

Bereich: Operatoren

Inkrement und Dekrement

Package: de.dhbwka.java.exercise.operators

Klasse: IncrementDecrement

Aufgabenstellung:

Was gibt das folgende Programm `IncrementDecrement` aus?

Lösen Sie die Aufgabe zunächst „auf Papier“, kontrollieren Sie das Ergebnis dann am Rechner!

```
package de.dhbwka.java.exercise.operators;

public class IncrementDecrement {

    public static void main(String[] args) {
        int i=0;
        int j=0;
        j = ++i;
        int k = j++ + ++i;
        System.out.println("k: " + k);
        System.out.println("*: " + j++ + ++i);
        System.out.println(j++ + ++i);
        int m = j++ * ++i;
        System.out.println("m: " + m);
        int n = --j * --i;
        System.out.println("n: " + n);
        System.out.println("i: " + i);
        System.out.println("j: " + j);
    }
}
```

Bereich: Operatoren

Kaufmännisches Runden (2)

Package: `de.dhbwka.java.exercise.datatypes`

Klasse: `Round`

Aufgabenstellung:

Testen Sie Ihre Lösung der Aufgabe „Kaufmännisches Runden“ des Übungsblatts Primitive Datentypen mit negativen Zahlen, z.B. $-\pi$ bzw. $-e$!

Ist die Rundung korrekt? Falls nicht, entwickeln Sie eine Variante, die sowohl positive als auch negative Zahlen korrekt rundet. Verwenden Sie hierzu den Bedingungsoperator.

Bereich: Operatoren

Ostertermin berechnen

Package: de.dhbwka.java.exercise.operators

Klasse: Easter

Aufgabenstellung:

Berechnen Sie in einem Java-Programm `Easter` den Ostertermin für das Jahr „jahr“ nach der von Gauß im Jahr 1816 veröffentlichten Formel (hier für den gregorianischen Kalender).

```
a = jahr mod 19
b = jahr mod 4
c = jahr mod 7
k = jahr div 100
p = (8k + 13) div 25
q = k div 4
m = (15 + k - p - q) mod 30
n = (4 + k - q) mod 7
d = (19a + m) mod 30
e = (2b + 4c + 6d + n) mod 7
ostern = (22 + d + e)    (bezogen auf 1. März)
```

Der „32. März“ entspricht also dem 1. April usw.

Lassen Sie die Jahreszahl vom Nutzer eingeben und geben Sie den passenden Ostertermin aus!

Hinweise:

`div` liefert den Ganzzahlanteil des Ergebnisses der ganzzahligen Division.

`mod` (Modulo-Funktion) liefert den Divisionsrest bei ganzzahliger Division.

Bereich: Operatoren**Vorrang von Operatoren****Package:** de.dhbwka.java.exercise.operators**Klasse:** Priority**Aufgabenstellung:**

Was gibt das folgende Programm `Priority` aus?

Lösen Sie die Aufgabe zunächst „auf Papier“, kontrollieren Sie das Ergebnis dann am Rechner!

```
package de.dhbwka.java.exercise.operators;
```

```
public class Priority {
```

```
    public static void main(String[] args) {
        System.out.println("1: " + (5 / 2 * 2));
        System.out.println("2: " + (9. / 2 * 5));

        boolean a = true, b = false, c = false;
        System.out.println("3: " + (a && b || c));

        char ch = 'c';
        System.out.println("4: " + ('a' + 1 < ch));

        int i = 1, j = 2, k = 3;
        System.out.println("5: " + (-i - 5 * j >= k + 1));

        i = 1;
        if (a || (++i == 2)) {
            System.out.println("6: " + i);
        }

        i = 1;
        if (a | (++i == 2)) {
            System.out.println("7: " + i);
        }
    }
}
```