


Artemis 6.4.0

Course Overview

  ge64baw

Courses > Praktikum: Grundlagen der Programmierung WS22/23 > Exercises > W02H03 - Wurzelhilfe fuer Fortgeschrittene

W02H03 - Wurzelhilfe fuer Fortgeschrittene

Hausaufgabe

Hard

Submission due: 9 months ago

Complaint due: 9 months ago

Points: 6 of 6

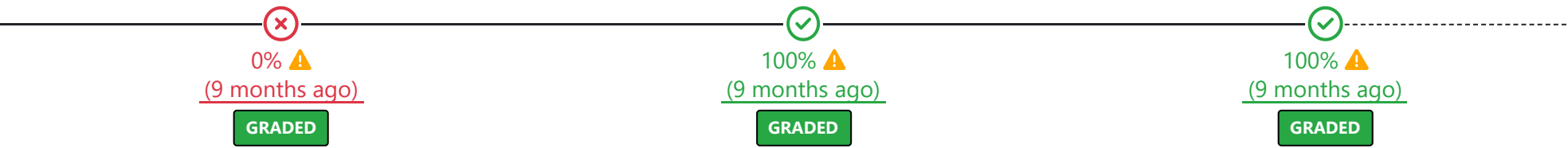
Assessment: automatic ?

 Resume practice in exercise

 100%  (9 months ago)

GRADED

Recent results:



Show all results

Tasks:

W02H03 - Wurzelhilfe für Fortgeschrittene

Die junge Mathuinin Ruth braucht dringend deine Hilfe. In der Pinguin Schule steht aktuell das schriftliche Quadratwurzelziehen auf dem Stundenplan. Ruth hat auch schon die Grundlagen verinnerlicht. Leider verrechnet sie sich aber noch sehr oft. Um ihre Zwischenschritte überprüfen zu können, bittet sie dich, ein entsprechendes Programm zu implementieren.

Update 31.10.

Einige der häufigsten Zulip-Fragen wurden in der Aufgabenstellung klargestellt und einige Tippfehler behoben: Tippfehler im maximalen input-Wert, Ausgabe der input-Zahl, überflüssiges ' ' im Ausgabetext, Anmerkung zu 0-er Blöcken, Hinweis zu `static` Variablen

Update 2.11.

Vereinheitlichung der Zahlendarstellung mit "." als dezimales Trennzeichen, die auch von Artemis verwendet wird.

Aufgabe

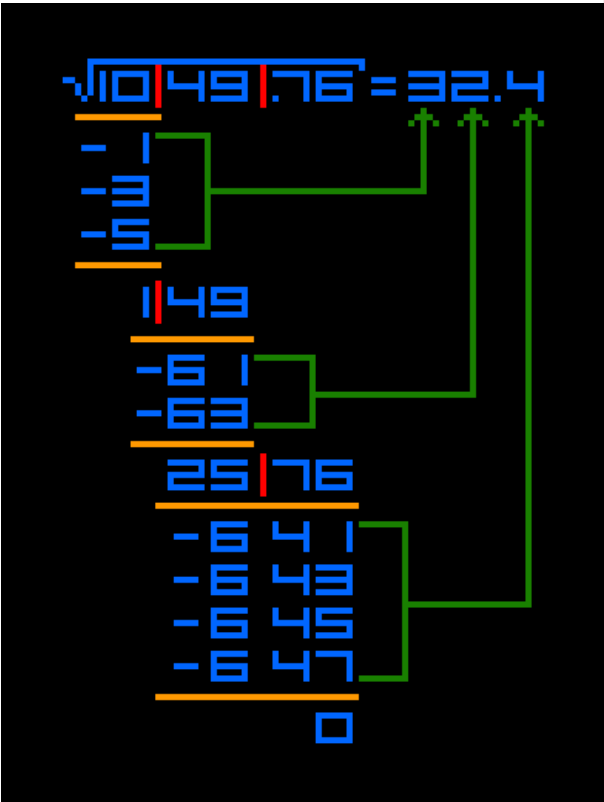
Implementiere die Methode `sqrt`. Diese erhält einen `double n`, dessen Quadratwurzel berechnet werden soll. Dafür soll die Methode des schriftlichen Quadratwurzelziehens verwendet werden, die weiter unten erklärt wird. Außerdem sollen bestimmte Zwischenschritte auf der Konsole ausgegeben werden. Auch dazu findest du weiter unten mehr Informationen. Es gelten folgende Einschränkungen:

- `|n| <= Integer.MAX_VALUE = 2,147,483,647`
- Für die Berechnung sind nur die ersten 4 Nachkommastellen von `n` relevant. Alle weiteren können (außer in der 1. Ausgabezeile) abgeschnitten werden.
- Die Methode soll das exakte Ergebnis auf 2 Nachkommastellen berechnen. D.h., dass nicht gerundet wird, sondern vor der 3. Nachkommastelle "abgeschnitten" wird. (1.347 wird zu 1.34)
- Funktionen aus `java.lang.Math` sind nicht erlaubt

Schriftliches Quadratwurzelziehen

Quadratwurzeln lassen sich wie folgt (schriftlich) berechnen:

- Die Zahl muss in Zifferngruppen der Größe 2 geteilt werden, ausgehend vom Dezimalpunkt (also für den ganzzahligen Anteil nach links, für die Nachkommastellen nach rechts). Bei einer ungeraden Anzahl an Ziffern wird mit einer 0 aufgefüllt. 1.234 wird z.B. 01 | 23 | 40 geteilt, nicht 12 | 34)
- Starte mit der ersten Zifferngruppe als Minuend, mit 1 als Subtrahend. Solange das Ergebnis ≥ 0 ist, berechne die Differenz und erhöhe den Subtrahend um 2
- Die Anzahl der unterschiedlichen Subtrahenden, die in Schritt 2 abgezogen wurden, ist die erste Ziffer des Ergebnisses
- Die übrigen Ziffern des Ergebnisses lassen sich ähnlich berechnen: Nimm den Rest des vorherigen Schritts und hänge die nächste Zifferngruppe an (falls alle Zifferngruppen bereits abgearbeitet wurden, hänge 00 an). Dies ist der neue Minuend. Berechne den neuen Start-Subtrahend indem du eine 1 an das Doppelte des aktuellen Ergebnisses anhängst. Solange das Ergebnis ≥ 0 ist, berechne die Differenz und erhöhe den Subtrahend um 2. Die Anzahl der abgezogenen Subtrahenden ist die nächste Ziffer des Ergebnisses.
- Schritt 4 solange wiederholen, bis alle Ziffern der Eingabezahl abgearbeitet wurden und der Rest nach einem Schritt 0 ergibt oder die gefragte Genauigkeit erreicht wurde. Der Dezimalpunkt



kommt im Ergebnis vor der Ziffer, zu deren Berechnung die erste Zifferngruppe nach dem Komma der Eingabezahl genutzt wurde.

Ruth hat dir auch ein Beispiel vorbereitet (siehe Bild rechts und Animation ganz unten):

In diesem Beispiel soll die Quadratwurzel von 1049.76 berechnet werden. Dafür wird die Zahl zuerst in Gruppen aufgeteilt (1.): 10 | 49 | 76. Im nächsten Schritt (2.) wird von der 10 erst 1 (Zwischenergebnis: 9), dann 3 (Zwischenergebnis: 6) und schließlich 5 (Zwischenergebnis: 1) abgezogen. 7 kann nicht mehr abgezogen werden, da das Ergebnis sonst negativ wäre. Da also 3 Zahlen abgezogen wurden, ist 3 auch die 1. Ziffer des Ergebnisses (3.). Jetzt kann entsprechend weitergerechnet werden: Zuerst wird an den Rest des letzten Schritts (1) die nächste Zifferngruppe angehängt (49). Der nächste Minuend ist also 149. Der erste Subtrahend ergibt sich aus dem doppelten des bisherigen Ergebnis ($2 \cdot 3 = 6$) und einer angehängten 1, also 61. Aus $149 - 61 - 63 = 25$ folgt dann 2 als nächste Ziffer des Ergebnis (aktuell 32) und Rest 25 für den nächsten Schritt. Da nun der Dezimalpunkt erreicht ist, muss er auch im Ergebnis eingefügt werden (32.). Jetzt fehlt noch die letzte Zifferngruppe: neue Zifferngruppe an letzten Rest anhängen -> Minuend 2576. 1 an das doppelte des bisherigen Ergebnis für den neuen Subtrahenden anhängen: $2 \cdot 32 = 64$ => Subtrahend 641. Aus $2576 - 641 - 643 - 645 - 647 = 0$ erhalten wir die neue Ergebnisziffer 4. Außerdem haben wir Rest 0 und keine verbleibende Ziffernfolge. Die Berechnung ist also abgeschlossen und das Ergebnis 32.4 ist exakt.

Ausgabe

Um ihre Rechnungen zu überprüfen, möchte Ruth, dass dein Programm folgende Ausgabe produziert:

- 1. Zeile: "Wurzel aus <Zahl, deren Wurzel berechnet wird (mit allen übergebenen Nachkommastellen, . als Trennzeichen)>"
- 2. Zeile leer
- Für jede Zifferngruppe folgenden Block, wobei x die Anzahl der Subtrahenden ist:
 "<aktueller Minuend>

 -<1. Subtrahend>
 -<...>
 -<x. Subtrahend>

 Rest: <aktueller Rest>
 neue_Ergebnis_Ziffer: <neue Ergebnis Ziffer = x>
 <leere Zeile>"
- Als letzte Zeile: "Ergebnis: <das Ergebnis>"
- Für eine negative Zahl soll nur einzige Zeile ausgegeben werden, nämlich: "Keine negativen_Wurzeln!"

▼ Für das obige Beispiel würde sich also folgende Ausgabe ergeben:

```
Wurzel aus 1049.76

10
-----
-1
-3
-5
-----
Rest: 1
neue Ergebnis Ziffer: 3

149
-----
-61
-63
-----
Rest: 25
neue Ergebnis Ziffer: 2

2576
-----
-641
-643
-645
-647
-----
Rest: 0
neue Ergebnis Ziffer: 4

Ergebnis: 32.4
```

▼ Ein weiteres Beispiel für eine Ausgabe: Wurzel aus 4

```
Wurzel aus 4.0
```

```
4
```

```
-----
```

```
-1
```

```
-3
```

```
-----
```

```
Rest: 0
```

```
neue Ergebnis Ziffer: 2
```

```
Ergebnis: 2.0
```

Ruth hat außerdem noch ein paar Anmerkungen, um dir die Aufgabe zu erleichtern:

- Die Ausgabe der Zahl in der 1. Zeile soll in der standardmäßigen `java`-String-Repräsentation der Zahl entsprechen, also z.b. "1.0E9" für eine Milliarde, "2.0" für 2 und "0.1" für 0.1. Gleiches gilt für die Ausgabe des finalen Ergebnis.
- Es sollen nur Subtrahenden ausgegeben werden, die auch erfolgreich vom Minuenden abgezogen werden können. Wenn es keinen gibt, dann folgen die beiden "-----"-Zeilen direkt aufeinander.
- 0er-Blöcke sind nicht relevant, wenn sie zum Berechnen der ersten oder letzten Ergebnis-Ziffern genutzt werden. Es muss aber immer mindestens ein Block ausgegeben werden. Die Ausgabe für Wurzel aus 0.0004 wäre also die selbe wie für Wurzel aus 4 in obigem Beispiel (außer 1. und letzter Zeile natürlich). Die 0er-Blöcke dürfen aber auch ausgegeben werden, wie im nächsten Beispiel.
- Artemis verwendet einen "." als dezimales Trennzeichen (vgl. Beispielausgaben)

▼ Folgende Ausgabe wäre für die Wurzel aus 4 also auch ok:

```
Wurzel aus 4.0
```

```
4
```

```
-----
```

```
-1
```

```
-3
```

```
-----
```

```
Rest: 0
```

```
neue Ergebnis Ziffer: 2
```

```
0
```

```
-----
```

```
-----
```

```
Rest: 0
```

```
neue Ergebnis Ziffer: 0
```

```
0
```

```
-----
```

```
-----
```

```
Rest: 0
```

```
neue Ergebnis Ziffer: 0
```

```
Ergebnis: 2.0
```

Das Beispiel nochmal Schritt für Schritt animiert:



Exercise details

Release date:	Oct 27, 2022 18:30
Submission due:	Nov 6, 2022 18:00
Complaint due:	Nov 13, 2022 18:00

Every student is allowed to complain once per exercise. In total 1000 complaints are possible in this course. You still have **998** complaints left. 