



# Dokumentation der SuM-Werkzeuge

[Uhr](#)

[Rechner](#)

[Textwerkzeug](#)



# Dokumentation der Klasse "Uhr"

Klassenname Uhr

Oberklasse

Beschreibung Die Klasse Uhr stellt eine Uhr zur Verfügung, die zusätzlich benutzt werden kann, um ein Programm eine bestimmte Zeit anzuhalten (Pause) sowie Stoppuhr-Funktionen bietet.

Bezugsklassen

Protokoll

**Auftrag** init

*nachher* Die Uhr ist initialisiert.

**Anfrage** datum : Zeichenkette

Diese Anfrage liefert das aktuelle Datum in der Form "Tag.Monat.Jahr".

**Anfrage** zeit : Zeichenkette

*nachher* Diese Anfrage liefert die aktuelle Uhrzeit in der Form "Stunde:Minute:Sekunde".

**Anfrage** tag : Ganzzahl

*nachher* Diese Anfrage liefert den aktuellen Tag im Monat als Ganzzahl.

**Anfrage** monat : Ganzzahl

*nachher* Diese Anfrage liefert den aktuellen Monat als Ganzzahl.

**Anfrage** jahr : Ganzzahl

*nachher* Diese Anfrage liefert die aktuelle Jahreszahl (vierstellig) als Ganzzahl.

**Anfrage** stunde : Ganzzahl

*nachher* Diese Anfrage liefert die aktuelle Stunde als Ganzzahl.

**Anfrage** minute : Ganzzahl

*nachher* Diese Anfrage liefert die aktuelle Minute als Ganzzahl.

**Anfrage** sekunde : Ganzzahl

*nachher* Diese Anfrage liefert die aktuelle Sekunde als Ganzzahl.

**Auftrag** starte

*nachher* Die Stoppuhr wurde gestartet.

**Auftrag** stoppe

*nachher* Die Stoppuhr wurde angehalten.

**Auftrag** gestoppteZeit: Zahl

*nachher* Diese Anfrage liefert die gestoppte Zeit in Sekunden auf Tausendstelsekunden genau.

**Auftrag** warte(pMillisekunden: Ganzzahl)

*nachher* Das aktuelle Programm wurde für eine bestimmte Zahl von Millisekunden angehalten.

**Auftrag** gibFrei

*nachher* Die Uhr steht nicht mehr zur Verfügung.



# Dokumentation der Klasse "Rechner"

Klassenname	Rechner
Oberklasse	
Beschreibung	Ein Rechner ist ein Hilfsmittel zur Berechnung mathematischer Funktionen. Daneben kann er auch Zufallszahlen erzeugen.
Bezugsklassen	
Protokoll	<p><b>Auftrag</b> init <i>nachher</i> Der Rechner ist betriebsbereit.</p> <p><b>Anfrage</b> zufallszahl : Zahl <i>nachher</i> Diese Anfrage liefert eine Zufallszahl von (einschließlich) 0 bis (ausschließlich) 1.</p> <p><b>Anfrage</b> ganzeZufallszahl (pVon, pBis : GanzeZahl) : GanzeZahl <i>nachher</i> Diese Anfrage liefert eine ganze Zufallszahl aus dem angegebenen Bereich einschließlich der Grenzen.</p> <p><b>Anfrage</b> betrag (pZahl : Zahl) : Zahl <i>nachher</i> Die Anfrage liefert den Betrag der angegebenen Zahl.</p> <p><b>Anfrage</b> quadrat (pZahl : Zahl) : Zahl <i>nachher</i> Die Anfrage liefert das Quadrat der angegebenen Zahl.</p> <p><b>Anfrage</b> wurzel (pZahl : Zahl) : Zahl <i>nachher</i> Die Anfrage liefert die Wurzel der angegebenen Zahl.</p> <p><b>Anfrage</b> potenz (pBasis, pExponent : Zahl) : Zahl <i>nachher</i> Die Anfrage liefert die Potenz mit der angegebenen Basis und dem Exponenten.</p> <p><b>Anfrage</b> rund (pZahl : Zahl) : GanzeZahl <i>nachher</i> Die Anfrage rundet die angegebene Zahl.</p> <p><b>Anfrage</b> ganzerAnteil (pZahl : Zahl): GanzeZahl <i>nachher</i> Die Anfrage liefert den ganzzahligen Anteil der angegebenen Zahl.</p> <p><b>Auftrag</b> gibFrei <i>nachher</i> Der Rechner steht nun nicht mehr zur Verfügung.</p>
Hinweis	Die Anfragen „betrag“, „quadrat“ und „potenz“ sind kontextsensitiv, d.h. bei einem ganzzahligen Parameter liefern sie auch ein ganzzahliges Ergebnis.



# Dokumentation der Klasse „Textwerkzeug“

Klassenname	Textwerkzeug
Oberklasse	
Beschreibung	Ein Textwerkzeug ist ein Hilfsmittel zur Bearbeitung und Analyse von Zeichenketten. Zur Trennung von Wörtern gibt es ein Trennungszeichen. Standardmäßig ist das ein Leerzeichen.
Bezugsklassen	
Protokoll	<p><b>Auftrag</b> <code>init</code> <i>nachher</i> Das Textwerkzeug ist betriebsbereit.</p> <p><b>Anfrage</b> <code>länge(pText : Zeichenkette) : GanzeZahl</code> <i>nachher</i> Diese Anfrage liefert die Anzahl der Zeichen in dem angegebenen Text.</p> <p><b>Anfrage</b> <code>zeichenAn(pText:Zeichenkette; pStelle:GanzeZahl) : Zeichen</code> <i>nachher</i> Diese Anfrage liefert das Zeichen an der Stelle in dem angegebenen Text.</p> <p><b>Auftrag</b> <code>setzeTrennungszeichen(pTrennungszeichenkette : Zeichenkette)</code> <i>nachher</i> Die Trennungszeichenkette für die nachfolgenden Anfrage ist gesetzt.</p> <p><b>Anfrage</b> <code>wortAn(pText:Zeichenkette; pWortnummer:GanzeZahl) : Zeichenkette</code> <i>nachher</i> Diese Anfrage liefert das Wort mit der angegebenen Wortnummer in dem angegebenen Text. Die Wörter sind dabei durch die vorher definierte Trennungszeichenkette voneinander getrennt.</p> <p><b>Anfrage</b> <code>teilZeichenkette(pText:Zeichenkette; pVon, pBis:GanzeZahl) : Zeichenkette</code> <i>nachher</i> Die Anfrage liefert den den angegebenen Teil des Textes. Die Grenzen gehören dabei dazu.</p> <p><b>Anfrage</b> <code>positionVon(pText, pSuchtext:Zeichenkette) : GanzeZahl</code> <i>nachher</i> Die Anfrage liefert den Anfang des in dem übergebenen Textes enthaltenen Suchtextes. Ist der Suchtext nicht enthalten, wird 0 zurückgegeben.</p> <p><b>Anfrage</b> <code>textOhne(pText:Zeichenkette; pVon, pBis:GanzeZahl) : Zeichenkette</code> <i>nachher</i> Die Anfrage liefert den Text ohne die in den angegebenen Grenzen enthaltenen Zeichen.</p> <p><b>Anfrage</b> <code>textMit(pText, pNeu : Zeichenkette; pStelle : GanzeZahl) : Zeichenkette</code> <i>nachher</i> Die Anfrage liefert einen Text zurück, der durch Einfügen des Teil "pNeu" an der angegebenen Stelle entsteht.</p> <p><b>Anfrage</b> <code>kleinschrift(pText:Zeichenkette):Zeichenkette</code> <i>nachher</i> Die Anfrage liefert den Text umgewandelt in Kleinschrift zurück.</p> <p><b>Anfrage</b> <code>grossschrift(pText:Zeichenkette):Zeichenkette</code> <i>nachher</i> Die Anfrage liefert den Text umgewandelt in Großschrift zurück.</p> <p><b>Anfrage</b> <code>istGleich(pText1, pText2 : Zeichenkette):Wahrheitswert</code> <i>nachher</i> Die Anfrage überprüft, ob die beiden Texte gleich sind.</p>



**Anfrage** istKleiner(pText1, pText2 : Zeichenkette):Wahrheitswert

*nachher* Die Anfrage überprüft, ob der erste Text (alphabetisch angeordnet) kleiner ist als der zweite Text.

**Anfrage** istGroesser(pText1, pText2 : Zeichenkette):Wahrheitswert

*nachher* Die Anfrage überprüft, ob der erste Text (alphabetisch angeordnet) größer ist als der zweite Text.

**Anfrage** istZahl(pText:Zeichenkette):Wahrheitswert

*nachher* Die Anfrage überprüft, ob der angegebene Text eine Zahl ist.

**Anfrage** istGanzeZahl(pText:Zeichenkette):Wahrheitswert

*nachher* Die Anfrage überprüft, ob der angegebene Text eine ganze Zahl ist.

**Anfrage** alsZahl(pText:Zeichenkette):Zahl

*vorher* Der angegebene Text ist eine Zahl.

*nachher* Die Anfrage konvertiert den angegebenen Text in eine Zahl.

**Anfrage** alsGanzeZahl(pText:Zeichenkette):GanzeZahl

*vorher* Der angegebene Text ist eine ganze Zahl.

*nachher* Die Anfrage konvertiert den angegebenen Text in eine ganze Zahl.

**Anfrage** alsText(pZahl:Zahl / GanzeZahl) : Zeichenkette

*nachher* Die Anfrage konvertiert die angegebene Zahl in eine Zeichenkette.

**Anfrage** verkettung(pText1, pText2 : Zeichenkette):Zeichenkette

*nachher* Die Anfrage liefert eine Zeichenkette zurück, die aus den dem ersten Text, verknüpft mit dem zweiten Text besteht.

**Auftrag** gibFrei

*nachher* Das Textwerkzeug steht nun nicht mehr zur Verfügung.