

Aufgabe 1: "Dezimal-Umrechnung"**[12 Punkte]**

Gesucht ist ein Programm, welches eine **positive Dezimalzahl** in ein **anderes Zahlensystem** "umrechnet".

Folgende Zahlensysteme sollen berücksichtigt werden:

Binär-System	Oktal-System	Hexadezimal-System
0, 1	0 bis 7	0 bis 9 und A = 10 B = 11 C = 12 D = 13 E = 14 F = 15

Vervollständigen Sie dazu folgendes Programm:

```
class Convert {
    public static void main(String [] args) {

        Out.print("\nDezimal-Konverter");
        Out.print("\n=====");

        // Eingabe einer positiven Dezimalzahl
        int n = inputDecimal();

        // Eingabe des Zielsystems b=Binaer, o=Oktal, h=Hexadezimal
        char system = inputTargetSystem();
        // system: 'b'=Binaer, 'o'=Oktal, 'h'=Hexadezimal

        // Konvertiere in Zielsystem
        String targetRepresentation = convertTo(n, system);
        // targetRepresentation: e.g. "A7BC", "10100", "2147"

        // Gib konvertierte Repraesentation im entsprechenden Zielsystem aus
        Out.print("\n" + n + " im "
            + printTargetSystem(system) + " ist " + targetRepresentation);

    } // main()

    // TODO -----
} // end Convert
```

Beispiele:

```
Dezimal-Konverter
=====
Dezimalzahl: -99
    keine gueltige Dezimalzahl!
Dezimalzahl: 17

Zielsystem [b=Binaer, o=Oktal, h=Hexadezimal]: x
    keine gueltige Auswahl!
Zielsystem [b=Binaer, o=Octal, h=Hexadezimal]: o

17 im Oktal-System ist 21
```

```
Dezimal-Konverter
=====
Dezimalzahl: 3

Zielsystem [b=Binaer, o=Oktal, h=Hexadezimal]: b

3 im Binaer-System ist 11
```

```
Dezimal-Konverter
=====
Dezimalzahl: 123

Zielsystem [b=Binaer, o=Oktal, h=Hexadezimal]: h

123 im Hexadezimal-System ist 7B
```

Hinweise:

- Überlegen Sie welche Methoden es zu realisieren gilt.
 - Überlegen Sie auf Grund der Verwendung der einzelnen Methoden, welche Parameter und Rückgabewerte diese Methoden aufweisen müssen.
 - Finden Sie selbständig ggf. weitere sinnvolle Methoden.
 - Die Zahlenrepräsentation im Zielsystem soll ein `String` sein.
- Folgendes hilft Ihnen dazu vielleicht weiter:

```
static String helloWorld() {
    String s = "Hello";
    s = s + " World!";
    return s;
}
```

Abgabe:

→ Mit Ihrem Code **vervollständigtes JAVA-Programm "Convert"**.

Aufgabe 2: "Plattenweg"**[12 Punkte]**

Gesucht ist eine Methode, welche ermittelt, **ob ein Weg**, mit einer gegebenen Länge **way**, durch Platten unterschiedlicher Länge **genau gepflastert** werden kann (dh. die Platten dürfen nicht geteilt werden, aber jedewede Kombination der Platten kann gewählt werden), und **falls ja, wie viele Platten minimal** benötigt werden.

Es stehen dazu folgende Platten mit der jeweiligen Länge zur Verfügung:

Platte A 23 cm	Platte B 37 cm	Platte C 53 cm
-----------------------	-----------------------	-----------------------

Die Pflasterung kann wie folgt erfolgen:

$$\text{noPlates}(\text{way}) = \begin{cases} \text{keine Belegung möglich,} & \text{falls } \text{way} < \min(A, B, C) \\ 1, & \text{falls} \\ & \text{way} = A \text{ oder} \\ & \text{way} = B \text{ oder} \\ & \text{way} = C \\ 1 + \min(\text{noPlates}(\text{way} - A), & \text{sonst, falls Belegung möglich} \\ & \text{noPlates}(\text{way} - B), \\ & \text{noPlates}(\text{way} - C)) \end{cases}$$

Vervollständigen Sie dazu folgendes Programm.

```

class Paving {

    // TODO - noPlates(way) -----

    public static void main(String [] args) {

        Out.print("\nWay: ");
        int way = In.readInt();
        // berechne wie viele Platten benoetigt werden
        int noPlates = noPlates(way);
        if (noPlates == -1) {
            Out.print("Weg der Laenge " + way
                + " mit Platten der Laenge "
                + A + ", " + B + " bzw. " + C
                + " nicht genau auslegbar!");
        } else {
            Out.print("Weg der Laenge " + way
                + " mit " + noPlates + " Platten der Laenge "
                + A + ", " + B + " bzw. " + C
                + " genau auslegbar!");
        }
    } // end main()

} // end class Paving

```

Hinweise:

- Für die Belegung mit Platten sind verschiedene Alternativen möglich, von der ausgehend weitere Alternativen möglich sind. Wählen Sie daher eine **rekursive Lösung**.
- Überlegen Sie im Besonderen, mit welchem Wert ausgedrückt wird, dass keine Belegung möglich ist. Berücksichtigen Sie dies bei Ihrem Vorgehen beim Finden des kürzesten Weges.
- Finden Sie selbständig ggf. weitere sinnvolle Methoden.

Abgabe:

➔ Mit Ihrem Code **vervollständigtes JAVA-Programm "Paving"**.