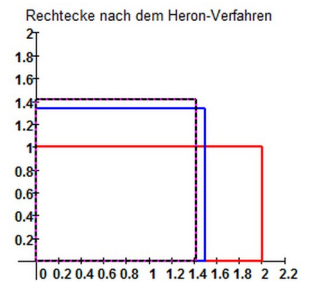


**Information:**

Das Heron-Verfahren zur Bestimmung der Quadratwurzel einer Zahl nutzt ausschließlich die Grundrechenarten Division und Addition, liefert sehr schnell eine recht genaue Lösung und ist benannt nach dem griechischen Mathematiker Heron von Alexandria.

Das Verfahren war in Mesopotamien bereits zur Zeit von Hammurapi I. (ca. 1750 v. Chr.), eines Königs von Babylon, bekannt. Um 100 n. Chr. wurde es von Heron von Alexandria im ersten Buch seines Werkes *Metrica* beschrieben.

(Quelle: Wikipedia)

**Aufgaben:**

- Informieren Sie sich über die Verfahrensweise dieses Algorithmus.
- Stellen Sie am Beispiel der Wurzel aus 20 das Verfahren dar, starten Sie mit den Faktoren 1 und 20:
  - bildlich** durch gezeichnete Rechtecke mit Flächeninhalt 20 (Kästchen), zuletzt (fast) ein Quadrat
  - algebraisch** durch fortgesetztes Prüfen, ob die beiden Faktoren bereits (fast) die gleiche Größe haben, und ggf. Bilden neuer Faktoren u. a. aus dem arithmetischen Mittel beider Faktoren
- Notieren Sie in Form eines Programm-Ablauf-Plans (PAP), eines Struktogramms oder in Pseudo-Code die Arbeitsweise dieses Algorithmus.
- Modellieren Sie z. B. in Form eines Implementationsdiagramms die Klassen ConHeron und Heron. ConHeron enthält ein Heron-Objekt, ruft dessen Heron-Methoden auf und zeigt abschließend auf der Konsole Informationen und das Ergebnis an, bildet also den View (vgl. MVC-Konzept). Heron hat als Attribute `quadrat`, `wurzel`, `genauigkeit`, `schritte` und enthält als Methoden das Heron-Verfahren, bildet also den Controller und das Model (vgl. MVC-Konzept).
- Implementieren Sie anfangs die iterativ (also mit einer Wiederholungsstruktur) arbeitende Heron-Methode `gibWurzelIterativ(double pQuadrat, double pGenau)`, die den ermittelten Wurzelwert im Attribut `wurzel` einträgt und auch zurückgibt.
- Ergänzen Sie in der ConHeron-Methode `main()` den Aufruf dieser Heron-Methode und eine ConHeron-Methode `zeigeInfo()`, die mit Hilfe vorhandener `get`-Methoden des Heron-Objekts einen informative Aussage auf der Konsole anzeigt:  
Als Wurzel der Zahl 20 wurde iterativ nach 5 Schritten der Wert 4.472 ermittelt.
- Implementieren Sie nun das Heron-Verfahren auch rekursiv, also mit zwei weiteren Heron-Methoden: `gibWurzelRekursiv()` und die sich selbst aufrufende Methode `wurzelRekursiv(double pFaktor)`, die den derzeitigen Faktor annimmt und (ggf.) einen genaueren Faktor zurückliefert.  
Hinweis: Es genügt der Heron-Methode `wurzelRekursiv()`, nur einen Faktor als Parameter zu übernehmen, da ihr sowohl die Ausgangszahl als auch die Genauigkeit als Attribute zugänglich sind. Damit kann sowohl der zweite Faktor ermittelt als auch die Genauigkeit geprüft werden.  
Zusätzlich kann auch einmal eine rekursive Lösung versucht werden, die nicht auf diese Klassen-Attribute zurückgreift.
- Lösen Sie auch das Problem, dass bei der Informations-Anzeige mit `zeigeInfo()` nur die gesicherte Anzahl Nachkommastellen des Wurzelwerts (Faktors) angezeigt werden sollen. Wird als Genauigkeit 0,01 übergeben, dann ist nur eine (die erste) Nachkommastelle gesichert!

**Hilfreiche Links:**

[1] Wikipedia: [[Heron von Alexandria](#)] und [[Heron-Verfahren](#)]