Aufgabe 2:

Im weiteren sollen Sie einige Fragen zu dem folgenden kleinen Perl-Programm, das absichtlich kommentarlos ist, beantworten:

```
use strict;
use warnings;

my @zeilen;

# Belege das Feld:
$zeilen[0] = 'Zebra';
$zeilen[1] = 'Affe';
$zeilen[2] = 'Loewe';
$zeilen[3] = 'Baer';
$zeilen[4] = 'Pinguin';

my $anzahl = 5;

my $aktion = 1;

while ($aktion)
{
    $aktion = 0;
    for (my $i = 0; $i < $anzahl - 1; $i++)
    {
        if ($zeilen[$i] ge $zeilen[$i + 1])
        {
            ($zeilen[$i], $zeilen[$i + 1]) = ($zeilen[$i + 1], $zeilen[$i]);
            $aktion = 1;
        }
    }
}

for (my $i = 0; $i < $anzahl; $i++)
{
    print "$zeilen[$i]\n";
}
```

1. Welche Ausgabe erzeugt das oben stehende Programm, wenn es mit Hilfe des Perl-Interpreters laufen gelassen wird?

2. Was tut das Programm generell, d.h. was täte es, wenn das Feld `@zeilen` mit anderen Werten gefüllt wäre?

3. Welchen Berechnungsaufwand zieht dieses Programm, bezogen auf die Anzahl von Elementen in `@zeilen` nach sich? D.h. wie verlängert sich seine Laufzeit, wenn `@zeilen` beispielsweise doppelt, dreimal, viermal, etc. so viele Einträge enthält?

Aufgabe 3:

Lesen (und verstehen) Sie das folgende kleine Perl-Programm:

```
#
# Das folgende Programm soll die Summe aller
# natuerlichen Zahlen von 1 bis einschliesslich
# 100 berechnen und ausgeben.
#

use strict;
use warnings;

my $ende = 100;

for (my $i = 1; $i < $ende; $i++)
{
    $summe = $summe + 1;
}

print "$summe\n";
```

1. In dem obenstehenden Programm sind drei Fehler versteckt – finden und benennen Sie sie (diese Aufgabe zählt entsprechend auch dreifach, d.h. pro gefundenem Fehler erhalten Sie 8 Punkte).

Aufgabe 4:

Welche der folgenden Dokumente sind während der Analysephase eines Projektes zu erstellen? Markieren Sie die entsprechenden Einträge durch Ankreuzen.

<input type="checkbox"/>	Lasten-/Pflichtenheft
<input type="checkbox"/>	Dokumentation
<input type="checkbox"/>	Glossar
<input type="checkbox"/>	Programmcode
<input type="checkbox"/>	Projektabschlußbericht

Aufgabe 6:

1. Zeichnen Sie ein Klassendiagramm für eine allgemeine Klasse *Fahrzeuge*, die über die Attribute *Hersteller*, *Fahrer*, *Geschwindigkeit* und *Klimaanlage* verfügt.

2. Zeichnen Sie nun zwei hiervon abgeleitete Klassen zusammen mit dieser Hauptklasse in ein Klassendiagramm ein. Die beiden Unterklassen sind *U-Boote*¹ und *Rennwagen*².

Diese beiden Unterklassen verfügen über die zusätzlichen Attribute *Tauchtiefe* beziehungsweise *Kosmetikspiegel* (nicht (!) für U-Boote!).

¹Auf ausdrücklichen Wunsch Herrn Ulmanns!

²Auf ausdrücklichen Wunsch von Frau Dr. Roder!