

<b>Name:</b>
<b>Matrikelnr.:</b>

## **Klausur im Modul IT 1, März 2008. Studiengänge WIB und ESB 1. Semester**

### **Hinweise**

- Dauer der Klausur: **90 Minuten**
- Erreichbare Punktzahl: 240 Punkte
- Der Aufgabensatz umfasst 14 Seiten.
- **Die Aufgaben sind auf den Aufgabenblättern in dem dafür vorgesehenen Platz zu beantworten.** Nur im Ausnahmefall, z.B. falls aufgrund größerer Korrekturen der Platz nicht mehr reicht, kann die Blattrückseite zu Hilfe genommen werden. Dies muss dann bei der Aufgabe deutlich vermerkt werden ("Siehe Rückseite!").
- Auf den Aufgabenblättern sind Name und Matrikelnummer einzutragen.

### **1. Programmieren (Rathke)**

**(insgesamt 120 Punkte)**

Das folgende Programm enthält den Enumerationstyp „Make“ und die Klassen „Vehicle“, „Car“ und „Race“. Alle Aufgaben in diesem Abschnitt beziehen sich auf dieses Programm.

```
enum Make {VOLKSWAGEN, AUDI, OPEL, DAIMLER, BMW, PORSCHE}
```

```
class Vehicle {  
    int speed = 0;  
    int location = 0;  
  
    public int getSpeed() {  
        return speed;  
    }  
  
    public int getLocation() {  
        return location;  
    }  
}
```

```
public class Car extends Vehicle {  
    private Make make = null;  
    private String model = "";  
    private java.awt.Color color = null;  
    private int serialNr = 0;  
  
    public void accelerate (int speedIncr) {  
        speed += speedIncr;  
    }  
  
    public void go (int seconds) {  
        location += speed*seconds;  
    }  
  
    public String toString () {  
        return make + " " + model + " " + serialNr;  
    }  
}
```



<b>Name:</b>
<b>Matrikelnr.:</b>

c) Nennen Sie 6 **zusätzliche** Operatoren (die in dem Programm **nicht (!)** verwendet werden) aus 3 verschiedenen Kategorien und nennen Sie auch diese Kategorien (6 Punkte).

Operator	Kategorie

**Teilaufgabe 1.2**

**(6 Punkte)**

Nennen Sie sämtliche Attribute der Klasse „Car“

Attribut

**Teilaufgabe 1.3**

**(16 Punkte)**

Programmieren Sie den fehlenden Konstruktor.

<b>Name:</b>
<b>Matrikelnr.:</b>

### Teilaufgabe 1.4

**(16 Punkte)**

Re-Implementieren Sie die Klasse Car, indem Sie ein Interface namens „Driveable“ definieren, das von jeder Klasse implementiert werden kann, die die in Car implementierten Methoden „accelerate“ und „go“ realisiert.

a) Wie ist Definition dieses Interface? (10 Punkte)

b) Wie ist die neue Signatur der Klasse Car? (6 Punkte)

### Teilaufgabe 1.5

**(12 Punkte)**

Ergänzen Sie die Definition von Car um Code-Elemente, die es erlauben, über die Anzahl der erzeugten Car-Instanzen buch zu führen.

a) Welche zusätzliche Variable benötigen Sie und wie sieht deren Deklaration aus? (4 Punkte)

b) Um welche Zeile sollte der Konstruktor von „Car“ ergänzt werden, um die Anzahl der erzeugten Instanzen zu aktualisieren? (8 Punkte)

### Teilaufgabe 1.6

**(14 Punkte)**

Skizzieren Sie eine typische Ausgabe der „main-Methode“ der Klasse „Race“.

<b>Name:</b>
<b>Matrikelnr.:</b>

### Teilaufgabe 1.7

**(18 Punkte)**

Schreiben Sie eine Java-Kontrollfluss-Anweisung, in der 100 Car-Instanzen erzeugt, auf eine Geschwindigkeit von 50 beschleunigt und dann für 10 Sekunden fahren gelassen werden.

<b>Name:</b>
<b>Matrikelnr.:</b>

## 2. Betriebssysteme und Rechnerarchitekturen (Langenbach) (insgesamt 48 Punkte)

### Teilaufgabe 2.1 (24 Punkte)

Kreuzen Sie bitte ausschließlich alle richtigen Aussagen an. Hinweis: In jedem Aussagenblock gibt es mindestens eine richtige und mindestens eine falsche Aussage. Wenn Sie alle Kreuzchen an der richtigen Stelle gesetzt haben, erhalten Sie 24 Punkte. Jeder Fehler, d.h. jedes fälschlicherweise angekreuzte bzw. nicht angekreuzte Kästchen, bewirkt 2 Punkte Abzug. Bei 12 Fehlern und mehr erbringt diese Teilaufgabe 0 Punkte.

#### a) Grundlagen

- Eine spezifische Auslegung des Mooreschen Gesetzes besagt, dass sich die Rechenleistung pro 1.000 \$ in den vergangenen 40 Jahren jährlich etwa vervierfacht hat.
- Die Taktfrequenz eines Rechners berechnet sich nach folgender Formel:  
Taktfrequenz =  $1/\text{Periodendauer des Taktsignals}$
- Eine bedeutende Erkenntnis der Booleschen Algebra ist, dass aus den drei Operatoren Konjunktion, Disjunktion und Negation alle anderen Funktionen durch Kombination erzeugt werden können.
- Taktpegelgesteuerte Flip-Flops können ihren Zustand nur an einer Taktflanke ändern.

#### b) Prozessor

- Superskalare Architekturen basieren auf dem Konzept des Pipelining.
- Der klassische Von-Neumann-Rechner zählt zur Klasse der SIMD- (Single Instruction Multiple Data) Architekturen nach Flynn.
- Die Hauptplatine eines PCs lässt sich in North- und Southbridge unterteilen.
- Die Leistungsfähigkeit eines Rechners wird u. a. determiniert durch die interne Verarbeitungsbreite und die Taktfrequenz.

#### c) Speicher und Eingabe/Ausgabe

- Hauptspeichermodulen der aktuellen Generation liegt das Prinzip der Übertragung zweier Datenpakete pro Taktzyklus zugrunde.
- Die Entdeckung des 2007 mit dem Nobelpreis ausgezeichneten Riesenmagnetowiderstands ermöglichte wesentliche kürzere Zugriffszeiten auf magnetische Speicherplatten.
- Schnittstellen sind Bestandteile von Ein-/Ausgabegeräten, denen die Steuerung von Unterbrechungsanforderungen obliegt und die den Datentransfer regeln und den Buszugriff realisieren.
- Über digitale Graphikkartenausgänge (z. B. DVI) können ausschließlich digitale Bildschirme angeschlossen werden.

<b>Name:</b>
<b>Matrikelnr.:</b>

**d) Prozesse und Threads**

- Wartet ein Prozess im Zustand „blockiert“ auf ein Ereignis und selbiges tritt ein, wird er üblicherweise zunächst in den Zustand „bereit“ versetzt, ehe ihm der Prozessor zugeteilt werden kann.
- Der Prozessdeskriptor (Prozesskontrollblock) enthält u. a. Informationen zum jeweiligen Elternprozess sowie zu etwaigen Kindprozessen.
- Multitasking und Multithreading sind zwei Konzepte, die sich gegenseitig ausschließen.
- Ein neuer Prozess kann erzeugt werden, indem ein bereits existenter Prozess kopiert wird.

**e) Scheduling**

- Prozessorvergabestrategien mit Verdrängung finden insb. in Echtzeitsystemen verwendet, da dort auf kurzfristig auftretende Erfordernisse ggf. mit einem sofortigen Prozesswechsel reagiert werden muss.
- Die Strategie „Shortest Remaining Time Next“ kann interpretiert werden als Variante der Strategie „Shortest Job First“ mit Verdrängung.
- Die Strategie „Highest Response Ratio Next“ ist speziell darauf konzipiert, Prozesse mit langer Gesamtlaufzeit zu favorisieren.
- Bei Mehrprozessorsystemen ist grundsätzlich kein Scheduling erforderlich, da in diesen die Prozesse nicht um einen einzigen Prozessor konkurrieren müssen.

**f) Interprozesskommunikation und Deadlocks**

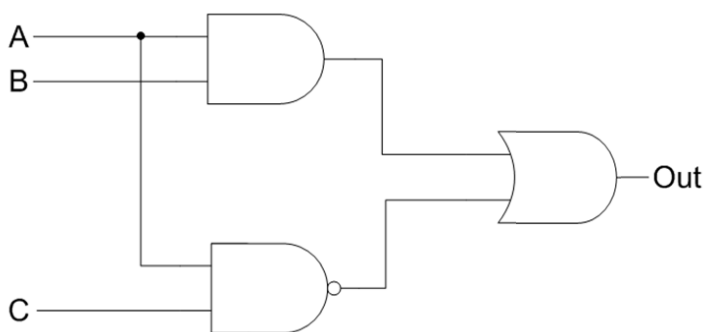
- Die Synchronisation des Informationsaustausches zwischen zwei Threads ist grundsätzlich unproblematischer als der Informationsaustausch zwischen zwei Prozessen.
- Das Prinzip des wechselseitigen Ausschlusses dient der Vermeidung von Konflikten im Kontext von Wettkampfbedingungen bei der Interprozesskommunikation.
- Eine Menge von Prozessen befindet sich in einem Deadlock-Zustand, wenn jeder Prozess aus der Menge auf ein Ereignis wartet, das nur ein anderer Prozess aus der Menge auslösen kann.
- Wird das Auftreten von Deadlocks in einem System nicht durch einen geeigneten Mechanismus explizit vermieden, wird es während der Systemlaufzeit zwangsläufig irgendwann zu einem Deadlock kommen.

**Name:**

**Matrikelnr.:**

**Teilaufgabe 2.2 (12 Punkte)**

Gegeben sei folgendes Schaltnetz:



Ermitteln Sie bitte die Funktionstabelle für das dargestellte Schaltnetz. Nutzen Sie hierzu bitte folgende Vorlage:

A	B	C	Out

*Raum für etwaige Zwischenberechnungen:*



<b>Name:</b>
<b>Matrikelnr.:</b>

**Teilaufgabe 2.3 (12 Punkte)**

Gegeben sei ein System, in welchem zum Zeitpunkt 0 fünf Prozesse mit folgenden geschätzten individuellen Gesamtlaufzeiten erzeugt werden:

Prozess	Gesamtlaufzeit (in Zeiteinheiten)
A	4
B	2
C	5
D	4
E	1

- a) Bringen Sie zunächst die fünf Prozesse in die Reihenfolge, nach der sie gemäß der Scheduling-Strategie „Shortest Job First“ abgearbeitet werden. (2 Punkte)

**Reihenfolge nach Shortest Job First:**

---

- b) Berechnen Sie nunmehr anhand der gewählten Reihenfolge die mittlere Verweildauer (Wartezeit) der Prozesse im System: (4 Punkte)

**Berechnung der mittleren Verweildauer (Wartezeit):**

- c) Welche Reihenfolge der Abarbeitung ergibt sich, wenn das Scheduling zwar weiterhin der Strategie „Shortest Job First“ folgt, die Prozesse B und E jedoch erst zum Zeitpunkt 3 im System erzeugt werden? (2 Punkte)

**geänderte Reihenfolge:**

---

**Name:**

**Matrikelnr.:**

- d) Berechnen Sie die mittlere Verweildauer (Wartezeit) der Prozesse im System erneut unter den in Teilaufgabe c) genannten Rahmenbedingungen: (4 Punkte)

**Berechnung der mittleren Verweildauer (Wartezeit) unter geänderten Rahmenbedingungen:**

<b>Name:</b>
<b>Matrikelnr.:</b>

**3. Datenbanken (Riekert)****(insgesamt 72 Punkte)****Teilaufgabe 3.1****(36 Punkte)****In einer Datenbank sollen folgende Sachverhalte über eine Schule gespeichert werden:**

1. Jeder *Schüler* hat ein *Geburtsjahr*, einen *Nachnamen* und einen *Vornamen*.
2. Jede *Klasse* hat eine eindeutige *Bezeichnung* (z.B. „4a“) und gehört einem *Zug* an (z.B. „technisch“).
3. Jeder *Lehrer* hat eine eindeutige *Personalnummer*, einen *Nachnamen* und einen *Vornamen*.
4. Jeder Schüler *gehört* genau einer Klasse *an*, jede Klasse umfasst mehrere Schüler.
5. Jede Klasse wird von genau einem Lehrer *betreut*. Jeder Lehrer kann maximal eine Klasse betreuen, es gibt aber auch Lehrer, die keine Klasse betreuen.
6. Jeder Lehrer *unterrichtet* in mehreren Klassen. Jede Klasse wird von mehreren Lehrern unterrichtet.

**a) (18 Punkte) Zeichnen Sie ein Entitäten-Beziehungsmodell, das die oben beschriebenen Sachverhalte wiedergeben kann.**

- Vermerken Sie bei den Beziehungsmengen auch die jeweils zutreffenden **Mächtigkeiten** (1, m, n). Verwenden Sie auch **Intervalldarstellungen** (z.B. 0..1), wenn dies sinnvoll ist.
- Bitte zeichnen Sie auch die Merkmale in Form von **Ellipsen** ein.

<b>Name:</b>
<b>Matrikelnr.:</b>

b) (18 Punkte) Entwerfen Sie eine Tabellenstruktur zur Aufnahme der oben beschriebenen Sachverhalte über die Schule.

Zeichnen Sie die Tabellen bitte entsprechend dem nachstehenden Muster:

Tabellenname

<u>Feldname</u>	Feldname	Feldname	...

Befolgen Sie dabei bitte folgende Hinweise:

- Verwenden Sie die nachfolgenden Tabellenskelette zur Darstellung der Tabellen. Eventuell nicht benötigte Tabellenskelette lassen Sie bitte frei!
- Versuchen Sie, mit möglichst **wenigen** Tabellen auszukommen
- Geben Sie jeder Tabelle einen geeigneten **Namen** und tragen Sie die Namen der Tabellenfelder ein.
- **Unterstreichen** Sie die Primärschlüsselfelder.
- Verbinden Sie die Fremdschlüsselfelder durch **Pfeile** mit den zugehörigen Primärschlüsselfeldern. Die **Pfeilspitzen sollen auf die Primärschlüsselfelder zeigen!**






<b>Name:</b>
<b>Matrikelnr.:</b>

**Teilaufgabe 3.2**

**(36 Punkte)**

In einer Datenbank befinden sich die Tabellen `stadt` und `land` mit den in der Abbildung dargestellten Inhalten.

**Stadt**

Id	Bezeichnung	Land	Bevoelkerung	Sprache
1	Berlin	DE	3.400.000	DE
2	Genf	CH	180.000	FR
3	Zürich	CH	350.000	DE

**Land**

Code	Bezeichnung	Bevoelkerung	Flaeche
CH	Schweiz	7.500.000	42.000
DE	Deutschland	82.000.000	357.000
FR	Frankreich	62.000.000	544.000

- a) (6 Punkte) Bitte tragen Sie das **Ergebnis** der folgenden SQL-Abfrage in das darunter stehende Tabellenskelett ein. Vergessen Sie bitte nicht die **Spaltenüberschriften** der Ergebnistabelle.

```
SELECT Bezeichnung, Bevoelkerung FROM Land
WHERE Flaeche < 500000 ORDER BY Bezeichnung;
```


- b) (6 Punkte) Bitte tragen Sie das **Ergebnis** der folgenden SQL-Abfrage in das darunter stehende Tabellenskelett ein. Vergessen Sie bitte nicht die **Spaltenüberschriften** der Ergebnistabelle.

```
SELECT Land, COUNT(*) AS Anzahl, MAX(Bevoelkerung) AS Maximum
FROM Stadt GROUP BY Land;
```


- c) (6 Punkte) Mit welcher SQL-Abfrage ermittelt man die **kleinste Fläche**, die bei den Ländern vorkommt?

--

<b>Name:</b>
<b>Matrikelnr.:</b>

- d) (6 Punkte) Gesucht ist eine Liste bestehend aus **Bezeichnung** und **Bevölkerung** aller **Städte**. Die Liste soll absteigend **nach der Bevölkerung** sortiert sein, also große Bevölkerungszahlen sollen zuerst kommen. Schreiben Sie eine SQL-Abfrage, die diese Liste ermittelt!

- e) (6 Punkte) Welche Information wird mit der nachfolgenden Abfrage ermittelt? Sie brauchen die Werte in der Ergebnistabelle nicht zu berechnen, Sie sollten nur deutlich machen, **welche Bedeutung das Ergebnis hat** und **in welcher Form es präsentiert wird**.

```
SELECT Bezeichnung, Bevoelkerung/Flaeche FROM Land
ORDER BY Bevoelkerung/Flaeche;
```

- f) (6 Punkte) Bitte tragen Sie das **Ergebnis** der folgenden SQL-Abfrage in das darunter stehende Tabellenskelett ein. Vergessen Sie bitte nicht die **Spaltenüberschriften** der Ergebnistabelle.

```
SELECT Bezeichnung, Bevoelkerung FROM Stadt
UNION
SELECT Bezeichnung, Bevoelkerung FROM Land;
```
