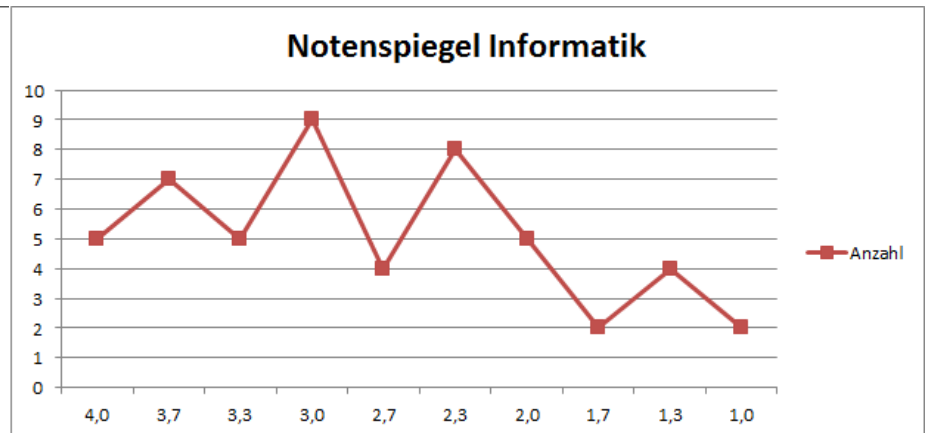


**Geoinformatik und Vermessung Prüfung 502 : Informatik (JAVA)****Ergebnisse**

teilgenommen: 73  
nicht bestanden: 22  
  
bestanden: 51  
davon 4er: 12  
davon 3er: 18  
davon 2er: 15  
davon 1er: 6

**Aufgabe 1: Tracing****4 Punkte**

Was gibt das unten stehende Java-Programm auf der Konsole aus?

```
package klausur;
public class Tracing {
    public static void main(String[] args) {
        int a = 5;
        boolean b = false;
        System.out.println(a + " / " + b);
        a = Math.max(a,2);
        b = true;
        System.out.println(a + " / " + b);
        ++a;
        b = a > 10;
        System.out.println(a + " / " + b);
        a = b ? 1 : 2;
        b = !b;
        System.out.println(a + " / " + b);
    }
}
```

**Aufgabe 2: Fehler****12 Punkte**

Das folgende Java-Programm enthält 8 syntaktische Fehler, die der Compiler erkennt und das Programm deshalb nicht übersetzen kann. Außerdem entspricht das Programm an 4 Stellen nicht den Konventionen. Markieren Sie alle 12 Fehler, geben Sie jeweils an, um welche der beiden Fehlerarten es sich handelt, und machen Sie einen Vorschlag, wie man den Fehler beseitigen könnte.

```
paggage klausur;  
  
import java.util.Arrays;  
  
public class errors {  
  
    public static void getZero() {  
  
        return -0;  
  
    }  
  
    public static void main(String[] args) {  
  
        Arrays.sort(args);  
  
        for (int i=0; i<args.length; +i) {  
  
            System.out.println(args[i]);  
  
        }  
  
        System.out.println(i);  
  
        int i = args.length;  
  
        getZero(i);  
  
        getZero(); System.out.println(true + i);  
  
        scanner.close();  
  
        boolean Ende = true;  
  
    }  
  
}  
  
}
```

**Aufgabe 3: Programmieren I****10 Punkte**

Schreiben Sie ein Java-Programm, das eine Stern-Pyramide auf der Konsole ausgibt. Die Höhe ( $h$ ) der Pyramide entspricht der Anzahl an Zeilen und wird vom Benutzer eingelesen. Beispiel-Ausgaben:

$h = 3$

```
  *
 ***
*****
```

$h = 5$

```
  *
 ***
*****
*****
*****
```

```
package klausur;
import java.util.Scanner;
public class Pyramid {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Wie hoch soll die Pyramide sein? ");
        int h = scanner.nextInt();
```

```
        scanner.close();
    }
}
```

**Aufgabe 4: Methoden aufrufen****12 Punkte**

Ergänzen Sie bei diesem Programm die Parameter-Werte der Methodenaufrufe (unten in der main-Methode), sodass die angegebene Konsolen-Ausgabe zustande kommt (siehe Kommentar).

```
public class Overloading {
    public static int test() {
        return 3;
    }
    public static boolean test(double x) {
        return x >= 5;
    }
    public static int test(int x) {
        return 2 * x - 1;
    }
    public static int test(String x) {
        return x.length()+100;
    }
    public static String test(int[] x) {
        String y = "Hallo";
        for (int i=0; i<x.length; ++i) {
            y += x[i];
        }
        return y;
    }
    public static int test(int x, boolean y) {
        if (x <= 1)
            return 1;
        return y ? x * test(x - 1, y) : x + test(x - 1, y);
    }

    public static void main(String[] args) {
        System.out.println("a: " + test(           )); // a: 5
        System.out.println("b: " + test(           )); // b: 10
        System.out.println("c: " + test(           )); // c: false
        System.out.println("d: " + test(           )); // d: Hallo
        System.out.println("e: " + test(           )); // e: 24
        System.out.println("f: " + test(           )); // f: 102
        System.out.println("g: " + test(           )); // g: true
        System.out.println("h: " + test(           )); // h: 28
    }
}
```

**Aufgabe 5: Datentypen und Operatoren**

**10 + 4 = 14 Punkte**

a) Geben Sie das Ergebnis sowie den Datentyp des Ergebnisses der folgenden Ausdrücke an.

<b>Ausdruck</b>	<b>Ergebnis</b>	<b>Datentyp</b>
true    false		
(double) 2 * 4 - 3		
9 / 2		
6 % 1		
7.2 <= 4.9 + 1		
"Pappe".replace('p','t')		
(new char[2]).length		
6 != 2 ? '4' : '8'		
6 > 2 && !false && true		
7 < 7 == 7 > 7		

b) Berechnen Sie 5 & 6. Stellen Sie dazu die involvierten Zahlen in Binär-Darstellung dar.

5 =

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--

6 =

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--

5 & 6 =

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--

Was ist das Ergebnis?

**Aufgabe 6: Methoden mit Arrays****12 Punkte**

Schreiben Sie eine Methode `difference`, die von zwei boolean-Arrays bestimmt, in wie vielen Positionen sich die beiden Arrays unterscheiden. Haben die beiden Arrays eine unterschiedliche Länge, so gilt jeder Index, der über die Länge des kleineren Arrays hinaus geht, als unterschiedlich. Schreiben Sie die Methode so, dass sie zur angegebenen main-Methode kompatibel ist.

```
public class MethodsWithArrays {  
  
    public static void main(String[] args) {  
        boolean[] a = {true,true,false,true,false};  
        boolean[] b = {true,true,true,true,false};  
        boolean[] c = {false,false,true};  
        System.out.println(difference(a,a)); // Ausgabe: 0  
        System.out.println(difference(a,b)); // Ausgabe: 1  
        System.out.println(difference(b,c)); // Ausgabe: 4  
        System.out.println(difference(c,a)); // Ausgabe: 5  
    }  
}
```

}

**Aufgabe 7: Programmieren II****7 Punkte**

Ein Mann hat  $n$  Steine vor sich. Er möchte diese Steine in möglichst viele kleine Portionen zerlegen, aber jede der kleinen Portionen soll unterschiedlich groß sein.

Schreiben Sie ein Java-Programm, das eine ganze Zahl ( $n$ ) vom Benutzer einliest und in möglichst viele Teile zerlegt, wobei keine Zahl in der Zerlegung doppelt vorkommen darf. Beispiele:

```
Eingabe: 4
Zerlegung: 1 3
```

```
Eingabe: 6
Zerlegung: 1 2 3
```

```
Eingabe: 16
Zerlegung: 1 2 3 4 6
```

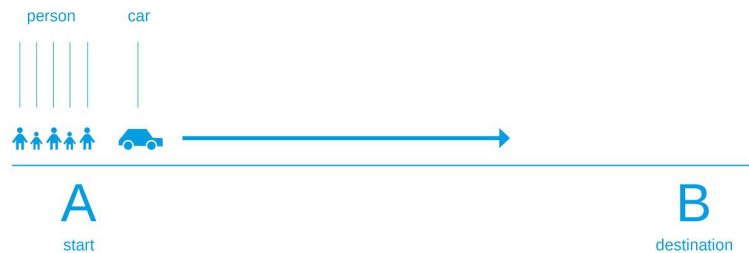
```
package klausur;
import java.util.Scanner;
public class Partition {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Eingabe: ");
        int n = scanner.nextInt();

        scanner.close();
    }
}
```

**Aufgabe 8: Klassen****16 Punkte**

Eine Personengruppe soll von A (start) nach B (destination) transportiert werden. Es steht allerdings nur ein Auto mit begrenzter Kapazität zur Verfügung und nicht jede Person hat einen Führerschein. Daher muss eventuell mehrmals hin und her gefahren werden.



Gegeben sind die zwei folgenden Klassen mit den beschriebenen Instanzmethoden.

```
public class Person {  
  
    public void enterCar(Car car) { ... }  
        Lässt die Person in das als Parameter angegebene Auto steigen  
        car: Das Auto, das betreten werden soll  
  
    public boolean hasLicence() { ... }  
        Gibt zurück, ob die Person einen Führerschein hat  
        return (true = hat einen Führerschein, false = hat keinen Führerschein)  
  
    public void driveToDestination() { ... }  
        Wenn die Person sich in einem Auto befindet und einen Führerschein hat,  
        lässt diese Methode sie das Auto zum Ziel fahren  
  
    public void driveToStart() { ... }  
        Wenn die Person sich in einem Auto befindet und einen Führerschein hat,  
        lässt diese Methode sie das Auto zum Start fahren  
  
}  
  
public class Car {  
  
    public int getCapacity() { ... }  
        Gibt die Gesamtzahl an Sitzplätzen im Auto zurück  
        return (Die maximale Anzahl an Personen, die mit diesem Auto fahren können)  
  
    public void empty() { ... }  
        Alle Personen, die sich in diesem Auto befinden, steigen aus  
  
}
```



Programmieren Sie die Methode `public static void transport(Person[] p, Car c)`. Sie soll das als Parameter übergebene Array von Personen mit Hilfe des als Parameter übergebenen Autos zum Ziel bringen. Beachten Sie dabei, dass im Auto immer nur so viele Personen sein dürfen, wie es die Kapazität des Autos zulässt. Außerdem können nur Personen, die einen Führerschein besitzen, das Auto fahren. Gehen Sie davon aus, dass mindestens eine Person mit Führerschein im Array ist.

```
public class Transportation {  
    public static void transport(Person[] p, Car c) {
```

```
    }
```

```
}
```

**Aufgabe 9: Fehlerbehandlung****6 Punkte**

Das folgende Java-Programm berechnet den Betrag einer ganzen Zahl. Es stürzt allerdings ab, wenn der Benutzer keine ganze Zahl eingibt (`InputMismatchException`). Ergänzen Sie das Programm um exception handling und sorgen Sie dafür, dass der Benutzer bei einem Fehler immer wieder erneut die Möglichkeit erhält eine Zahl einzugeben.

```
package klausur;

import java.util.InputMismatchException;
import java.util.Scanner;

public class Abs {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Eingabe: ");
        int value = scanner.nextInt();

        System.out.println("Der Betrag davon ist " + Math.abs(value));

        scanner.close();

    }
}
```

**Aufgabe 10: Wissensabfrage****7 Punkte**

Bei den folgenden Aufgaben gibt jede korrekt angekreuzte Antwort einen halben Punkt, jede falsch angekreuzte Antwort einen halben Punkt Abzug. Insgesamt gibt es bei dieser Aufgabe jedoch mindestens 0 Punkte.

a) Was kann man mit dem Eclipse-Debugger tun und was nicht? Kreuzen Sie an!

	<b>Ja</b>	<b>Nein</b>
Den Wert einer Variablen verändern		
Den Namen einer Variablen verändern		
Den Datentyp einer Variablen verändern		
Programmablauf anhalten		
Programmablauf fortsetzen		

b) Welche der folgenden Variablen kann mit null initialisiert werden?

	<b>Ja</b>	<b>Nein</b>
String a		
boolean[] b		
char c		
double d		
Scanner e		
float[] f		

c) Auf ein Attribut, das als „private“ deklariert ist, kann man zugreifen...

	<b>Ja</b>	<b>Nein</b>
innerhalb der Klasse, in der das Attribut deklariert wurde		
innerhalb des Pakets, in dem sich die Klasse befindet		
auch außerhalb des Pakets, in dem sich die Klasse befindet		