KLAUSUR 1 SAE E1FI 27.11.2013 Zeit: 45 Minuten

Name, Vorname:

Hilfsmitttel:
keine

Hinweise (unbedingt beachten):

→ Alle Aufgaben müssen bearbeitet werden.
→ Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
→ Aufgabenblätter bitte auch abgeben.
→ Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
→ Die rote Farbe darf nicht benutzt werden.
→ Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
→ Selbsterklärende Variablennamen benutzen.
→ Programme müssen benutzerfreundlich sein.
→ EVA-Prinzip muss benutzt werden.
→ Einrücken der entsprechenden Programmteile.
→ Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
→ In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
→ Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 3+4+4 P

a) Welche 3 Möglichkeiten (3 Worte nennen) gibt es, einen Algorithmus darzustellen?
b) Erklären Sie die Begriffe Syntax und Semantik?
c) Was ist ein Compiler ?

2) 5 P

a) Wie viele Zustände genau kann man mit 8 Byte angeben? (als Hochzahl schreiben)
b) Näherungsweise Berechnung!

3) 10P

Der folgende syntaktisch korrekte Teil eines Java-Programm soll die kleinste Zahl (dreier Zahlen z1, z2. z3) berechnen und in der Variablen min speichern.
Wenn das Programm korrekt ist, schreiben Sie "Der Algorithmus ist korrekt" und geben 5 Beispiele (mit konkreten Werten für z1, z2, z3 und dem berechneten Wert von min) an, wo er korrekt wird.
Wenn das Programm nicht korrekt ist, schreiben Sie " Der Algorithmus ist nicht korrekt" und geben ein **konkretes** Beispiel (mit konkreten Werten für z1, z2, z3 und dem berechneten Wert von min) an, bei dem der Algorithmus falsch wird.

```
int main(){
  ...
  min = 0;
  if(z1<z2){
    min=z1;
  }
  if(z3<min){
    min=z3;
  }
  ...
}
```

4) 12P

Erstellen Sie ein Struktogramm, das das Minimum und das Maximum dreier Zahlen berechnet und auf dem Bildschirm ausgibt.
Es dürfen nur einseitige Verzweigungen benutzt werden!

5) 12P

Durch den folgenden Algorithmus (Flußdiagramm) soll von 50 Schülern jeweils der Prozentsatz der erreichten Punkte bzgl. der Gesamtpunktzahl in einer Klassenarbeit errechnet und ausgegeben werden.

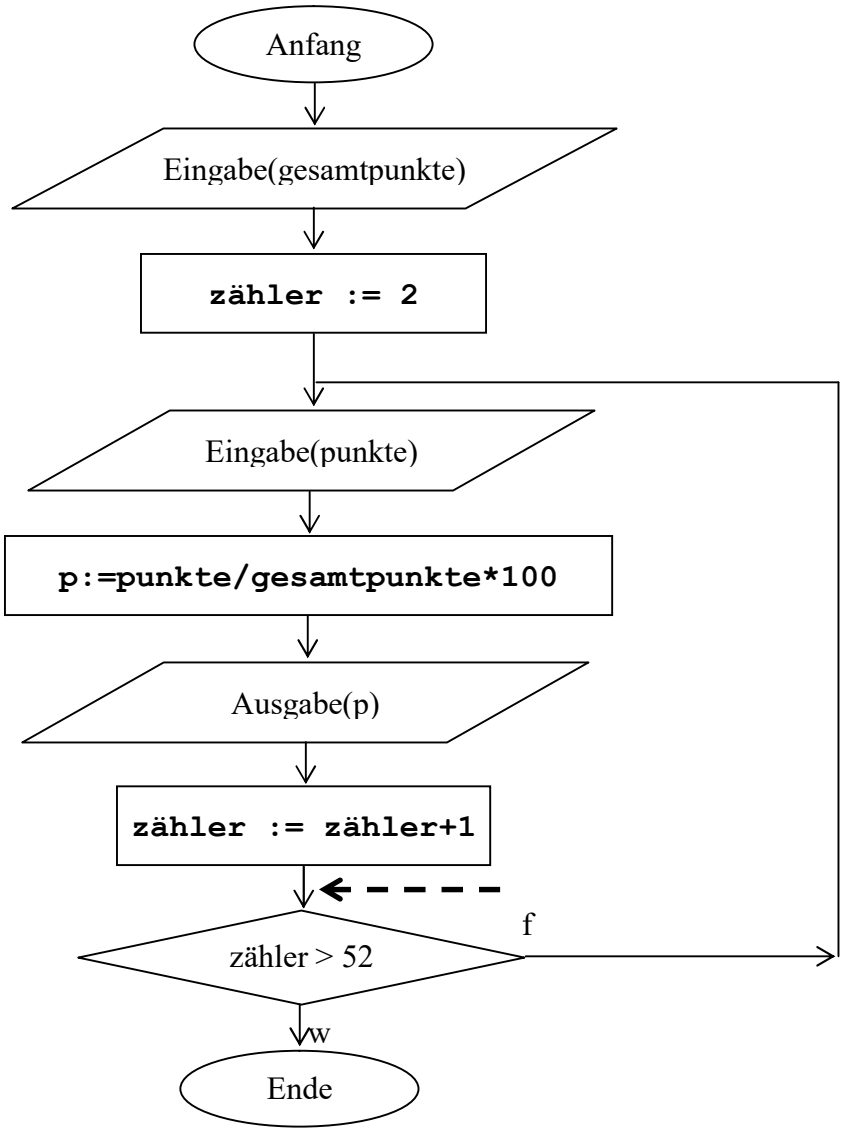
Dazu wird das Programm immer wieder an der mit <--- bezeichneten Stelle vor der Verzweigung (bei jedem Schleifendurchgang) gedanklich angehalten (Protokoll) und die Werte der Variaben zähler und die Anzahl der ausgegebenen Punkte der Schüler protokolliert.

a) Tragen Sie dazu in der Tabelle unten die Werte von zähler und die Anzahl der ausgegebenen Schüler bei den ersten 5 Durchgängen ein.

b) Welche Werte haben zähler und Anzahl der ausgegebenen Schüler, wenn das Programm das **letzte** Mal an die mit <---bezeichnete Stelle kommt?

c) Ist die folgende Lösung korrekt? Begründen Sie mit der Tabelle unten!

Anzahl der ausgegebenen Schüler	Wert der Variablen zähler



Lösungen:

1)

a) 3P

Flussdiagramm, Struktogramm, Programm

b) 4P

Die Syntax definiert die äußeren Formgesetze dieser Programmiersprache (ähnlich den grammatikalischen Regeln einer natürlichen - wie z.B. der englischen- Sprache).
Die Semantik ist der Bedeutungsinhalt (ähnlich der Bedeutung der einzelnen Worte einer natürlichen - wie z.B. der italienischen - Sprache) der einzelnen Objekte einer Programmiersprache.

c) 4P

Ein Compiler ist ein Übersetzer, der einen in einer höheren Programmiersprache formulierten Text (ein sogenanntes Programm) in einen aus Maschinenbefehlen bestehenden Text (einem sogenannten Maschinenprogramm) verwandelt. Dieses kann dann vom Mikroprozessor abgearbeitet (ausgeführt) werden.

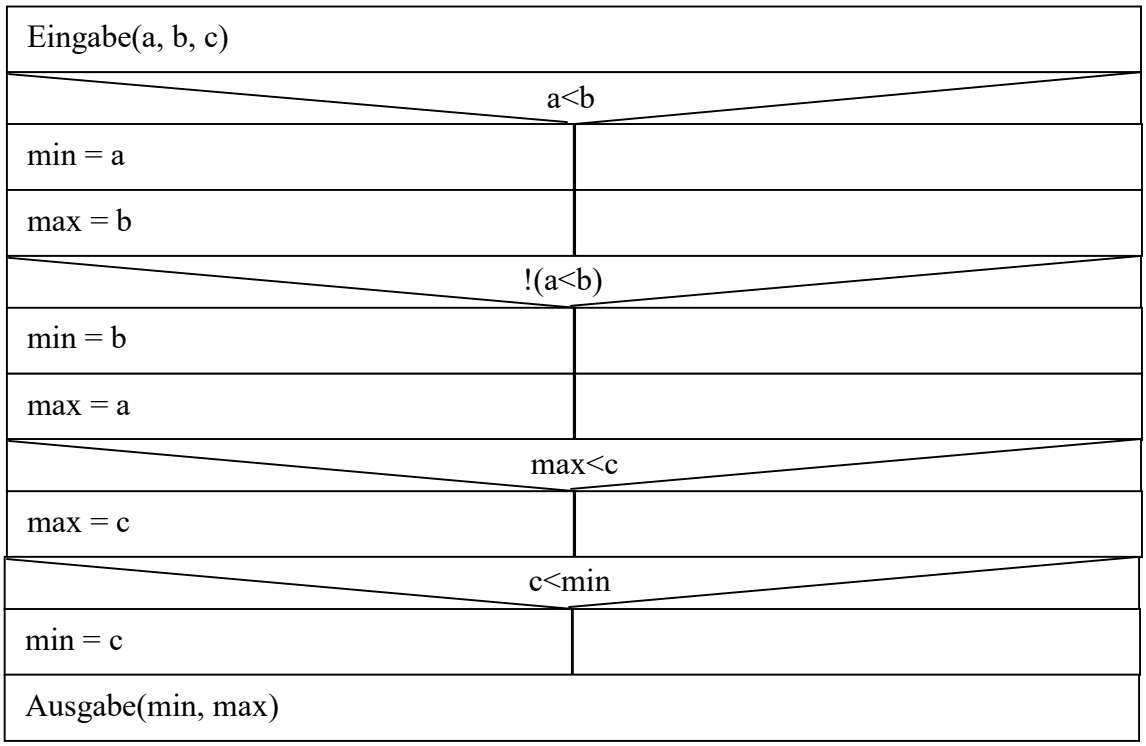
2) 5 P

$$2^{64} = 2^{60+4} = 2^{60} \cdot 2^4 = 16 \cdot 2^{60} = 16 \cdot 2^{10 \cdot 6} = 16 \cdot (2^{10})^6 \approx 16 \cdot (10^3)^6 = 16 \cdot 10^{18}$$

3) 10P

Der Algorithmus ist nicht korrekt:
z1 = 30 z2 = 20 z3 = 10 ==> min = 0

4) 12P



5a)

Anzahl der ausgegebenen Schüler	Wert der Variablen zähler
1	3
2	4
...	...
51	53

b) zähler = 53, Anzahl der Schüler = 51

c) Programm nicht korrekt, da laut Tabelle 51 Schüler verarbeitet werden.

KLAUSUR 1 SAE E1FI Nachtermin Zeit: 90 Minuten

Name, Vorname:

Hilfsmitttel:
keine

Hinweise (unbedingt beachten):

→ Alle Aufgaben müssen bearbeitet werden.
→ Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
→ Aufgabenblätter bitte auch abgeben.
→ Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
→ Die rote Farbe darf nicht benutzt werden.
→ Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
→ Selbsterklärende Variablennamen benutzen.
→ Programme müssen benutzerfreundlich sein.
→ EVA-Prinzip muss benutzt werden.
→ Einrücken der entsprechenden Programmteile.
→ Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
→ In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
→ Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 10 P

Simulieren Sie die folgende if-else Verzweigung durch eine oder mehrere einseitige Verzweigungen, wobei außer dem nicht Operator ! keine weiteren logischen Operatoren verwendet werden dürfen.

```
if (B1 && B2){
  A1
}
else{
  A2
}
```

2) 20 P

a) Schreiben Sie ein C-Programm (nur den Verarbeitungsteil), das von 2 Mengen A und B (die jeweils aus ganzen Zahhlen bestehen) mit
A = {a1, a2} mit a1 ≠ a2
B = {b1, b2} mit b1 ≠ b2
den Durchschnitt bestimmt.
Der Durchschnitt sind genau die Zahlen, die sowohl in A als auch in B vorkommen.

b) Erstellen Sie dazu ein Testprotokoll mit 5 verschiedenen Tests.

3) 20 P

Eine natürliche Zahl n (n>=2) heißt Primzahl, wenn sie nur durch 1 und sich selbst teilbar ist.
Beispiele für Primzahlen: 2, 3, 5, 7, 11, 13, ...

a) Erstellen Sie ein Flussdiagramm, das folgendes macht:
Es muß von einer eingegebenen ganzen Zahl z >= 2 feststellt werden, ob diese eine Primzahl ist. Das Ergebnis muß dann auf dem Bildschirm ausgegeben werden.

b) Erstellen Sie dazu ein Testprotokoll mit 5 verschiedenen Tests.

KLAUSUR 2 SAE E1FI 8.1.2014 Zeit: 45 Minuten

Name, Vorname:

Hilfsmitttel:
keine

Hinweise (unbedingt beachten):

→ Alle Aufgaben müssen bearbeitet werden.
→ Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
→ Aufgabenblätter bitte auch abgeben.
→ Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
→ Die rote Farbe darf nicht benutzt werden.
→ Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
→ Selbsterklärende Variablennamen benutzen.
→ Programme müssen benutzerfreundlich sein.
→ EVA-Prinzip muss benutzt werden.
→ Einrücken der entsprechenden Programmteile.
→ Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
→ In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
→ Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

Bemerkungen zum Quellcode:
Die Eingabe über Tastatur kann mit dem Pseudocode "eingabe" gemacht werden:

Beispiel:
Die Eingabe einer ganzen Zahl kann man mit dem folgenden Pseudocode schreiben (wobei z eine integer-Variable bedeutet):
eingabe(z);

1) 3P
Was ist eine Endlosschleife?

2) 8P
Zeichnen Sie das Flussdiagramm der folgenden Anweisung (Pseudocode):

```
while(B){
  A;
}
```

3) 10P
a) Was ist allgemein der Hauptunterschied (bzgl. der Anzahl der Durchgänge) zwischen einer while-Anweisung und einer do-while Anweisung ?
b) Geben Sie dazu jeweils (mitAnzahl der Durchgänge) ein **konkretes** Beispiel (Programmteil) an, wobei der Schleifenkörper und die Bedingung jeweils gleich sein muß.

4) 10P
a) Ein Anwender soll eine ganze Zahl zwischen 1 und 5 (je einschließlich) über Tastatur eingeben. Schreiben Sie dazu einen Programmausschnitt in Java, in dem der Anwender gezwungen wird. so lange eine Zahl einzugeben, bis diese Bedingung erfüllt ist. Erst dann soll im Programm die nächste Anweisung erreicht werden.
Die Eingabe einer ganzen Zahl kann man mit dem folgenden Pseudocode schreiben (wobei z eine integer-Variable bedeutet):
eingabe(z);

b) Geben Sie den Programmverlauf in einem Schreibtischtest (Belegung der Variablen) an für: Anwender gibt Zahl 3 ein, Anwender gibt Zahl 123 ein.

5) 6P

In den folgenden Programmauschnitten gibt der Anwender für i eine ganze Zahl ein.
a) Für welche Werte von i bringen die Programmausschnitte die gleiche Anzahl von Meldungen auf den Bildschirm ?
b) Für welche Werte von i bringen die Programmausschnitte nicht die gleiche Anzahl von Meldungen auf den Bildschirm ?

```
eingabe(i);
while (i<=3){
  i = i+1;
  Sop("Hallo Welt");
}
```

```
eingabe(i);
do{
  i = i+1;
  Sop("Hallo Welt");
} while(i<=3);
```

Bem:
Sop ist die Abkürzung für `System.out.println`

6) 14P

a) Geben Sie einen Programmausschnitt in Java an, der mit Hilfe einer Schleife die Summe
13 + 15 + 17 + ...+ 103 + 105
berechnet.

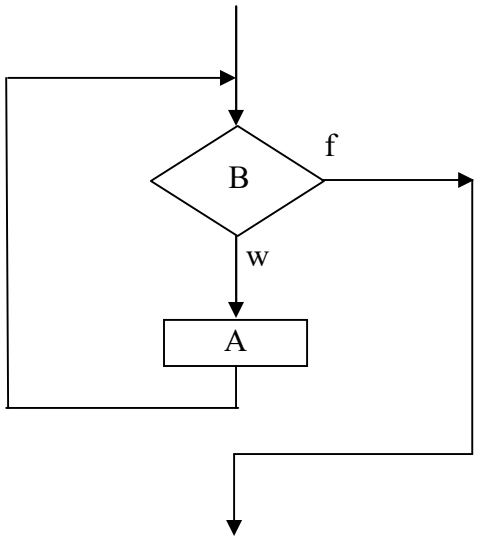
b) Begründen Sie mit Hilfe eines Schreibtischtest, daß das Programm korrekt ist.
Z.B. indem man das Programm (Belegung der Variablen) so abändert, daß es die Summe 13 + 15 berechnet und dann das Programm so lange gedanklich durchgeht (Belegeung der Variablen bei jedem Durchgang angeben und im Quellcode über den Variablen notieren), bis es beendet ist.

Lösungen:

1) 3P

Eine Endlosschleife ist eine Schleife, die nie verlassen wird.

2) 8P



3) 4+6P

while-Anweisung: 0 - 00

do-while Anweisung: 1 - 00

```
while (1<1){
  x=5;
}
```

0 Durchgänge

```
do{
  x=5;
}while(1<1)
```

1 Durchgang

4) 10P

a) 6P

```
...
do {
  System.out.println("ganze Zahl zwischen 1 und 5 eingeben");
  eingabe(i);
}while(!(i>=1 && i<=5));
```

b) 4P

Eingabe: i = 3

(!(3>=1 && 3<=5)) wird falsch und damit wird Schleife verlassen

Eingabe: i = 123

(!(123>=1 && 123<=5)) wird wahr und damit wird Schleife wiederholt.

5) 6P

i< = 3: gleiche Anzahl von Meldungen

i > 3 : verschiedene Anzahl von Meldungen

6) 10+4P

a) 10P

```
...
int summe=0;
int i=13;
while (i<=105){
  summe = summe + i;
  i = i+2;
}
...
```

b) 4P

```
...
int summe=0;
int i=13;                 // i: 13
while (i<=15){            // 13<=15 ; 15<=15 ; 17<=15
  summe = summe + i;      // summe: 0+13; 13+15
  i = i+2;                //i: 15, 17
}
...
```

KLAUSUR 2 SAE E1FI Nachtermin Zeit: 45 Minuten

Name, Vorname:

Hilfsmitttel:
keine

Hinweise (unbedingt beachten):

→ Alle Aufgaben müssen bearbeitet werden.
→ Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
→ Aufgabenblätter bitte auch abgeben.
→ Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
→ Die rote Farbe darf nicht benutzt werden.
→ Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
→ Selbsterklärende Variablennamen benutzen.
→ Programme müssen benutzerfreundlich sein.
→ EVA-Prinzip muss benutzt werden.
→ Einrücken der entsprechenden Programmteile.
→ Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
→ In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
→ Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 50 P

Definition:
Die Quersumme einer ganzen Zahl ist die Summe seiner Ziffern.
Beispiele:
Die Quersumme von 367 ist: 3 + 6 + 7 = 16
Die Quersumme von 1996 ist: 1 + 9 + 9 + 6 = 25
Die Quersumme von -13 ist: 4 (Quersumme ist immer **≥0**)

a) Erstellen Sie ein **Struktogramm**, das von einer ganzen Zahl (Datentyp integer) die Quersumme berechnet.
Beachten Sie:
Die Quersumme ist immer **≥0**

b) Machen Sie dazu dazu verschiedene Tests (dokumentierte Protokolle mit Testdaten). Geben Sie jeweils an, ob die Tests erfolgreich waren oder nicht.

KLAUSUR 3 SAE E1FS 7.5.2014 Zeit: 45 Minuten

Name, Vorname:

Hilfsmitttel:
keine

Hinweise (unbedingt beachten):

→ Alle Aufgaben müssen bearbeitet werden.
→ Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
→ Aufgabenblätter bitte auch abgeben.
→ Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
→ Die rote Farbe darf nicht benutzt werden.
→ Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
→ Selbsterklärende Variablennamen benutzen.
→ Programme müssen benutzerfreundlich sein.
→ EVA-Prinzip muss benutzt werden.
→ Einrücken der entsprechenden Programmteile.
→ Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
→ In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
→ Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

1) 14P
a) Was ist eine Klasse ?
b) Was ist ein Objekt?
c) Was enthält eine Klasse?
d) Was ist der Sinn einer Methode?
e) Was ist ein Attribut?
f) Was bedeutet private?
g) Was bedeutet public?

2) 10P
gegeben sei folgender Programmausschnitt (Hund ist eine Klasse):

```
...
Hund myh1;
myh1 = new Hund();
Hund myh2;
myh2 = new Hund();
...
```

Was veranlassen diese obigen 4 Zeilen Programmcode im Arbeitsspeicher?

Bezeichnung	Adresse	Inhalt
myh1		
myh2		

Vervollständigen Sie den folgenden Satz:
myh1 bezeichnet kein Objekt, sondern ....

Auf der Rückseite geht es weiter !!!!!!!

3) (Alle Teilaufgaben müssen in **einem** Programm realisiert werden) 28P
Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.

a) Erstellen Sie die Klasse Konto, (mit genau 2 Attributen, genau 2 set-Methoden, genau 2 get-Methoden und genau 2 Konstruktoren), die ein Sparkonto mit einem Kontostand und einem Zinssatz repräsentieren soll.

b) Erzeugen Sie ein Konto mit dem Kontostand von 5 (Euro) und einem Zinssatz von 3%

c) Verändern Sie (nur mit Hilfe der get-bzw. set-Methoden) den Kontostand um den Wert +1000 (Euro) und den Zinssatz um -2 Prozentpunkte.
Bemerkung:
Der neue Kontostand darf nicht selbst mit einem Taschenrechner berechnet werden, also 5 + 1000 nicht selbst mit dem Taschenrechner berechnen, sondern dies muß programmtechnisch realisiert werden.
Das gleiche gilt für den Zinssatz..

d) Ermitteln Sie mit Hilfe der entsprechenden Methoden den neuen Kontostand und den neuen Zinssatz und geben diese auf dem Bildschirm aus.

e) Erzeugen Sie ein anderes, neues Konto mit dem Kontostand von 10 (Euro) und einem Zinssatz von 5%.

Lösung:

1) 14 Punkte

a) Ein Bauplan, nach dem ein Objekt erstellt wird

b) Ein nach einem Bauplan im Arbeitspeicher angelegter, reservierter Speicher.

c) Attribute und Methoden.

d) Attribute zu lesen und zu beschreiben.

e) Daten in einer Klasse.

f) Auf private Member darf nur innerhalb einer Klasse zugegriffen werden.

g) Auf public Member darf innerhalb und außerhalb einer Klasse zugegriffen werden.

2) 10 Punkte

Bezeichnung	Adresse	Inhalt
myh1	0800	0900
	...	
	0900	Attribute
		des
		Objekts
myh2	0200	0300
	...	
	0300	Attribute
		des
		Objekts

Vervollständigen Sie den folgenden Satz:
myh1 bezeichnet kein Objekt, sondern eine Referenz auf ein Objekt.

3)

```
public class MainKonto1 {
    public static void main(String[] args) {
         Konto k1, k2;
        // 2 P
        k1= new Konto(5,3);
        // 6 P
        k1.setKontostand(k1.getKontostand()+1000);
        k1.setZinssatz(k1.getZinssatz()-2);
        // 4 P
        System.out.println("neuer Kontostand=
                                          "+k1.getKontostand());
        System.out.println("neuer Zinssatz=
                                          "+k1.getZinssatz());
        // 2 P
        k2= new Konto(10,5);
    }
}

class Konto{
    // 2 Punkte
    private double kontostand;
    private double zinssatz;

    // 2 Punkte
    public Konto(){
        kontostand = 0;
        zinssatz = 0;
    }

    // 2 Punkte
    public Konto(double pKontostand, double pZinssatz){
        kontostand = pKontostand;
        zinssatz = pZinssatz;
    }

    // 2 Punkte
    public double getKontostand() {
        return kontostand;
    }

    // 2 Punkte
    public double getZinssatz() {
        return zinssatz;
    }

    // 2 Punkte
    public void setKontostand(double kontostand) {
        this.kontostand = kontostand;
    }

    // 2 Punkte
    public void setZinssatz(double zinssatz) {
        this.zinssatz = zinssatz;
    }
}
```

KLAUSUR 4 SAE E1FS 2.7.2014 Zeit: 45 Minuten

Name, Vorname:

Hilfsmitttel:
keine

Hinweise (unbedingt beachten):

→ Alle Aufgaben müssen bearbeitet werden.
→ Der Name und Vorname muß in DRUCKSCHRIFT auf jedes Aufgabenblatt und auf jedes Lösungsblatt geschrieben werden.
→ Aufgabenblätter bitte auch abgeben.
→ Die Lösungsblätter müssen in folgender Form durchnummeriert werden. Beispiel: 1/4 2/4 3/4 4/4
→ Die rote Farbe darf nicht benutzt werden.
→ Lassen Sie bitte auf der linken Seite einen mindestens 3cm breiten Rand.
→ Selbsterklärende Variablennamen benutzen.
→ Programme müssen benutzerfreundlich sein.
→ EVA-Prinzip muss benutzt werden.
→ Einrücken der entsprechenden Programmteile.
→ Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.
→ In allen Klassen dürfen in keiner Methode Ein- oder Ausgaben (auf dem Bildschirm) vorkommen, außer es handelt sich um reine Ausgabe- oder Eingabe Methoden.
→ Bei Nichtbeachtung dieser Hinweise gibt es einen Punkteabzug !!!!

AUFGABEN

(Alle Teilaufgaben müssen in **einem** Programm realisiert werden)
Sichtbarkeits- und Zugriffsmodifizierer wie public, usw. dürfen nicht weggelassen werden.

I)
Ein Spielkasino will ihre Glücksspiele digitalisieren. Unter anderem soll dazu ein objektorientiertes Programm erstellt werden.

1) 22P
Erstellen Sie dazu die Klasse Wuerfel, (mit **genau** den 2 Attributen "laenge" und "zustand", den entsprechenden get-und set-Methoden und **genau** den Methoden "wuerfeln(...)", "getOberflaeche(...)", "getVolumen(...)" und **genau** einen Konstruktor mit einem Parameter).
Das Attribut "zustand" repräentiert die Augenzahl der oben liegenden Seite des Würfels.
Wenn ein Würfel erzeugt wird, hat er somit eine oben liegenden Seite, die mit einem Würfelbecher ermittelt wurde.
Um den Würfelvorgang zu simulieren bekommen Sie von einem Mathematiker die Methode "zufall()" spendiert, die eine ganze Zahl zwischen 1 und 6 zurückgibt.

2) 2P
Warum ist es sinnlos die Methode setZustand(...) zu implementieren (programmieren) ?

Die folgenden Aufgaben bitte in main() realisieren.

3) 2P
Erzeugen Sie den Würfel "lucky" mit der Seitenlänge 1.

4) 2P
Erzeugen Sie den Würfel "hope" mit der Seitenlänge 2.

5) 12P

Erstellen Sie das integer-Feld "zahlenArray" der Länge 5.
Würfeln Sie 2 Mal hintereinander mit dem Würfel "lucky" und dann 3 Mal hintereinander mit dem Würfel "hope" und speichern Sie die gewürfelten Werte hintereinander in dem Array "zahlenArray" ab.

6) 5P

Um zu überprüfen, wie gut der Würfel den Zufall simuliert, wird der Mittelwert der erwürfelten Werte ermittelt.
Berechnen Sie dazu mit Hilfe einer Schleife den Mittelwert der abgelegten Werte im Array "zahlenArray" (Mittelwert abspeichern in einer entsprechenden Variablen).

7) 4P

Ermitteln Sie mit der entsprechenden Methode das Volumen und die Oberfläche der zwei Würfel "lucky" und "hope" (Volumen und Oberfläche abspeichern in entsprechenden Variablen).

8) 4P

Ermitteln Sie mit der entsprechenden Methode die zuletzt gewürfelte Augenzahl der Würfels "lucky" und "hope" (Augenzahlen abspeichern in entsprechenden Variablen).

Lösungen

1)

```
class Wuerfel{
    private double laenge;                      // 1P
    private int zustand;                        // 1P

    public void setLaenge(double laenge){       // 2P
        this.laenge=laenge;
    }

    public double getLaenge(){                  // 2P
        return laenge;
    }

    public int getZustand(){                    // 2P
        return zustand;
    }

    public Wuerfel(double laenge){              // 4P
        this.laenge=laenge;
        zustand=wuerfeln();
    }
    int wuerfeln(){                            // 4P
        int erg;
        erg=(int)(100*Math.random());
        zustand=erg;
        return erg;
    }

    public double getOberflaeche(){            // 3P
        return(6*laenge*laenge);
    }

    public double getVolumen(){                // 3P
        return(laenge*laenge*laenge);
    }
}
```

2) 2P

Da der Zustand nur durch die Methode wuerfeln() verändert wird, hat es keinen Sinn ihn mit einer weiteren Methode zu manipulieren.

```
    public static void main(String[] args) {
        int  zahl1, zahl2;
        int summe=0;
        double mittelwert;
        double flaeche1, flaeche2;
        double volumen1, volumen2;

        Wuerfel lucky=new Wuerfel(1);          // 2P
        Wuerfel hope=new Wuerfel(2);           // 2P
        int zahlenArray[] = new int[5];        // 2P

        zahlenArray[0]=lucky.wuerfeln();       // 2P
        zahlenArray[1]=lucky.wuerfeln();       // 2P
        zahlenArray[2]=hope.wuerfeln();        // 2P
        zahlenArray[3]=hope.wuerfeln();        // 2P
        zahlenArray[4]=hope.wuerfeln();        // 2P

                                               // 5P
        for(int i=0;i<5;i++){
            summe=summe+zahlenArray[i];
        }
        mittelwert=summe/5.0;

        flaeche1=lucky.getOberflaeche();       // 1P
        volumen1=lucky.getVolumen();           // 1P
        flaeche2=hope.getOberflaeche();        // 1P
        volumen2=hope.getVolumen();            // 1P

        zahl1=lucky.getZustand();              // 2P
        zahl2=hope.getZustand();               // 2P

    }
}
```