

Zu bearbeitende Übungen 3.15 – 3.21

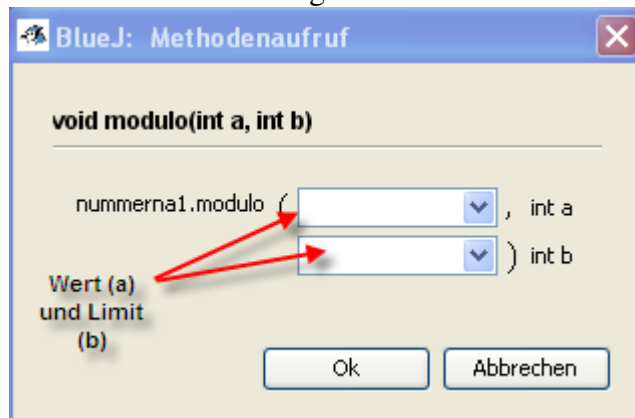
3.15 Erklären sie den Modulo-Operator. Möglicherweise benötigen Sie dazu weitere Quellen (Java-Ressourcen online, andere Java-Bücher ect.), um Details nachzulesen.

Der Modulo-Operator wird dazu verwendet wenn ein Wert bei einem vorher gesetztem *limit* zurück auf „0“ gesetzt werden muss. Z.B. beim Umschalten einer Uhr von 23:59 Uhr auf 00:00 Uhr.

3.16 Was ist das Ergebnis des Ausdrucks $(8 \% 3)$?

Das Ergebnis des Ausdrucks $(8 \% 3)$ wäre demnach 2 mit dem Rest 2, der Modulo-Operator gibt aber nur den Rest der Division wieder, in diesem Fall also 2.

Für a wird 8 für b 3 eingesetzt.



3.17 Welches sind die möglichen Werte des Ausdrucks $(n \% 5)$, mit n als einer Variablen vom Typ *int*?

Wenn n eine Variable des Typs *int* ist ergeben sich mögliche Werte im Bereich von -4 bis 4, da das Limit der Aufgabe 5 ist und dort der Wert n wieder auf 0 gesetzt wird und der Rest angezeigt wird.

3.18 Welches sind die möglichen Werte des Ausdrucks $(n \% m)$, mit n und m als Variablen vom Typ *int*?

Wenn n und m Variablen des Typs *int* sind kann der Rest der Division maximal $-m+1$ bis $m-1$ sein.

3.19 Erklären Sie ausführlich, wie die Methode *erhoehen* funktioniert.

Die Methode *erhoehen* funktioniert relativ einfach, sie erhöht den Wert jedes Mal um +1, dass passiert folgender Maßen:

```
{ wert = wert + 1 % limit }
```

Das *limit* beschreibt hier die maximale grÖÙe, den der *wert* haben kann. Zum Beispiel die Uhr, der Minutenzeiger wird immer um eins erhht. Sobald er aber die „60“ erreichen wrde wird er durch das *limit* begrenzt und auf „0“ zurckgesetzt.

```
wert = 59 + 1 % 60; also 0
```

Limit:

```
public class Nummernanzeige
{
    private int limit;
    private int wert;

    public Nummernanzeige(int anzeigeLimit)
    {
        limit = anzeigeLimit;
        wert = 0;
    }

    public int gibWert()
    {
        return wert;
    }
}
```

limit wird initialisiert (points to `private int limit;`)

Limit setzten (points to `limit = anzeigeLimit;`)

3.20 Schreiben sie die Methode erhhen so um, dass sie anstatt des Modulo-Operatos eine *if*-Anwendung benutzt. Welche Lsung ist besser?

Modulo-Operator:

```
public void erhhen()
{
    wert = (wert + 1) % limit;
}
```

Modulo Operator (points to `%`)

if-Anweisung:

```
public void erhhen()
{
    wert();
    if (wert() >= 0)
    {
        wert + 1;
    }
    anzeigeAktualisieren();
}
```

Der Modulo-Operator ist unkomplizierter und verursacht wesentlich weniger Schreibarbeit. Er kann jedoch den Wert einfach nur erhhen. Die if-Anweisung ist in sofern besser da sie auf eine nderung reagieren kann. Sobald hier die Minuten wieder auf den Wert 0 zurckgesetzt werden erhht sie die Stunde um 1.

3.21 Testen Sie die Klasse *Nummernanzeige* im *Zeitanzeige*-Projekt in BlueJ, indem Sie einige Objekte erzeugen und ihre Methoden aufrufen.