



## Aufgabe 1 (12 Punkte)

Programmieren Sie die Methode `public static void printMinMax` aus. Die Methode soll den Wert des kleinsten und des größten Elementes im Array `a` mit Hilfe von `System.out.print` ausgeben. Denken Sie auch an den Fall eines leeren Feldes.

```
public class Aufgabe1 {  
  
    public static void printMinMax(int[] a) { // jetzt sind Sie dran!  
        if (a.length==0) System.out.println(„Array ist leer“);  
        else {  
            int max=a[0]; min=a[0];  
            for (int i=1;i<a.length;i++) {  
                if (a[i]<min) min := a[i];  
                if (a[i]>max) max := a[i];  
            }  
            System.out.println(“Min = ”+min+”, Max = ”+max);  
        }  
    }  
}
```

## Aufgabe 2 (8 Punkte)

Gegeben sei folgende rekursive Java-Funktion  $f$  für positive Argumentwerte  $a, b$ :

```
public static int f(int a, int b) {  
    if (a>b)  
        return f(a-b,b);  
    else if (a<b)  
        return f(a,b-a);  
    else return a;  
}
```

a) Welcher Wert wird  $k$  durch den Aufruf

$\text{int } k=f(4,6);$  zugewiesen?

Begründen Sie Ihre Antwort, indem Sie das Ergebnis jedes Aufrufs der Funktion  $f$  in Reihenfolge protokollieren. Geben Sie die Argumente des jeweiligen Aufrufs an.

Aufruf mit Parameter	Ergebnis des Aufrufes
$f(4,6)$	2
$f(4,2)$	2
$f(2,2)$	2

b) Was berechnet  $f$ ? Geben Sie eine mathematische Formel oder eine verbale Beschreibung der Funktion an.

$f$  berechnet den ggT von  $a$  und  $b$ .

Bew.: Wenn  $a=b$  ist, ist  $a=b=\text{ggT}(a,b)$ , das liefert der 2. else-Teil.

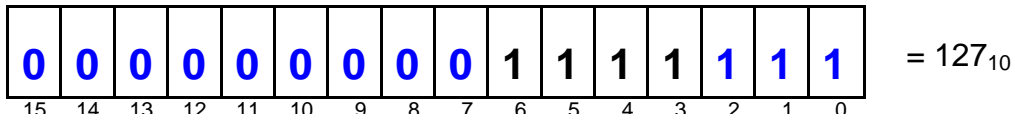
Sei nun  $a>b$ . Wir zeigen, dass  $\text{ggT}(a,b)=\text{ggT}(a-b,b)$ . Aus der Def. des ggT folgt:  $a=x*\text{ggT}(a,b)$ . und  $b=y*\text{ggT}(a,b)$  mit teilerfremden  $x$  und  $y$ . Wir berechnen daraus  $a-b=(x-y)*\text{ggT}(a,b)$ , also ist  $\text{ggT}(a,b)$  auch Teiler von  $a-b$ . Da  $y$  und  $x-y$  ebenfalls teilerfremd sind, ist  $\text{ggT}(a,b)=\text{ggT}(a-b,b)$ . Der Fall  $b>a$  geht analog.

### Aufgabe 3 (12 Punkte)

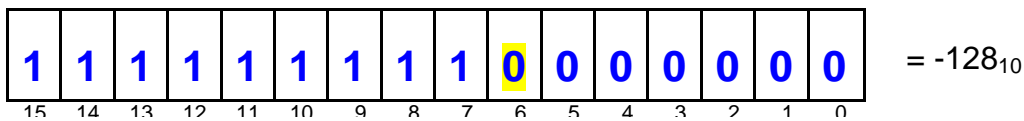
Wir verwenden im Folgenden die Binärdarstellung ganzer Zahlen im Zweierkomplement mit 16 Bit.

- a) Geben Sie die Binärdarstellung von  $127_{10}$  an!
- b) Geben Sie die Binärdarstellung von  $-128_{10}$  an!
- c) Welche Dezimalzahl hat die Binärdarstellung  $111111111100010$  ?

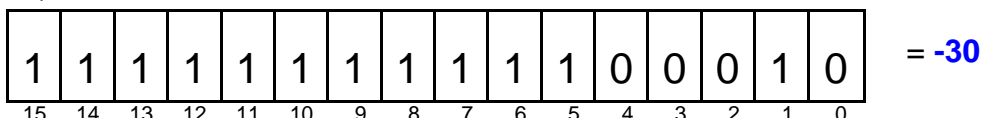
a)



b)



c)



## Aufgabe 4 (12 Punkte)

a) Formalisieren Sie die Aussagen 1-6 über die Gesamtheit aller Häuser in Aussagenlogik, indem Sie entsprechende aussagenlogische Formeln angeben. Verwenden Sie die fett gedruckten Anfangsbuchstaben als Variablennamen (z.B. G für das Haus ist **gelb**).

1. Ein Haus ist **gelb** oder **blau** oder **weiß**.
2. Ein Haus hat ein **Reetdach**, ein **Ziegeldach** oder ein **Schieferdach**.
3. Wenn ein Haus ein **Reetdach** hat und nicht **gelb** ist, dann ist es **weiß**..
4. Häuser, die **weiß** oder **gelb** sind, haben **Ziegeldächer** oder **Schieferdächer**.
5. **Gelbe** Häuser haben kein **Schieferdach**.
6. Häuser mit einem **Schieferdach** oder mit einem **Reetdach** sind nicht **blau**.

Hinweis: Sie sollen nicht ausrechnen, ob es überhaupt Häuser gibt, die diesen Bedingungen genügen, Sie müssen nur die Aussagen formalisieren!

1.  $G \vee B \vee W$
2.  $R \vee Z \vee S$
3.  $R \wedge \neg G \Rightarrow W$
4.  $W \vee G \Rightarrow Z \vee S$
5.  $G \Rightarrow \neg S$
6.  $S \vee R \Rightarrow \neg B$

b) Überprüfen Sie mit Hilfe einer Wahrheitstabelle, ob  $(q \Rightarrow p) \wedge p$  eine Tautologie ist, d.h. für alle Belegungen der Variablen wahr ist.

$p$	$q$	$q \Rightarrow p$	$(q \Rightarrow p) \wedge p$
<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>
<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>
<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>
<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

- $(q \Rightarrow p) \wedge p$  ist allgemeingültig
- $(q \Rightarrow p) \wedge p$  ist nicht allgemeingültig

## Aufgabe 5 (12 Punkte)

Implementieren Sie eine Klasse **Zeitspanne**, die eine Zeitspanne in Tagen, Stunden, Minuten und Sekunden festlegt. Die interne Speicherung sei normiert, d.h. die Werte von Sekunden und Minuten liegen zwischen 0 und 59, die Stunden zwischen 0 und 23. Tag kann beliebig groß (im Wertebereich von long) werden. Wir betrachten nur positive Zeitspannen. Wir machen folgende Vorgaben:

```
public class Zeitspanne {
    protected int s; // Sekunden 0<=s<60
    protected int m; // Minuten 0<=m<60
    protected int h; // Stunden 0<=s<24
    protected long d; // Tage d>=0
}
```

- a) Implementieren Sie einen Konstruktor, der die Zeitspanne der Dauer d Tage, h Stunden, m Minuten und s Sekunden erzeugt: Denken Sie an den Aufruf der Methode `normiere()` (siehe b)), damit auch nicht normierte Übergabewerte verarbeitet werden können.

```
public Zeitspanne (long d, int h, int m, int s) { // hier ergänzen
    this.s=s;
    this.m=m;
    this.h=h;
    this.d=d;
    normiere();
}
}
```

- b) Vervollständigen Sie die Methode `normiere`, die die Zeitwerte wie anfangs beschreiben normiert.

```
void normiere() {
    //normiere Sekunden
    while (s>=60) {
        s -=60;
        m++;
    }
    // normiere Minuten
    while (m>=60) {
        m -=60;
        h++;
    }

    // normiere Stunden
    while (h>=24) {
        h -=24;
        d++;
    }
}
```

## Aufgabe 5 Forts.

- c) Implementieren Sie eine Methode `add`, die eine Zeitspanne aufaddiert. Denken Sie an die notwendige Normierung! Beispiel `Zeitspanne(0,0,0,20).add(0,0,0,100)` liefert dieselbe Zeitspanne wie `Zeitspanne(0,0,2,0)`.

```
public void add(int d1, int h1, int m1, int s1) {  
    s +=s1;  
    m +=m1;  
    h +=h1;  
    d +=d1;  
    normiere()  
}
```

## Aufgabe 6 (16 Punkte)

Diese Aufgabe umfasst 4 Multiple-Choice Cluster mit je 4 Ankreuzfragen. Für jedes Cluster gilt: wenn alle 4 Kreuze an der richtigen Stelle stehen, gibt es 4 Punkte für das Cluster. Ein falsches Kreuz gibt einen Punkt Abzug. Wer 2 richtige und 2 falsche Kreuzchen in einem Cluster macht, erhält  $1+1-1-1=0$  Punkte. Zum Trost: es gibt keine negativen Gesamtpunktzahlen, jedes Cluster bringt 0 bis 4 Punkte.

		Ja	Nein	Frage
a)				<b>Grundlagen</b>
	1.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Unter Komplexität versteht man den Speicherbedarf eines Programmes.
	2.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Jede mathematische Funktion ist berechenbar.
	3.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	{true} ist eine zulässige Eingabespezifikation für einen Algorithmus.
	4.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ausleeren eines Hallenbad-Beckens mit einem Teelöffel ist ein effektiver Algorithmus.
b)				<b>Java</b>
	1.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<code>long int real;</code> ist korrekter Java-Code.
	2.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Jede nicht abstrakte Java-Klasse muss eine main-Methode besitzen.
	3.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ein Java-Objekt kann aus dem Speicher entfernt werden, wenn keine Referenz mehr auf das Objekt zeigt,
	4.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Wenn eine Klasse keinen Konstruktor hat, meldet der Compiler einen Fehler.
c)				<b>Programmieren</b>
	1.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Java kennt keine Rekursion in mehreren Argumenten.
	2.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Ein Programm ist genau dann partiell korrekt, wenn es für mindestens eine Eingabe ein korrektes Ergebnis berechnet.
	3.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Die Datenstruktur „Schlange“ (queue) verwendet das First-in-first-out-Prinzip
	4.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<code>for (int i; i&lt;10; i--) System.out.print(i);</code> ist korrekter Java-Code.
d)				<b>Logik</b>
	1.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	$(a \vee b) \wedge (\neg c \vee d)$ ist in konjunktiver Normalform
	2.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	A ist allgemeingültig, falls $\neg A$ unerfüllbar ist.
	3.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	$a \Rightarrow a$ ist allgemeingültig
	4.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Für jede erfüllbare aussagenlogische Formel A gilt: $A \vee \neg A \equiv true$