

**Name:**

**Matrikelnummer:**

**Bearbeitungszeit:** 60 min.

Gegeben ist das Interface `nullstellen.gegeben.Funktion` mit der Methode

```
public double f(double x);
```

das eine mathematische Funktion  $f : \mathbb{R} \rightarrow \mathbb{R}$  darstellt.

Schreiben Sie die `public` Klasse `nullstellen.Suche` mit einem parameterlosen Konstruktor und der Methode

```
public double findeNullstelle(Funktion fun, double a, double b, double eps),
```

die näherungsweise eine Nullstelle der Funktion `fun.f` im Intervall  $[a, b]$  mit der Genauigkeit `eps` zurückgibt. Ein zurückgegebenen Wert `x` gilt dann als korrekt, wenn es eine Nullstelle `x0` der Funktion `fun.f` gibt mit  $|x-x_0| < \text{eps}$ .

Die Methode `findeNullstelle(fun, a, b, eps)` muss nur dann ein korrektes Ergebnis liefern, wenn `fun.f(a) >= 0` und `fun.f(b) <= 0` ist, und wenn die Funktion `fun.f` stetig ist. Sie können also annehmen, dass diese Voraussetzungen beim Methodenaufruf erfüllt sind.

*Hinweis:* Wenn für eine stetige Funktion  $f$  gilt, dass  $f(a) \geq 0$  und  $f(b) \leq 0$ , dann besitzt die Funktion  $f$  zumindest eine Nullstelle im Intervall  $[a, b]$ . Eine Näherung der Nullstelle kann gefunden werden, indem das Intervall, in dem die Nullstelle liegen muss, sukzessive verkleinert wird.

Die Methode `findeNullstelle()` soll auch für kleine Werte von `eps` (z.B. `eps=1e-9`) rasch eine Nullstelle finden.

*Beispiel:* In der beiliegenden Klasse `Cosinus` ist die Cosinusfunktion implementiert. Die Cosinusfunktion hat Nullstellen bei  $\pi/2 \approx 1.5708$ ,  $3\pi/2 \approx 4.7124$ ,  $5\pi/2 \approx 7.8540, \dots$  Korrekte Ergebnisse des Aufrufs `findeNullstelle(cos, 0, 10, 0.001)` sind zum Beispiel 1.5712, 4.7121 oder 7.8532.

