

Aufgabenblatt 05

Bearbeitungsende: 13.11.2022

Installieren Sie zur Nutzung des Debuggers eine IDE Ihrer Wahl. Im Internet, ggf. auch im LEA-Ordner Dokumentation, finden Sie Hinweise zu ihrer Installation, Konfiguration und Verwendung. Nehmen Sie sich Zeit, sich damit vertraut zu machen.

Portieren Sie zur Übung ein Beispiel aus der Vorlesung oder eine Lösung einer Übungsaufgabe, das/die aus mehreren Klassen besteht, in Ihre IDE und führen Sie es/sie dort aus.

Aufgabe 1 [Programmierung – nicht einzureichen]

Verwenden Sie den Debugger, um die verbleibenden Fehler in der Methode `istPrimzahl` in der Klasse `MatheBug` zu finden, die in der Vorlesung vorgestellt wurde.

Aufgabe 2 [Programmierung – nicht einzureichen]

Die (in LEA bereitgestellte) Klasse `MatheBug` enthält folgende Methoden, die fehlerhaft implementiert sind:

- (a) `min(int a, int b, int c, int d)` soll das Minimum der 4 übergebenen Zahlen zurückgeben.
- (b) `mean(int a, int b, int c, int d)` soll den ganzzahligen (ggf. Abschneidung nach dem Komma, kein Runden) Mittelwert der 4 übergebenen Zahlen zurückgeben.
- (c) `nextToMean(int a, int b, int c, int d)` soll den Wert der 4 übergebenen Zahlen zurückgeben, der am nächsten am Mittelwert liegt.

Liegen mehrere Argumente gleich nah am Mittelwert, so soll das letzte von ihnen ausgewählt werden (siehe Beispiel).

- (d) `eingabe` nimmt als Argumente einen `java.util.Scanner` und einen `java.io.PrintStream` an. Die Methode soll wiederholt – solange vorhanden – 4 Werte aus dem Scanner lesen und verarbeiten. Letzteres heißt, dass die Werte als Argumente an die obigen Methoden übergeben und die Resultate in der folgenden Weise in den Stream ausgegeben werden sollen:

```
Eingabe:_2_-4_6_8
Minimum:_4
Mittelwert:_2
am naechsten am Mittelwert:_6
```

Finden Sie mit Hilfe des Debuggers die Fehler und korrigieren Sie sie in einer ansonsten identischen Klasse `Mathe`. Versuchen Sie bei der Korrektur (es gibt viele Ansätze), den Code so wenig wie möglich zu ändern.

Hinweis: Betrachten Sie im Debugger den Programmverlauf z.B. für die folgenden 5 Eingaben:

<leere Eingabe>

-2 -4 -6 -8

2 4 6 8 0 0 0 0

3 12 62 82 90 2 62 77 12 52 26 17

3 12 62 82 90 13 2 62 55 66 77 11 45 12 75 45

Aufgabe 3 [Programmierung – nicht einzureichen]

Nach dem Jahreswechsel 2008 nach 2009 waren Musikabspielgeräte eines großen Herstellers zunächst nicht mehr benutzbar, weil im Programmcode zur Berechnung des Datums ein Fehler vorhanden war – wie man leider erst nachträglich analysierte.

In der (in LEA bereitgestellten) Klasse `DatumBug` liegt in der Methode `jahr` in ähnlicher Weise fehlerhafter Code vor. Aus einer gegebenen Anzahl von Tagen soll darin die Jahreszahl berechnet werden, indem so lange von der Tagesanzahl 365 bzw. 366 abgezogen wird, bis die Anzahl der verbliebenen Tage kleiner als 366 ist.

Die Methode erhält als Eingabe eine Zahl, die die Anzahl der Tage ab dem 01.01.1980 darstellt. Für den 01.01.1980 hat sie den Wert 1.

Verwenden Sie den Debugger, um den Fehler zu finden, und beheben sie diesen.

Hinweise:

- Der 01.01.2009 war der 10594. Tag nach dieser Zählung.
- Der Fehler tritt nicht nur beim Wechsel nach 2009 auf.
- Setzen Sie einen geeigneten *Conditional Breakpoint*.

Lösungen zu mit [Programmierung] markierten Aufgaben sind im **Praktomat** einzureichen.

Lösungen zu mit [Programmierung – nicht bewertet] markierten Aufgaben können ebenfalls im **Praktomat** eingereicht werden, werden jedoch nicht bewertet.

Allgemeine **Fragen** zu den Aufgaben können Sie im **LEA-Forum „Übungsaufgaben“** stellen.

Hilfe bei der Lösung der Aufgaben erhalten Sie in den **Übungen** und in der **Studierwerkstatt**.