

**Bereich: Primitive Datentypen**

**Überlauf, Zweierkomplement**

Schwierigkeit: ★☆☆☆☆

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

**Klasse:** `ShortValue`

**Aufgabenstellung:**

Schreiben Sie ein Programm `ShortValue`, in dem Sie eine `short`-Variable mit dem Wert 32767 initialisieren und den Wert der Variablen ausgeben!

Erhöhen Sie dann den Wert der Variablen um 1 und geben Sie den neuen Wert der Variablen aus! Welcher Wert wird ausgegeben und warum? Gibt es eine Fehlermeldung des Laufzeitsystems?

**Hinweis:**

Für die Ausgabe der Variablenwerte kann das Beispielprogramm `Quotient` im Kapitel „Die Programmiersprache Java“ als Vorbild genommen werden.

## Bereich: Primitive Datentypen

### Kaufmännisches Runden (1)

Schwierigkeit: ★☆☆☆☆

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

**Klasse:** Round

#### Aufgabenstellung:

Das „Casten“ (dt. die Typumwandlung) von Gleitkommazahlen in ganze Zahlen bedeutet im Allgemeinen einen Informationsverlust, denn die Nachkommastellen werden abgeschnitten, z.B. erfolgt im folgenden Programmstück die Ausgabe von 3 bzw. 2 für die ganzzahligen Varianten von  $\pi$  bzw.  $e$ .

```
double pi = 3.1415926; // Naehung der Kreiszahl Pi
double e  = 2.7182818; // Naehung der Eulerschen Zahl e

int piInt = (int) pi;
int eInt  = (int) e;

System.out.println("Pi ganzzahlig: " + piInt); // Ausgabe: 3
System.out.println("e ganzzahlig:  " + eInt);  // Ausgabe: 2
```

Mit anderen Worten: Nachkommastellen werden immer abgeschnitten bzw. es wird unabhängig vom Wert der Nachkommastellen immer abgerundet.

Schreiben Sie ein Programm `Round`, welches das obige Programm-Fragment so abändert, dass die Zahlen  $\pi$  bzw.  $e$  kaufmännisch gerundet werden, d.h. wenn der Nachkommawert  $\geq 0.5$  ist wird aufgerundet, ansonsten abgerundet!