

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 Variablenamen 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) a) Was ist ein Algorithmus ? (6 P)
b) Welche Eigenschaften hat er?

2) a) Was ist ein Literal ? (3 P)
b) Geben Sie ein Beispiel.
c) Muss ein Literal deklariert werden?

3) a) Warum gibt es bei der Ausführung folgender, syntaktisch korrekter Anweisung Probleme ?
b) Wo werden Probleme verursacht (beim Compilieren oder während des Programmlaufs)?

```
int d; (5 P)  
d = 6 / (1/3);
```

4) Ist die Verwendung des cast-Operators in den folgenden 2 syntaktisch korrekten Anweisungen unbedingt nötig ? Begründen Sie ! (8 P)

```
double d, r;  
int i;  
r=2;  
i=1;  
d = (double)2 * 3.14 * r;  
i = (int)(4 * 2.718 * i);
```

5) Welche Wert(e) können die folgenden syntaktisch korrekten Ausdrücke jeweils annehmen ? (3 P)

```
z = 10  
z == 10
```

6) Erstellen Sie ein Struktogramm, das das Minimum und das Maximum dreier Zahlen berechnet und auf dem Bildschirm ausgibt. (12P)
Es dürfen nur einseitige Verzweigungen benutzt werden!

7)

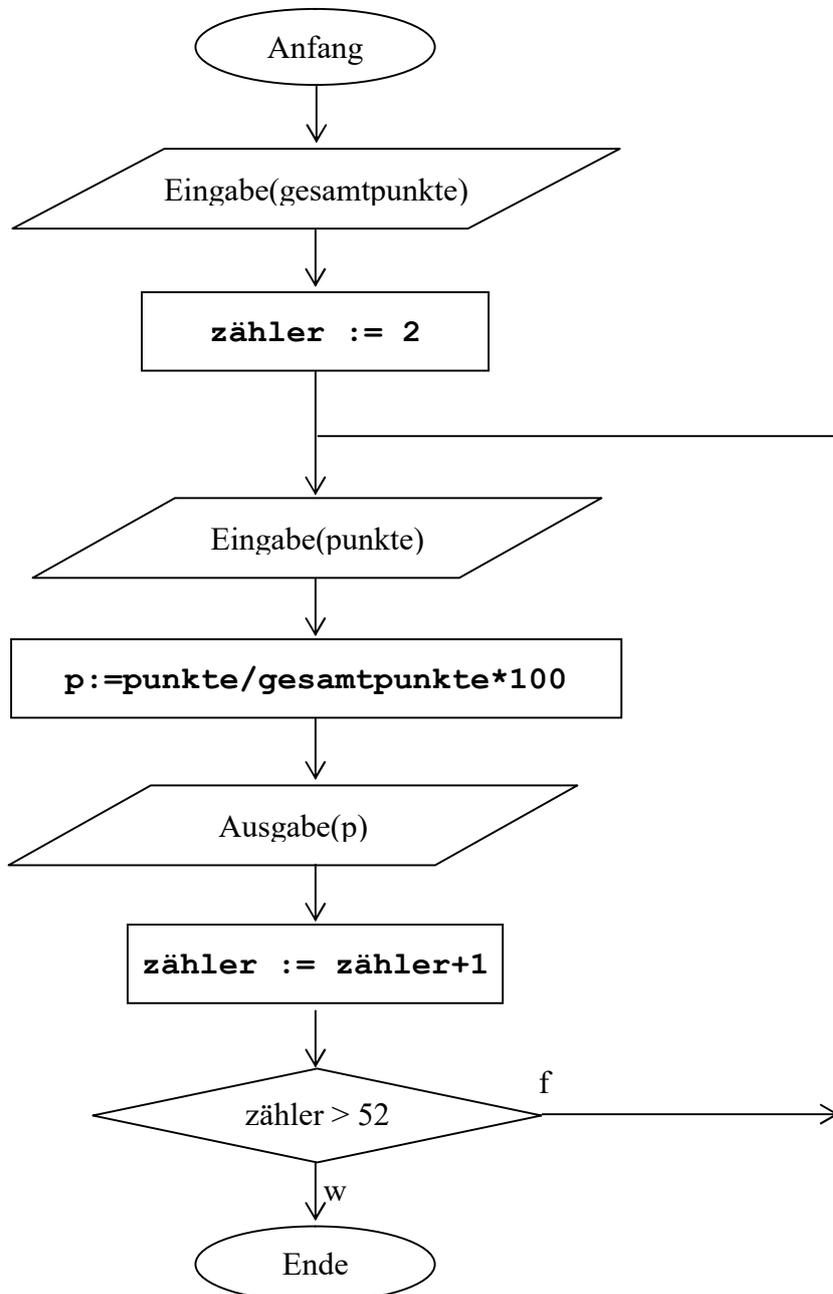
Es soll von 50 Schülern jeweils der Prozentsatz der erreichten Punkte bzgl. der Gesamtpunktzahl in einer Klassenarbeit errechnet und ausgegeben werden.

a) Ist die folgende Lösung korrekt ? Begründen Sie mit der Tabelle unten! (8 P)

Suchen Sie dazu eine geeignete Stelle im Programm aus (**markieren** diese mit einem Pfeil), an der Sie das Programm gedanklich bei jedem Schleifendurchgang anhalten und dort den Wert der Variablen zähler und die Anzahl der ausgegebenen Schüler protokollieren:

b) Falls sie nicht korrekt ist, verbessern sie die Lösung (mit neuer Tabelle) (5 P)

Anzahl der ausgegebenen Schüler	Wert der Variablen zähler



Lösungen:

1) Ein Algorithmus ist ein Verfahren zur Lösung eines gegebenen Problems. Er ist eindeutig, endlich und schrittweise.

2) Literale sind Bezeichner mit einem festen Wert wie z.B. 9

3) $d = 6 / (1/3);$

a)

$1/3$ hat den Wert 0, also wird durch Null dividiert.

Diese Anweisung wird von keinem Mikroprozessor ausgeführt.

b) Der Compiler akzeptiert diese Anweisung. Erst während der Programmausführung wird erscheint eine Fehlermeldung.

4) a)

$d = (\text{double})2 * 3.14 * r;$

2 ist ein Integer-Wert. Bei Multiplikation mit einem double-Wert (hier 3.14) wird der Integer-Wert 2 vor der Multiplikation implizit (automatisch, ohne Zutun des Programmierers) in einen double-Wert umgewandelt. Das bedeutet, daß die cast Umwandlung mit (double) nicht gemacht werden muß.

b)

$i = (\text{int})(4 * 2.718 * i)$

Der Wert von $4 * 2.718 * i$ ergibt einen double-Wert. Wenn man die explizite Umwandlung mit dem cast Operator (int) wegläßt, wird dieser double-Wert in der Integer-Wert Variablen i abgespeichert. Dadurch entsteht ein Datenverlust. Dies meldet der Compiler.

5) Der Ausdruck $z = 10$ hat immer den Wert 10.

Der Ausdruck $z == 10$ hat den Wert true oder false

6)

Eingabe(a, b, c)	
$a < b$	
min = a	
max = b	
$!(a < b)$	
min = b	
max = a	
$max < c$	
max = c	
$c < min$	
min = c	
Ausgabe(min, max)	

7a)

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

Es werden nicht 50, sondern 51 Schüler ausgegeben.

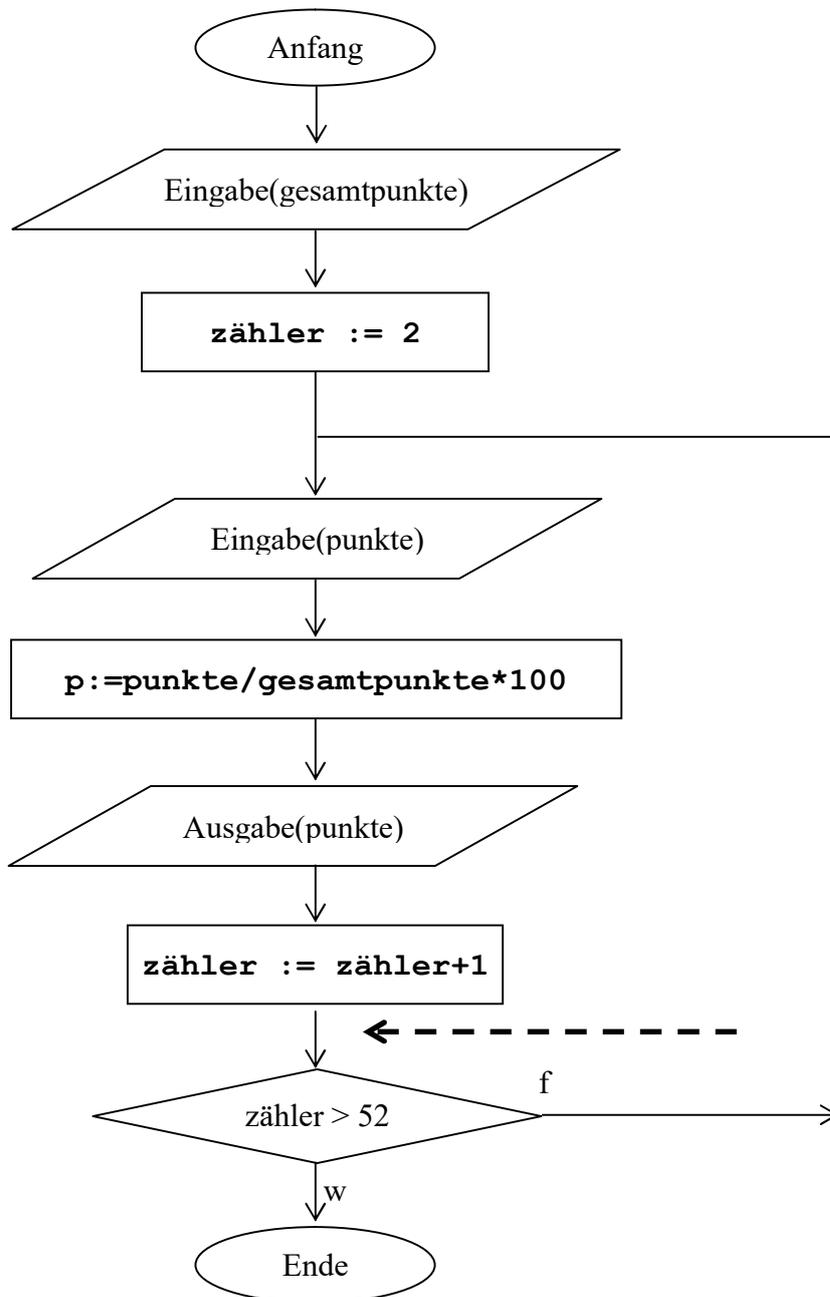
Begründung: siehe Tabelle.

b) Korrektur:

Statt: $\text{zähler} > 52$

besser: $\text{zähler} \geq 52$

Tabelle: wie obige Tabelle, aber Zeile: 50 --> 52



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 Variablenamen 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) 5P

Erzeugt der Compiler in der folgenden, syntaktisch korrekten Anweisung eine Fehlermeldung ? Begründen Sie !

```
float d;  
d= 2 * 3.14 * d;
```

2) 5P

a) Wie viele Zustände genau kann man mit 4 Byte darstellen ?
Geben Sie nur die Formel an.

3) 40P

Drei über Tastatur eingegebene Zahlen sollen der Größe nach sortiert und dann auf dem Bildschirm ausgegeben werden. (Zuerst die kleinste Zahl, dann die mittlere und zum Schluß die größte Zahl).

Erstellen Sie dazu ein Struktogramm.

Lösungen:

1)

$d = 2 * 3.14 * d;$

Der Wert von $2 * 3.14 * d$ ergibt einen double-Wert.

Dieser double-Wert wird in der float-Variablen d abgespeichert. Dadurch kann ein Datenverlust entstehen. Dies meldet der Compiler.

2) a) 2^{32}

b) $2^{32} = 2^{2+30} = 2^2 \cdot 2^{30} = 4 \cdot 2^{30} = 4 \cdot 2^{10 \cdot 3} = 4 \cdot (2^{10})^3 \approx 4 \cdot 1000^3 \approx 4 \cdot 1000000000$

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 Variablenamen 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) (10P)

Sind die folgenden Ausdrücke syntaktisch korrekt (x und i haben Datentyp int)?

Wenn ja, geben Sie die Werte der Ausdrücke an.

- a) $x = (i = 8) - 1$
- b) $(x = i) = 8 - 1$
- c) $(x == 2) \&\& 12$
- d) $1 < x < 8$
- e) $(x == x + 1) \&\& ! (i == i - 1)$

2)

a) Was kann man (semantisch) mit einer if-else-Anweisung, aber nicht mit einer switch-Anweisung ? (5P)

b) Was kann man (semantisch) mit einer switch-Anweisung, aber nicht mit einer if-else-Anweisung ? (gibt es so einen Fall?) (5P)

3) Es soll über Tastatur eine **ganze** Zahl eingegeben werden.

Ist diese Zahl eine Schulnote zwischen 1 und 4 (je einschließlich) wird die Meldung "Prüfung bestanden" ausgegeben.

Ist diese Zahl die Schulnote 5 oder 6 wird die Meldung "Prüfung nicht bestanden" ausgegeben.

Sonst wird die Meldung "Diese Zahl ist keine Note" ausgegeben.

a) Realisieren Sie dies durch ein Java-Programm, das genau eine switch- Anweisung (keine if-else-Anweisung, keine if-Anweisung) verwendet. (15P)

b) Realisieren Sie dies durch Java-Programm, das genau eine if-else-if-Kette verwendet (keine switch-Anweisung, keine if-Anweisung, keine if-else-Anweisung). (15P)

Bemerkung:

Die Eingabe einer Zahl kann mit der Pseudofunktion eingabe(...) abgekürzt werden, wie z.B: eingabe(z)

Es muss nicht der ganze Programmkopf angegeben werden, sondern nur die Deklarationen und Anweisungen der main-Methode.

Lösungen:

1) a) Wert: 7

b) syntaktisch falsch

c) syntaktisch falsch

d) syntaktisch falsch

e) Wert: false

2)

a) switch: nur für ganzzahlige oder char Datentypen verwendbar

b) nichts

3) a)

```
package elfi_14_01_09_nr2;
import java.io.*;
public class MainElFI_14_01_09_nr2 {
    public static void main(String argv[]) throws IOException{
        int note;
        String mystr;

        BufferedReader myinput = new BufferedReader(new
                                                    InputStreamReader(System.in));
        System.out.println("Bitte eine ganze Note eingeben");
        mystr = myinput.readLine();
        note=Integer.parseInt(mystr);

        switch(note){    // int oder char
            case 1:
            case 2:
            case 3:
            case 4:
                System.out.println("Prüfung bestanden");
                break;    // nicht abweisend

            case 5:
            case 6:
                System.out.println("Prüfung nicht bestanden");
                break;

            default:    // falls kein case-Fall zutrifft
                System.out.println("Dies ist keine Note");
                break;
        }
    }
}
```

b)

```
// ....
if(note==1 || note==2 || note==3 || note==4){
    System.out.println("Prüfung bestanden");
} else if(note==5 || note==6){
    System.out.println("Prüfung nicht bestanden");
} else
    System.out.println("Dies ist keine Note");
}
```

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) (10P)

Sind die folgenden Ausdrücke syntaktisch korrekt (x und i haben Datentyp int)?

Wenn ja, geben Sie die Werte der Ausdrücke an.

- a) $x == x$
- b) $x = x - 3x + 2x$
- c) $(x == 2) || (x == -2)$
- d) $1 > i >= 0$
- e) $(x == x + 1) || (2 == 1 + 1)$

2)

a) Was kann man (semantisch) mit einer if-else-if-Anweisung (if-else-if-Kette), aber nicht mit if-else-Anweisung (gibt es so einen Fall, wenn ja, bitte Beispiel angeben)? (5P)

b) Geben Sie das Schema einer if-else-if-Anweisung (if-else-if-Kette) mittels eines Struktogramms an. (5P)

3)

Ein Programm soll das zu der in einer Prüfung erreichten Punktezahl p, (die der Lehrer über Tastatur eingibt und nicht ganzzahlig sein muss), folgende Meldung ausgeben:

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"

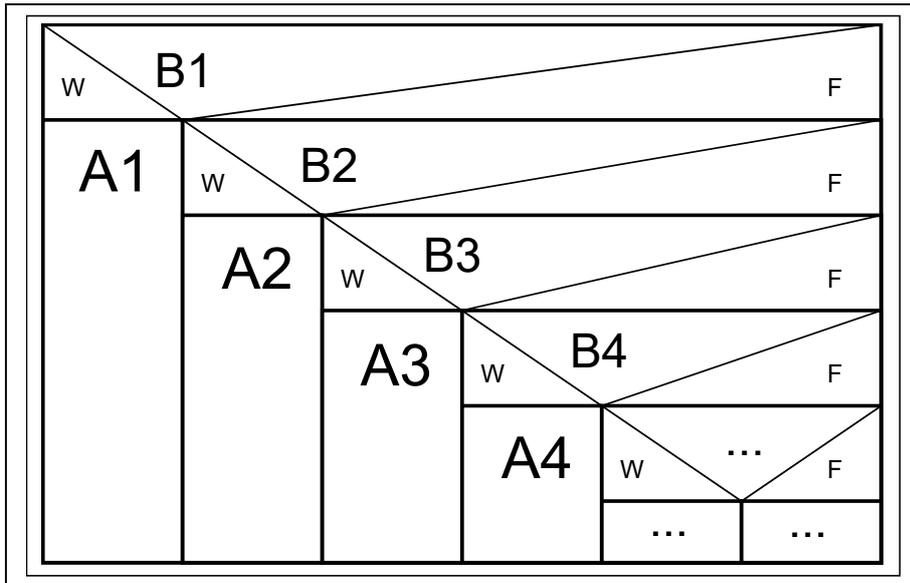
a) (15P)
Erstellen Sie dazu ein Struktogramm.

b) (15P)
Erstellen Sie dazu einen Java-Programmausschnitt (der sich semantisch vom vorigen Struktogramm unterscheidet), der mit einer einzigen if-else-if-Kette realisiert wird, wobei sich in keiner Bedingung eine logische UND bzw. ODER-Verknüpfung befinden darf.

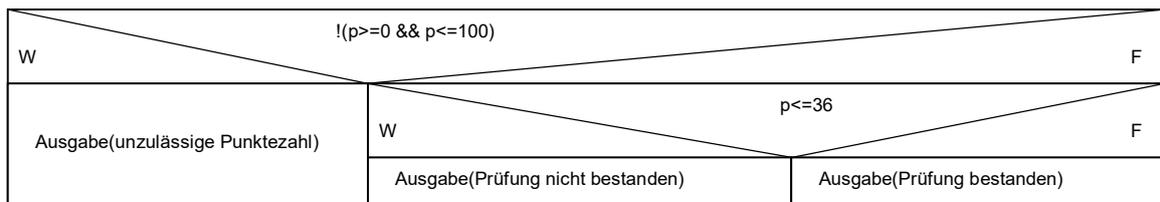
Lösungen

- 1)
 - a) Wert: 0
 - b) syntaktisch falsch
 - c) syntaktisch falsch
 - d) syntaktisch falsch
 - e) Wert: true

- 2) a) nichts
- b)



- 3) a)



b)

```
package elfi_15_01_09_nr3;
import java.io.*;
public class MainElFI_15_01_09_nr3 {
    public static void main(String argv[]) throws IOException{
        double punkte;
        String mystr;

        BufferedReader myinput = new BufferedReader(new
            InputStreamReader(System.in));
        System.out.println("Bitte erreichte Punkte eingeben");
        mystr = myinput.readLine();
        punkte =Integer.parseInt(mystr);

        if(punkte < 0)
            System.out.println("unzulässige Punktezahl");

        else if(punkte > 100)
            System.out.println("unzulässige Punktezahl");

        else if(punkte < 36)
            System.out.println("Prüfung nicht bestanden");
        else
            System.out.println("Prüfung bestanden");
    }
}
```

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) (50P)

Es soll der Ersatzwiderstand einer Parallelschaltung von 3 Widerständen berechnet werden. Wurde ein Widerstandwert kleiner oder gleich Null eingegeben, muß das Programm sofort beendet werden und eine entsprechende Meldung auf den Bildschirm ausgegeben werden. (insbesondere darf dann nicht mehr ein weiterer Widerstand eingegeben und der Ersatzwiderstand berechnet werden).

Dies muß mit dem EVA-Prinzip realisiert werden.

Außerdem muß in der Ausgabe noch angegeben werden, welcher Widerstandswert (z.B. 1. Widerstandwert) falsch eingegeben wurde.

Erstellen Sie das dazugehörige Struktogramm.

Bemerkungen:

a)
$$\frac{1}{R_G} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

b) Es widerspricht nicht dem EVA-Prinzip, wenn z.B. im Eingabeteil in bestimmten Variablen bestimmte Zustände abgespeichert werden (z.B: ob der 3. Widerstandswert kleiner oder gleich 0 ist) und diese dann im Ausgabeteil abgeprüft werden.

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) 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) 4P

gegeben sei folgender Programmausschnitt (Hund ist eine Klasse):

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

Was veranlassen diese zwei Zeilen Programmcode im Arbeitsspeicher?

Bezeichnung	Adresse	Inhalt
myh1	0800	

3) (Alle Teilaufgaben müssen in **einem** Programm realisiert werden) 32P

- a) Erstellen Sie die Klasse Punkt, (mit den entsprechenden Attributen, Methoden und zwei Konstruktoren), die einen Punkt (Pixel) in einer grafischen zweidimensionalen Oberfläche repräsentieren soll.
- b) Erzeugen Sie einen Punkt mit der x-Koordinate 5 und der y-Koordinate 10.
- c) Verändern Sie die Koordinaten um den Wert +3 in x- und den Wert -7 in y-Richtung.
- d) Ermitteln Sie mit Hilfe der entsprechenden Methoden die neuen Koordianten des Punktes und geben diese auf dem Bildschirm aus.
- e) Erzeugen Sie einen anderen, neuen Punkt mit der x-Koordinate 6 und der y-Koordinate 11.

Lösung:

1) 14 Punkte

- a) Ein Bauplan, nach dem ein Objekt erstellt wird
- b) Ein nach einem Bauplan im Arbeitsspeicher 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) 4 Punkte

Bezeichnung	Adresse	Inhalt
myh1	0800	0900
	...	
	0900	Attribute
		des
		Objekts

3)

```
public class MainKlassen6 {
    public static void main(String[] args) throws Exception{
        Punkt p1, p2;
        // 3 P
        p1= new Punkt(5,10);
        // 3 P
        p1.setX(p1.getX()+3);
        p1.setY(p1.getY()-7);
        // 3 P
        System.out.println("x= "+p1.getX());
        System.out.println("y= "+p1.getY());
        // 3 P
        p2= new Punkt(6,11);
    }
}
```

```
class Punkt{
    private int x; // 1P
    private int y; // 1P

    public Punkt(){ // 3P
        x=0;
        y=0;
    }

    public Punkt(int px, int py){ // 3P
        x=px;
        y=py;
    }

    public void setX (int px){ // 3P
        x=px;
    }

    public void setY (int py){ // 3P
        y=py;
    }

    public int getX(){// 3P
        return(x);
    }

    public int getY(){// 3P
        return(y);
    }
}
```

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) Was ist allgemein der Hauptunterschied zwischen einer While-Anweisung und einer do-while-Anweisung ?

Geben Sie ein **konkretes** Beispiel (Programmteil) an, bei dem bei gleicher Schleifenbedingung und gleichem Schleifenrumpf eine do-while-Anweisung etwas anderes macht, als eine while-Anweisung.

2) Gegeben ist der folgende syntaktisch korrekte Programmteil:

```
i=10;  
while (i<20)  
    System.out.println("Hallo Welt\n");
```

Wie oft gibt dieser Programmteil die Meldung "Hallo Welt" auf dem Bildschirm aus ?
Begründen Sie !

3) Gegeben ist der folgende syntaktisch korrekte Programmteil:

```
i=20;  
while (i<20)  
    System.out.println("Hallo Welt\n");
```

Wie oft gibt dieser Programmteil die Meldung "Hallo Welt" auf dem Bildschirm aus ?
Begründen Sie !

```

4)          // Programmausschnitt P1
           sum = 0;
           i = 1;

           --> while ... {
                sum = sum + i;
                i = i + 1;
            }

```

a) Tragen Sie insgesamt vier Mal die Werte der Variablen *i* und *sum* beim Erreichen des Programms an der mit --> gekennzeichneten Stelle in die Tabelle ein (unter der Annahme, dass die mit ... bezeichnete Schleifenbedingung wahr ist).

i					
sum					

b) Ergänzen Sie den Programmausschnitt P1 – durch die Angabe einer konkreten Schleifenbedingung – so, dass der Wert der Variablen *sum* nach Verlassen der Schleife gleich der Summe $1 + 2 + 3 + 4 + 5$ ist.

c) Verändern Sie den Programmausschnitt P1 – nur durch die Angabe einer anderen Initialisierung und einer konkreten Schleifenbedingung – so, daß die Summe $5 + 6 + 7 + 8$ berechnet wird. Tragen Sie die Werte für *sum* und *i* wieder in eine entsprechende Tabelle ein

```

5)          // Programmausschnitt P2
           sum = 0;
           i = 0;

           --> while ... {
                i = i + 1;
                sum = sum + i;
            }

```

a) Tragen Sie insgesamt vier Mal die Werte der Variablen *i* und *sum* beim Erreichen des Programms an der mit --> gekennzeichneten Stelle in die Tabelle ein (unter der Annahme, dass die mit ... bezeichnete Schleifenbedingung wahr ist).

i					
sum					

b) Ergänzen Sie den Programmausschnitt P2 – durch die Angabe einer konkreten Schleifenbedingung – so, dass der Wert der Variablen *sum* nach Verlassen der Schleife gleich der Summe $1 + 2 + 3 + 4 + 5$ ist.

c) Verändern Sie den Programmausschnitt P2 – nur durch die Angabe einer anderen Initialisierung und einer konkreten Schleifenbedingung – so, daß die Summe $5 + 6 + 7 + 8$ berechnet wird. Tragen Sie die Werte für *sum* und *i* wieder in eine entsprechende Tabelle ein

Lösungen:

1) Anzahl Durchgänge der do-while-Anweisung: ≥ 1

Anzahl Durchgänge der while-Anweisung: ≥ 0

while-Anweisung: vor dem Schleifenrumpf wird Bedingung überprüft

do-while-Anweisung: nach dem Schleifenrumpf wird Bedingung überprüft

```
i=10;
do{
  System.out.println(i);
}
while(i<10);
```

```
i=10;
while(i<10){
  System.out.println (i);
}
```

6P

2)

Unendlich oft, weil die Bedingung immer wahr ist.

3P

3)

Null Mal, weil die Bedingung falsch ist.

3P

4)

a)

i	1	2	3	4	5
sum	0	1	1+2	1+2+3	1+2+3+4

5P

b)

while (i<=5)

5P

c)

```
sum = 0;
i = 5;

--> while{ (i<=8)
  sum = sum +i;
  i = i+1;
}
```

Tabelle:

i	5	6	7	8	9
sum	0	5	5+6	5+6+7	5+6+7+8

9P

5)

a)

i	0	1	2	3	4
sum	0	1	1+2	1+2+3	1+2+3+4

5P

b)

while (i<=5)

5P

c)

```
sum = 0;  
i = 4;
```

```
--> while{ (i<8)  
    i = i+1;  
    sum = sum +i;  
}
```

Tabelle:

i	4	5	6	7	8
sum	0	5	5+6	5+6+7	5+6+7+8

9P

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) 20P

Erstellen Sie ein Struktogramm eines Programm, das die Fakultät einer ganzzahligen Zahl (≥ 1) berechnet.

$$n! = n * (n-1) * (n-2) * \dots * 1$$

Beispiele: $1! = 1$ $2! = 2*1$ $3! = 3*2*1$ $4! = 4*3*2*1$

Sie geben über Tastatur eine ganzzahlige Zahl (≥ 1) ein. Von dieser Zahl soll die Fakultät berechnet werden und das Ergebnis auf dem Bildschirm ausgegeben werden.

Beispiel:

Sie geben z.B. die Zahl 4 ein. Auf dem Bildschirm wird dann ausgegeben:

$$n! = 24$$

2) 30P

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,

Erstellen Sie ein Struktogramm eines Programms, das bestimmt, ob eine ganze Zahl n ($n \geq 2$) eine Primzahl ist: Sie geben über Tastatur eine ganze Zahl n ($n \geq 2$) ein. Auf dem Bildschirm soll dann ausgegeben werden, ob diese Zahl eine Primzahl ist oder nicht.

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 Variablenamen 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)

Erstellen Sie ein Struktogramm eines Programms, das die Potenz a^n einer reellen Zahl a bestimmt.

n muß eine **ganze** Zahl sein.

Im Hauptprogramm muss der Anwender a und eine ganze Zahl n eingeben können.

Das Programm berechnet dann das Ergebnis a^n und gibt es auf dem Bildschirm aus.

Definition der Potenz:

$$a^n = \underbrace{a \cdot a \cdot \dots \cdot a}_{n \text{ - mal}}, \text{ wenn } n > 0$$

$$a^n = 1, \text{ wenn } n = 0$$

$$a^{-n} = \frac{1}{a^n}, \text{ wenn } n < 0$$

Beispiele:

$$2^4 = 2 \cdot 2 \cdot 2 \cdot 2 = 16$$

$$5^0 = 1$$

$$3^{-2} = \frac{1}{3^2} = \frac{1}{9}$$

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

Euklidische Algorithmus:

Der größte gemeinsame Teiler (ggT) kann mit Hilfe des euklidischen Algorithmus berechnet werden. Es wird so lange dividiert, bis es keinen Rest mehr gibt.

Beispiel: ggT(64, 36)

$$64 = 1 * 36 + 28$$

$$36 = 1 * 28 + 8$$

$$28 = 3 * 8 + 4$$

$$8 = 2 * 4$$

also:

$$\text{ggT}(64, 36) = 8$$

1)

Erstellen Sie ein Struktogramm eines Programms, das mit Hilfe der euklidischen Algorithmus den größten gemeinsamen Teiler (ggT) zweier ganzer Zahlen n und m berechnet und auf dem Bildschirm ausgibt.

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

Bemerkung:

Alle Teilaufgaben müssen in **einem** Programm realisiert werden

1)

Ein Bruch besteht aus einem Zähler (ganze Zahl) und einem Nenner (ganze Zahl).

a)

Erstellen Sie die Klasse Bruch mit den entsprechenden Attributen, Methoden und einem Konstruktor.

Erstellen Sie außerdem die Methode kuerzen(), die einen Bruch maximal kürzt.

Beispiel:

12/16 maximal gekürzt ergibt 3/4

b)

Machen Sie im "Hauptprogramm" main() folgendes:

b1) Erzeugen Sie den Bruch 13 / 17.

b2) Setzen Sie (mit Hilfe der entsprechenden Methode) den Zähler auf 12.

b3) Setzen Sie (mit Hilfe der entsprechenden Methode) den Nenner auf 18.

b4) Kürzen Sie diesen Bruch mit Hilfe der entsprechenden Methode.

b5) Geben Sie den Wert des gekürzten Bruchs auf dem Bildschirm aus.

Lösungen

```
package elfi_6_5_09_nr1;
public class MainE1FI_6_5_09_nr1 {
    public static void main(String[] args) {
        Bruch b = new Bruch(13, 17);           2P
        b.setZaehler(12);                     2P
        b.setNenner(18);                      2P
        b.kuerzen();                          2P
        System.out.println("Bruch=" +
            b.getZaehler()+"/"+b.getNenner() ); 5P
    }
}

class Bruch{
    private int zaehler;                      1P
    private int nenner;                      1P

    public Bruch(int pZaehler, int pNenner){  3P
        zaehler = pZaehler;
        nenner = pNenner;
    }

    public void setZaehler(int pZaehler){    3P
        zaehler = pZaehler;
    }

    public int getZaehler(){                 3P
        return zaehler;
    }

    public void setNenner(int pNenner){     3P
        nenner = pNenner;
    }

    public int getNenner(){                 3P
        return nenner;
    }

    public void kuerzen(){                   20P
        int teiler=1;
        int ggt = 1;
        int erg;

        while(teiler <= zaehler){
            if(zaehler % teiler == 0 && nenner % teiler==0){
                ggt = teiler;
            }
            teiler++;
        }
        zaehler = zaehler / ggt;
        nenner = nenner / ggt;
    }
}
```

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

Bemerkung:

Alle Teilaufgaben müssen in **einem** Programm realisiert werden

1)

Die Quersumme einer Zahl ist die Summe seiner Ziffern.

Beispiel:

Quersumme von 789 = 24

a) 37 P

Erstellen Sie die Klasse Quersumme mit den Attributen zahl und quersumme, den entsprechenden Methoden und einem Konstruktor.

Wenn ein Objekt der Klasse Quersumme erzeugt wird, müssen in zahl und quersumme zunächst die gleiche Zahl stehen.

Die Methode berechneQuersumme() berechnet die Quersumme des Attributs quersumme und speichert diesen Wert wieder in dem Attribut quersumme ab.

b) 13 P

Machen Sie im "Hauptprogramm" main() folgendes:

b1) Erzeugen Sie ein Objekt der Klasse Quersumme (das das Attribut zahl mit dem Wert 678 vorbelegt).

b2) Berechnen Sie (mit Hilfe der entsprechenden Methode) die Quersumme dieser Zahl.

b3) Berechnen Sie (mit Hilfe der entsprechenden Methode) die Quersumme dieser gerade berechneten Quersumme.

b4) Welchen Wert hat die gerade berechnete Quersumme?

b5) Lesen Sie ((mit Hilfe der entsprechenden Methode)) den Wert dieser Quersumme in eine Variable.

b6) Geben Sie den Wert dieser Quersumme auf dem Bildschirm aus.

Lösungen:

```
package elfi_7_06_09_nr1;

public class MainElFI_7_06_09_nr1 {
    public static void main(String[] args) {
        int wert;
        Quersumme q = new Quersumme(678);           2P
        q.berechneQuersumme();                       2P
        q.berechneQuersumme();                       2P
        // Quersumme ist 3                           3P
        wert=q.getQuersumme();                       2P
        System.out.println("Quersumme="+wert);      2P
    }
}

class Quersumme{
    private int zahl;                               1P
    private int quersumme;                         1P

    public Quersumme(int pZahl){                   3P
        zahl = pZahl;
        quersumme=pZahl;
    }

    public void setZahl(int pZahl){                3P
        zahl = pZahl;
        quersumme = pZahl;
    }

    public int getZahl(){                          3P
        return zahl;
    }

    public int setQuersumme(int pQuersumme){      3P
        zahl = pQuersumme;
        quersumme = pQuersumme;
    }

    public int getQuersumme(){                    3P
        return (quersumme);
    }

    public void berechneQuersumme(){              20P
        int rest;
        int summe = 0;
        int erg=quersumme;

        while(erg>0){
            rest = erg % 10;
            erg = erg/10;
            summe = summe + rest;
        }
        quersumme=summe;
    }
}
```