

Name, Vorname:

Hilfsmittel:
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) 2P
Welchen Wert hat die Variable z nach Ausführung der folgenden Java-Anweisung ?
Begründen Sie !

`z = 1/4 * 4;`

2) 2P
Wie viele Bit genau braucht man, um genau 32 verschiedene Zustände darzustellen ?

3) 6P
Stellen Sie den folgenden Pseudocode durch ein Struktogramm dar, wobei dort nur einseitige Verzweigungen vorkommen dürfen.

```
if (B) {  
    A1;  
}  
else {  
    A2;  
}
```

4) 5P
Annahme: Die Variable x soll vor der Ausführung folgender Anweisung nur die zwei verschiedenen Zahlenwerte A bzw. B annehmen können:

`x = A + B - x;`

- a) Welche Werte kann dann x nach Ausführung dieser Anweisung annehmen ?
- b) Wie kann man dann diese Anweisung mit Hilfe einer Verzweigung in Java programmieren ?

5) 5P
a) Welchen Wert haben x und y nach Ausführung des folgenden Programms:
Geben Sie den Wert von x und y nach jeder Anweisung an.

```
...  
x = 7;  
y = 3;  
x = x - y;  
y = x + y;  
x = y - x;
```

...

b) Was macht dieses Programm also im Endeffekt mit dem Inhalt der Variablen x und y (verbale Beschreibung) ?

6) 10P

Erstellen Sie ein Struktogramm, das das Minimum dreier Zahlen (Variablen z1, z2, z3) berechnet, in der Variablen min speichert und auf dem Bildschirm ausgibt. EVA-Prinzip beachten!

7) 10P

Der folgende syntaktisch korrekte Teil eines Java-Programms soll die größte Zahl (dreier Zahlen z1, z2, z3) berechnen und in der Variablen max 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 max) 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 max) an, bei dem der Algorithmus falsch wird.

```
... main(...) {  
    ...  
    max = z2;  
    if (z1 < z2) {  
        max = z1;  
    }  
    if (max < z3) {  
        max = z3;  
    }  
    ...  
}
```

8) 10P

Erstellen Sie ein Struktogramm zu dem Programm, das zu der in einer Prüfung erreichten Punktezahl p, (die der Lehrer über Tastatur eingibt), folgende Meldung ausgibt:

p zwischen 0 und 36 Punkte (je einschließlich):	"Prüfung nicht bestanden"
p größer 36 Punkte und kleiner (oder gleich) 100 Punkte:	"Prüfung bestanden"
p nicht im Bereich zwischen 0 und 100:	"unzulässige Punktezahl"

Bemerkung: Die Punktezahl kann auch nicht ganzzahlig sein!

Lösungen:

1) 2P

$z = 0$, da 1 und 4 integer-Zahlen sind, deren Division den Wert 0 ergibt.

2) 2P

5 Bit

3) 6P

W	B
A1	max = a
W	nicht B
A2	

4) 5P

a) 2P

A bzw. B

b) 3P

if(x==A)

 x = B;

else

 x = A;

5) 5P

a) 3P

x = 7;

y = 3;

x = x - y; // x = 7-3 = 4

y = x + y; // y = 4+3 = 7

x = y - x; // x = 7-4 = 3

b) 2P

Das Programm vertauscht den Wert von x und y

6) 10P

Eingabe(z1, z2, z3)	
W	$z1 < z2$
min = z1	min = z2
W	$z3 < \text{min}$
min = z3	
Ausgabe(min)	

7)

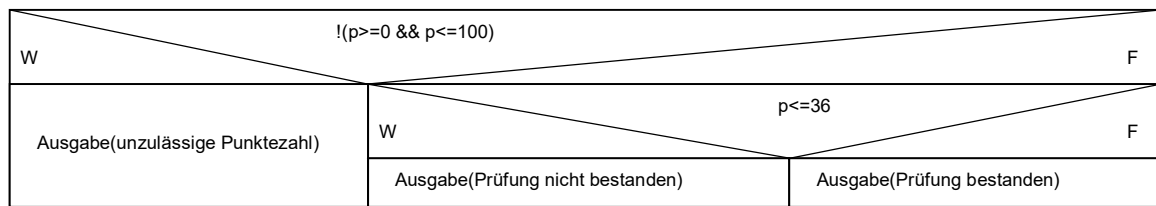
10P

Der Algorithmus ist nicht korrekt:

$z_1 = 30 \quad z_2 = 20 \quad z_3 = 10 \implies \max = 20$

8)

10P



Name, Vorname:

Hilfsmittel:
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 Punktabzug !!!!

AUFGABEN

- 1) 5P
a) Wie viele Zustände genau genau kann man mit 8 Byte angeben? (als Hochzahl schreiben)
b) Näherungsweise Berechnung!

- 2) 10P
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  
}
```

- 3) 15P
Erstellen Sie ein Struktogramm, das drei Zahlen (Variablen z1, z2, z3) der Größe nach in aufsteigender Reihenfolge sortiert, in die Variablen klein, mittel, gross speichert und auf dem Bildschirm ausgibt. EVA-Prinzip beachten!

4)

20 P

Eine natürliche Zahl n ($n \geq 2$) 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 $z \geq 2$ festgestellt werden, ob diese eine Primzahl ist. Das Ergebnis muß dann auf dem Bildschirm ausgegeben werden.

Tipp: Der Operator % berechnet den Rest bei einer Division. Damit kann man dann feststellen, ob eine Zahl eine andere teilt.

Beispiele:

$38 \% 3 = 2$, weil $38 : 3 = 12$ Rest 2, also teilt 3 nicht die Zahl 38

$38 \% 11 = 5$, weil $38 : 11 = 3$ Rest 5, also teilt 11 nicht die Zahl 38

$38 \% 2 = 0$, weil $38 : 2 = 19$ Rest 0, also teilt 2 die Zahl 38

Name, Vorname:

Hilfsmittel:
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 Punktabzug !!!!

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 Programmausschnitten gibt der Anwender für i einen Wert 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

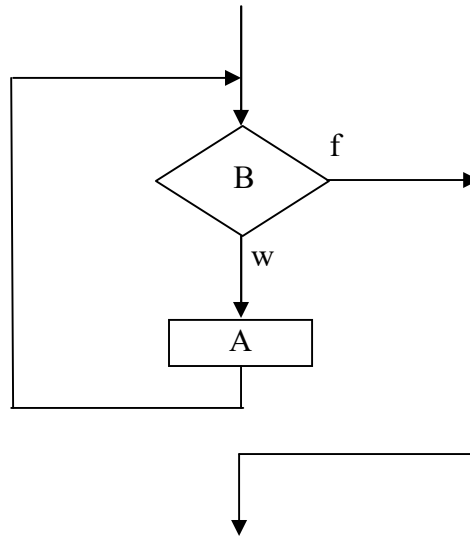
- a) Geben Sie einen Programmausschnitt in Java an, der mit Hilfe einer Schleife die Summe $13 + 15 + 17 + \dots + 103 + 105$ berechnet.

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 (Belegung 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;
while (i<=15){
    summe = summe + i;
    i = i+2;
}
...
```

// i: 13
// 13<=15 ; 15<=15 ; 17<=15
// summe: 0+13; 13+15
//i: 15, 17

Name, Vorname:

Hilfsmittel:
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 Punktabzug !!!!

AUFGABEN

1) 25 P

Eine natürliche Zahl $n > 0$ heißt vollkommene Zahl (perfekte Zahl), wenn sie gleich der Summe aller ihrer positiven Teiler (außer sich selbst) ist.

Beispiel:

Die Summe der Teiler von 6 (außer sich selbst): $1 + 2 + 3 = 6$, also ist 6 vollkommen.

Schreiben Sie ein Programm, das bestimmt, ob eine natürliche Zahl $n > 0$ vollkommen ist.

2) 25 P

Zu Beginn eines Jahres wird ein Anfangskapital K_0 zum Zinssatz p angelegt. Welchen Wert hat es nach

1, 2, 3, ... n Jahren, wenn die Zinsen auf dem Sparbuch bleiben ?

Mathematische Anleitung anhand eines Beispiels:

Voraussetzungen:

Anfangskapital $K_0 = 1000$

$p = 10\%$

n	Kapital K	Zinsen	neues Kapital
0	1000	$1000 * 10/100 = 100$	$1000 + 100 = 1100$
1	1100	$1100 * 10/100 = 110$	$1100 + 110 = 1210$
2	1210	$1210 * 10/100 = 121$	$1210 + 121 = 1331$
3

Schreiben Sie ein Programm, das das Endkapital K_{end} berechnet, das ein Anfangskapital K_0 bewirkt, wenn es zu einem Zinssatz von $p\%$ insgesamt n Jahre auf dem Konto angelegt wird (und die Zinsen auf dem Sparbuch bleiben) ?

Name, Vorname:

Hilfsmittel:
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)

35P

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 -367 ist: $3 + 6 + 7 = 16$

Die Quersumme von -1996 ist: $1 + 9 + 9 + 6 = 25$

a)

Schreiben Sie ein Programm, das von einer ganzen, über Tastatur eingegebenen Zahl n die Quersumme berechnet und auf dem Bildschirm ausgibt.

Bem: Die Quersumme einer negativen Zahl z ist gleich der Quersumme der positiven Zahl z.

b)

Machen Sie einen Test (nachvollziehbares Protokoll Ihrer Wahl) für die Zahl 234

2)

15P

Der Programmausschnitt (siehe unten) gibt etwas auf dem Bildschirm aus.

Die Tabelle zeigt einen 7 x 10 Bildschirm (7 Zeilen und 10 Spalten) an.

Tragen Sie die Ausgabe des Programms in die Tabelle (Bildschirm) ein.

Bem:

Schreiben Sie bei jedem Durchgang die neuen Werte der Variablen über die jeweilige Variable im Programm.


```
public static void main(...) {
    int i,j,k;
    i=0;
    while(i<3){
        for(j=0;j<3-i;j++){
            System.out.print(" ");
        }

        for(k=0;k<2*i+1;k++){
            System.out.print("*");
        }
        System.out.println();

        i=i+1;
    }
}
```

1)

```
public class MainQuersumme1 {
    public static void main(String[] args) {
        int zahl;
        int sum;
        int ziffer;
        int erg;
        Scanner scanner = new Scanner(System.in);

        sum = 0;
        System.out.println("Zahl eingeben");
        zahl = scanner.nextInt();

        if (zahl < 0)
            zahl = -zahl;

        erg = zahl;
        do{
            ziffer = erg % 10;
            erg = erg / 10;
            sum = sum + ziffer;
        }while (erg != 0);

        System.out.println("Quersumme = "+ sum);
    }
}
```

2)

[illegible]

Name, Vorname:

Hilfsmittel:
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 Punktabzug !!!!

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);

AUFGABEN

1) 25 P

Eine natürliche Zahl $n > 0$ heißt vollkommene Zahl (perfekte Zahl), wenn sie gleich der Summe aller ihrer positiven Teiler (außer sich selbst) ist.

Beispiel:

Die Summe der Teiler von 6 (außer sich selbst): $1 + 2 + 3 = 6$, also ist 6 vollkommen.

($28 = 1+2+4+7+14$ ist auch eine vollkommene Zahl).

Schreiben Sie ein Programm, das bestimmt, ob eine natürliche Zahl $n > 0$ vollkommen ist.

2) 25 P

Zu Beginn eines Jahres wird ein Anfangskapital K_0 zum Zinssatz p angelegt. Welchen Wert hat es nach

1, 2, 3, ... n Jahren, wenn die Zinsen auf dem Sparbuch bleiben ?

Mathematische Anleitung anhand eines Beispiels:

Voraussetzungen:

Anfangskapital $K_0 = 1000$

$p=10\%$

n	Kapital K	Zinsen	neues Kapital
0	1000	$1000 * 10/100 = 100$	$1000 + 100 = 1100$
1	1100	$1100 * 10/100 = 110$	$1100 + 110 = 1210$
2	1210	$1210 * 10/100 = 121$	$1210 + 121 = 1331$
3

Schreiben Sie ein Programm, das das Endkapital K_{end} berechnet, das ein Anfangskapital K_0 bewirkt, wenn es zu einem Zinssatz von $p\%$ insgesamt n Jahre auf dem Konto angelegt wird (und die Zinsen auf dem Sparbuch bleiben) ?