

Hausaufgabe 3

Abgabe: 19.11.2023 (23:59 Uhr)

Abgabe:

Für jede Aufgabe ist genau eine .java Datei im Moodle hochzuladen, insgesamt also diesmal **zwei Dateien**. Nur ein Upload führt zu keiner Bewertung. Die Bewertung gibt es ausschließlich persönlich im Praktikum (als Hausaufgabenabnahme), melden Sie sich dazu beim Übungsleiter oder Tutor*innen. Jedes Programm muss für sich lauffähig sein. Nehmen Sie den Quellcode aus der SU als Basis, wenn es sich anbietet.

Aufgabe 0: Natürlicher Logarithmus

Schreiben Sie ein Programm *LogarithmTwo*, zur Berechnung des natürlichen Logarithmus von 2, $\ln(2)$. (Sie müssen für diese Aufgabe *nicht* verstehen, was der Logarithmus ist oder tut.) Die Dezimalzahlentwicklung von $\ln(2)$ beginnt mit 0,6931471805599453094172321214581, was man sich in Java durch den Ausdruck `Math.log(2)` berechnen lassen kann (diesen können Sie zum Vergleich/Kontrolle verwenden). Nutzen Sie für Ihre Näherung die Formel

$$\ln(2) = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \dots$$

Beenden Sie die Berechnung, wenn die Näherungslösung über die Summanden weniger als 0.0023 vom echten Wert abweicht. Ab dem wievielten Summanden ist diese Genauigkeit erreicht? (Antwort zur Kontrolle: 217). Schreiben Sie Ihr Programm so, dass es die Anzahl benötigter Summanden in einer Schleife ermittelt und am Ende diese Zahl auf die Konsole ausgibt. Ebenfalls soll die berechnete Näherungslösung von $\ln(2)$ sowie der Referenzwert `Math.log(2)` ausgegeben werden.

Aufgabe 1: Einser

Schreiben Sie ein Programm *OneByOne*, welches eine positive Integerzahl n von der Konsole einliest und ausgibt, wie viele Einser (1) die binäre Darstellung dieser Zahl hat.

Verwenden Sie in Ihrer Schleife nur die Bit-Operationen Bit-Shift (`>>`) und Bit-weises Und (`&`) .

Außerdem dürfen Sie Zuweisung (`=`) und Addition (`+`) und Subtraktion (`-`) verwenden, sofern erforderlich.

Die Verwendung von Java-Hilfsmethoden, wie `Integer.bitCount()`, ist nicht erlaubt.

Beispiel: Für die Integerzahl 8 ist die Ausgabe 1, für die Integerzahl 1023 ist die Ausgabe 10.

Knobel-Aufgabe: Programmierkunst

Die Klasse *Beer* in der Datei *Beer.java* (Download aus Moodle) verwendet Zeichenkettenvariablen (String), die mit dem `+` Operator zusammengefügt werden können. Kompilieren Sie die Datei *Beer.java* und führen Sie sie aus. Was tut das Programm? Wie Sie sehen, ist dem Compiler egal, wie das Programm formatiert ist.

Formatieren Sie es neu nach den von uns genutzten Regeln.

(Keine Bewertung, aber ein guter Lerneffekt zu sehen, dass Code-Formatierung nützlich ist.)