

Pi näherungsweise berechnen

1. Erstelle ein Programm das die Zahl π näherungsweise nach einer der vier Methoden berechnet
2. **Freiwillig:**
Verwende dazu **BigDecimal** Zahlen
Lies das Tutorial hier durch <https://www.journaldev.com/16409/java-bigdecimal>
3. Die Ausgabe erfolgt nach freiem Belieben, soll aber auch die Dauer der Berechnung angeben.

Summe nach Viète

nach François Viète 1593:

$$\frac{2}{\pi} = \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \dots$$

Summe nach Leibniz

nach Gottfried Wilhelm Leibniz 1674

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \frac{4}{13} - \dots$$

Summe nach Wallis

nach John Wallis 1655

$$\frac{\pi}{2} = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \cdot \frac{8}{9} \dots$$

Summe nach Nilakantha

nach Nilakantha Somayaji im 15

Jahrhundert

$$\pi = 3 + \frac{4}{2 \cdot 3 \cdot 4} - \frac{4}{4 \cdot 5 \cdot 6} + \frac{4}{6 \cdot 7 \cdot 8} - \frac{4}{8 \cdot 9 \cdot 10} + \dots$$

Hinweise:

BigDecimal 2.18 ist unscaled 218 und hat ein scale von 2.

Beim Dividieren (u.a.) gibt man die precision an, also auf wie viele Stellen genau gerechnet werden soll. Im Beispiel wird auf 15 genau gerechnet.

Zusätzlich muss man noch eine **Rundungsmethode** angeben.

```
BigDecimal divide = z1.divide(z2, 15, RoundingMode.HALF_UP);
```

