Aufgaben zu Kontrollstrukturen (Pseudocode)

Aufgabe Schleifen

Stellen Sie mit Hilfe von Pseudocode das Hochzählen einer Variablen jeweils mit Hilfe einer "for" bzw. Mit Hilfe einer "while" Schleife dar.

Aufgabe If-Then

Aus welchen Bestandteilen besteht eine if-then-else Bedingung. Nennen Sie die Voraussetzungen, die eintreten müssen, damit der else-Teil aufgerufen wird.

Aufgaben zu primitiven Datentypen

Aufgabe primitive Datentypen

Geben Sie zu jedem der Datentypen "float", "int" und "boolean" die Standardinitialisierungs-Belegungen an, wenn der Programmierer diese nicht selbst vorgibt.

Aufgabe Definition versus Deklaration

Worin unterscheiden sich Deklaration und Definition einer Variablen? An welchen Stellen in der Klasse wird normalerweise die Definition und wo die Deklaration vorgenommen.

Aufgaben zu Boolschen Ausdrücken

Aufgabe Boolsche Ausdrücke

Die Variable a und b sind boolsche Variablen unbekannter Belegung. Verkürzen Sie folgende boolsche Ausdrücke, soweit es möglich ist:

- `(!(a & a)) & true`
- `(  (true | b  ) | ( false | b)  )`

Aufgaben zur Objektorientierung Klasse

Aufgabe Klasse

Geben Sie die Bestandteile einer Klasse an. Nennen Sie außerdem 3 Gründe, weshalb die vormals nicht-objektorientierte (prozedurale) Softwareentwicklung an Grenzen stieß, welche den Einsatz von Objektorientierung notwendig machte.

Aufgabe Bestandteile einer Klasse

Welchen Zweck haben Zustandsvariablen in einer Klasse. Können diese von den Verhaltensmethoden angesprochen werden?

Aufgaben zu statischen und nicht-statischen Methoden

Aufgabe Funktionsbestandteile

Aus welchen Bestandteilen setzt sich eine Funktion zusammen?

Aufgabe statische Methode

Wann ist es sinnvoll eine statische Methode zu verwenden? Welche Vorteile und Nachteile hat diese im Vergleich zu Member-Methoden?

Aufgabe Funktionsaufrufe

Geben Sie die drei Konsolenausgaben des folgenden Programms an.

```
public class StaticFunctions {

	private static int f(int x)
	{
		return x;
	}

	private static int f(int x,int y)
	{
		return (x*y);
	}

	public static void main(String[] args)
	{
		System.out.println(f(1));
		System.out.println(f(2,3));
		System.out.println(  f(  f(1) , f(2,3) )  );
	}
}
```

Aufgaben zur Konstruktoren

Aufgabe Rolle der Konstruktoren

Wofür werden Konstruktoren genutzt? Welche wichtige Aufgabe kommt ihnen zu? Wann werden Konstruktoren aufgerufen?

Aufgabe Konstruktorenparameter

Wie viele verschiedene Konstruktoren (diese unterscheiden sich durch die Anzahl und Typ der Parameter) kann eine Klasse haben.

Aufgaben zur Sichtbarkeit

Aufgabe Sichtbarkeit

Warum ist es sinnvoll die Sichtbarkeit von Methoden und Variablen einzuschränken, d.h. von public auf private zu setzen?

Aufgabe Sichtbarkeit bei der Vererbung

Welche Sichtbarkeitsangaben werden benötigt, damit eine abgeleitete Klasse die Methoden der Mutterklasse aufrufen kann.

Aufgaben zur Instantiierung und Ansicht der Halde

Aufgabe Halde

Gegen ist die Klasse Bus und die Methode main, in der Instanzen der Klasse Bus erstellt werden. Geben Sie den Zustand der Halde nach Ausführung der Methode main an (bevor die Instanzen durch den Garbage-Collector gelöscht werden).

```
public class Bus
{
	int mBaujahr;
	String mKennzeichen;
	int mSitzplätze;


	public static void main(String[] args)
	{

		Bus bus1=new Bus();
		Bus bus2=new Bus();
		Bus bus3=new Bus();
	}
}
```

Aufgabe Halde mit Array

Gegen ist die Klasse Bus und Busfahrgast und die Methode main, in der Instanzen der Klasse Bus und Busfahrgast erstellt werden. Geben Sie den Zustand der Halde nach Ausführung der Methode main an (bevor die Instanzen durch den Garbage-Collector gelöscht werden).

```
public class Busfahrgast {
	public String mTicket;
	public long mEinstiegszeit;
}
```

```
public class Bus
{
	private int mBaujahr;
	private String mKennzeichen;
	private int mSitzplaetze;
	private int mAnzahlFahrgaeste;
	private Busfahrgast[] mFahrgaeste;


	public Bus(int aSitzplaetze)
	{
		mSitzplaetze=aSitzplaetze;
		mFahrgaeste=new Busfahrgast[mSitzplaetze];
		mAnzahlFahrgaeste=0;
	}

	protected void einsteigen(Busfahrgast aFahrgast)
	{
		mFahrgaeste[mAnzahlFahrgaeste]=aFahrgast;
		mAnzahlFahrgaeste++;
	}

	public static void main(String[] args)
	{

		Busfahrgast marie=new Busfahrgast();
		Busfahrgast peter=new Busfahrgast();
		Bus bus=new Bus(10);
		bus.einsteigen(marie);
		bus.einsteigen(peter);
	}
}
```

Aufgabe Begriffe der Objektorientierung und UML

Geben Sie den Unterschied zwischen Klasse und Objektinstanz an. Beschreiben Sie an einem Beispiel beide Begriffe mit Hilfe des UML-Klassendiagrams.

Aufgabe UML Klassendiagramm

Geben Sie das UML Klassendiagram der Klassen aus Aufgabe "Halde mit Array" an.

Aufgabe Entwurf mit UML

Zur Verwaltung von ICE-Zügen in einem Anwendungsprogramm möchte ein deutsches Bahn-Unternehmen Objektorientierte Programmierung verwenden. Folgende Anforderungen sind bekannt:

Die Klasse IceFlotte besitzt ein Array des Types Train auf die mit Hilfe der Methode addTrain und getTrain(int index) von “außen” zugegriffen werden soll.

Die ICE-Züge sollen demnach als Objekte der Klasse Train in schon erwähntem Array gespeichert

werden.

Die Klasse Train soll neben dem Namen des ICE (vom Typ String) nun auch eine Variable hatKlimaAnlage (vom Typ boolean) enthalten. Auf beide Variablen soll ausschließlich mit Hilfe von get- und set Methoden zugegriffen werden.

Entwerfen Sie ein auf diese Anforderungen passendes UML-Klassendiagramm der technischen Spezifikation. Achten Sie hierbei auch auf die Darstellung der Sichtbarkeit der Methoden und Variablen.

Aufgabe package

Warum wird empfohlen Packages zu verwenden? Wo muss die Zugehörigkeit einer Klasse zu einem Package eingetragen werden?

Aufgabe Vererbung

Welche Ausgaben erscheinen auf der Konsole bei den 4 Klassen und ausgeführter main-Funktion:

```
package klausur_vererbung;

public class A {

	private String ausgabe()
	{
		return "Das ist Klasse A";
	}
}
```

```
package klausur_vererbung;

public class B extends A {

	protected String ausgabe()
	{
		return "das ist Klasse B";
	}
}
```

```
package klausur_vererbung;

public class C extends B{

	public String ausgabe()
	{
		return "Das ist Klasse C";
	}
}
```

```
package klausur_vererbung;

public class D extends C
{

}
```

Die Klasse *Start* vom Package *klausur_start* enthält die folgende main-Methode:

```
public static void main(String[ ] args) {
		D d=new D(); System.out.println(d.ausgabe());
		C c=new C(); System.out.println(c.ausgabe());
		B b=new B(); System.out.println(b.ausgabe());
		A a=new A(); System.out.println(a.ausgabe());
	}
```

Aufgabe Zugriffsrechte managen mit Vererbung

Für die Verwaltung von Zugriffsrechten der Rollen Person,Direktorin, Kassiererin, Putzmann, Kundin soll ein objektorientiertes Design genutzt werden.

Folgen Aktionen sind möglich: Bank_betreten(), Safe_öffnen(), Geld_auszahlen(), Fenster_öffnen(), Karte_stechen(), Geld_einzahlen(). Überlegen Sie, welche Aktion für welche Rolle geeignet ist und tragen sie die Aktionen in das unten stehende Klassendiagramm ein.