

KLASSENBSCHREIBUNGEN

OPERATIONS- UND ATTRIBUTBSCHREIBUNG FÜR DEN FILTER RECHNER

DV Projekt SoSe 2020 – Projektgruppe G

Elias Kopf, Jens Fuchs, Marc Schrenk, Marvin Merk, Thomas Schmid, Pascal Guttman

Fachklassen	Programmierer
Filter, RC_TP, RC_HP, RL_TP, RL_HP	Marc Schrenk, Pascal Guttman
Bauteil, R_ideal, C_ideal, L_ideal	Marvin Merk, Jens Fuchs
LMath, CMath, E_Reihe, Rechner	Elias Kopf, Thomas Schmid

Wahlaufgaben	Gewählt von
GUI	Marc Schrenk, Pascal Guttman
Abschluss Tests	Marvin Merk, Jens Fuchs
Koordinator	Elias Kopf
Diagramme	Thomas Schmid

Die Klassenbeschreibung sollte am besten in Kombination mit dem Klassendiagramm zum Programmieren verwendet werden. Um die Kompatibilität sicherzustellen sollte man sich bitte an folgende Konventionen halten:

1. Die Reihenfolge der Parameter wie hier angegeben beibehalten. Sollte ebenfalls mit dem Klassendiagramm übereinstimmen.
2. Die Groß- und Kleinschreibung beachten.
 - a. Methodennamen beginnen mit Kleinbuchstaben.
 - b. Attributnamen beginnen mit Kleinbuchstaben.
 - c. Klassennamen beginnen mit Großbuchstaben.
3. Umlaute („ä“, „ü“, „ö“) werden jeweils mit „ae“, „ue“ oder „oe“ geschrieben.
4. Prozente werden als „Null Komma“ angegeben. Also zB: 5% = 0.05
5. Zu Attributen die passenden set und get Methoden schreiben und nutzen:
 - a. „hole“ oder „setze“ klein schreiben und dann den Attributnamen mit großem Anfangsbuchstaben. zB: holeAttributname() und setzeAttributname()
 - b. Auch eigene Attribute einer Klasse möglichst mit „Set Methode“ setzen. Also - explizit im Konstruktor - die „Set Methode“ verwenden, um Attribute zu initialisieren. (egal ob private, protected oder public). Es sei denn es gibt andere Gründe für ein anderes initialisieren.
6. Wo „null“ steht ist der Zahlenwert gemeint. Ein Verweis auf einen fehlenden Wert wird durch „NULL“ gekennzeichnet.

Viel Spaß beim Programmieren Jungs! 😊

KLASSENVERZEICHNIS

Klassenverzeichnis	1
LMath	2
CMath.....	3
Bauteil	4
R_ideal	6
L_ideal.....	8
C_ideal.....	9
Filter	10
RC_TP	14
RC_HP.....	15
RL_TP.....	16
RL_HP	17
E_Reihe	18
Rechner	21

LMATH

KLASSENEIGENSCHAFTEN

Name: LMath
Eigenschaften: public, final

ATTRIBUTBESCHREIBUNG

-

OPERATIONSBESCHREIBUNG

KONSTRUKTOR

Name: LMath
Eigenschaften: private
Parameter: void
Rückgabe: -
Beschreibung: Keine Funktion. LMath wird nicht instanziiert.

X

Name: x
Eigenschaften: public, static
Parameter: double Frequenz, double Induktivität
Rückgabe: double Blindwiderstand
Beschreibung: Die Methode erhält eine Frequenz und eine Induktivität. Daraus wird der Blindwiderstand berechnet und als double zurückgegeben.

$$X_L = 2 * pi * f * L$$

F

Name: f
Eigenschaften: public, static
Parameter: double Blindwiderstand, double Induktivität
Rückgabe: double Frequenz
Beschreibung: Die Methode erhält einen Blindwiderstand und eine Induktivität. Daraus wird die Frequenz berechnet und als double zurückgegeben.

$$X_L = 2 * pi * f * L$$

L

Name: l
Eigenschaften: public, static
Parameter: double Blindwiderstand, double Frequenz
Rückgabe: double Induktivität
Beschreibung: Die Methode erhält Blindwiderstand und eine Frequenz. Daraus wird die Induktivität berechnet und als double zurückgegeben.

$$X_L = 2 * pi * f * L$$

CMATH

KLASSENEIGENSCHAFTEN

Name: CMath
Eigenschaften: public, final

ATTRIBUTBESCHREIBUNG

-

OPERATIONSBESCHREIBUNG

KONSTRUKTOR

Name: CMath
Eigenschaften: private
Parameter: void
Rückgabe: -
Beschreibung: Keine Funktion. CMath wird nicht instanziiert.

X

Name: x
Eigenschaften: public, static
Parameter: double Frequenz, double Kapazität
Rückgabe: double Blindwiderstand
Beschreibung: Die Methode erhält eine Frequenz und eine Kapazität. Daraus wird der Blindwiderstand berechnet und als double zurückgegeben.

$$X_c = \frac{1}{2 * pi * f * C}$$

F

Name: f
Eigenschaften: public, static
Parameter: double Blindwiderstand, double Kapazität
Rückgabe: double Frequenz
Beschreibung: Die Methode erhält einen Blindwiderstand und eine Kapazität. Daraus wird die Frequenz berechnet und als double zurückgegeben.

$$X_c = \frac{1}{2 * pi * f * C}$$

C

Name: c
Eigenschaften: public, static
Parameter: double Blindwiderstand, double Frequenz
Rückgabe: double Kapazität
Beschreibung: Die Methode erhält Blindwiderstand und eine Frequenz. Daraus wird die Kapazität berechnet und als double zurückgegeben.

$$X_c = \frac{1}{2 * pi * f * C}$$

BAUTEIL**KLASSENEIGENSCHAFTEN**

Name: Bauteil
Eigenschaften: abstract

ATTRIBUTBESCHREIBUNG**WERT**

Name: wert
Datentyp: double
Eigenschaften: protected
Set Methode: Nur positive Werte und 0. Namenskonvention beachten.
Get Methode: „holeWert()“ als public anbieten.
Beschreibung: Der Wert des jeweiligen Bauteils in SI-Einheiten.

TOLERANZ

Name: toleranz
Datentyp: double
Eigenschaften: protected
Set Methode: Nur positive Werte und 0. Namenskonvention beachten.
Get Methode: „holeToleranz()“ als public anbieten.
Beschreibung: Die Toleranz des jeweiligen Bauteils als Gleitkommazahl.

OPERATIONSBESCHREIBUNG**IMPEDANZ**

Name: impedanz
Eigenschaften: public, abstract
Parameter: double Frequenz
Rückgabe: double Impedanz
Beschreibung: Methode muss von jeder Kindklasse überschrieben werden (abstract). Funktion erhält die Frequenz und gibt die Impedanz des Bauteils bei dieser zurück.

IMPEDANZMAX

Name: impedanzMax
Eigenschaften: public, abstract
Parameter: double Frequenz
Rückgabe: double maximale Impedanz
Beschreibung: Methode muss von jeder Kindklasse überschrieben werden (abstract). Funktion erhält die Frequenz und gibt die Impedanz des Bauteils bei dieser zurück.

IMPEDANZMIN

Name: impedanzMin
Eigenschaften: public, abstract
Parameter: double Frequenz
Rückgabe: double minimale Impedanz
Beschreibung: Methode muss von jeder Kindklasse überschrieben werden (abstract). Funktion erhält die Frequenz und gibt die Impedanz des Bauteils bei dieser zurück.

HOLEWERT

Name: holeWert
Eigenschaften: public
Parameter: void
Rückgabe: double Wert
Beschreibung: Methode gibt den Wert des Bauteils zurück. Die Methode muss nicht in Kindklassen überschrieben werden.

HOLEWERTMAX

Name: holeWertMax
Eigenschaften: public
Parameter: void
Rückgabe: double maximaler Wert
Beschreibung: Methode gibt den maximalen Wert in der Toleranz des Bauteils zurück. Die Methode muss nicht in Kindklassen überschrieben werden.

$$wert_{max} = this.wert * (1 + this.toleranz)$$

HOLEWERTMIN

Name: holeWertMin
Eigenschaften: public
Parameter: void
Rückgabe: double minimaler Wert
Beschreibung: Methode gibt den minimalen Wert in der Toleranz des Bauteils zurück. Die Methode muss nicht in Kindklassen überschrieben werden.

$$wert_{min} = this.wert * (1 - this.toleranz)$$

R_IDEAL

KLASSENEIGENSCHAFTEN

Name: R_ideal
 Eigenschaften: public, extends Bauteil

ATTRIBUTBESCHREIBUNG

-

OPERATIONSBESCHREIBUNG

KONSTRUKTOR

Name: R_ideal
 Eigenschaften: public
 Parameter: double Widerstandwert, double Toleranz
 Rückgabe: -
 Beschreibung: Instanziert die Klasse R_ideal.
 Nutzt die Set Methode des Attributs „wert“ aus der Klasse „Bauteil“, um dieses mit dem Widerstandswert zu initialisieren.
 Nutzt die Set Methode des Attributs „toleranz“ aus der Klasse „Bauteil“, um dieses mit der Toleranz zu initialisieren.

IMPEDANZ

Name: impedanz
 Eigenschaften: public
 Parameter: double Frequenz
 Rückgabe: double Impedanz
 Beschreibung: Überschreiben der Funktion aus „Bauteil“.
 Rückgabe des Attributs „wert“. Unabhängig vom übergebenen Parameter.

$$R = wert$$

IMPEDANZMAX

Name: impedanzMax
 Eigenschaften: public
 Parameter: double Frequenz
 Rückgabe: double maximale Impedanz
 Beschreibung: Überschreiben der Funktion aus „Bauteil“.
 Rückgabe der maximal möglichen Impedanz.
 Unabhängig vom übergebenen Parameter.

$$R_{max} = wert * (1 + toleranz)$$

IMPEDANZMIN

Name: impedanzMin
 Eigenschaften: public
 Parameter: double Frequenz
 Rückgabe: double minimale Impedanz
 Beschreibung: Überschreiben der Funktion aus „Bauteil“.
 Rückgabe der minimal möglichen Impedanz.
 Unabhängig vom übergebenen Parameter.

$$R_{min} = wert * (1 - toleranz)$$

IMPEDANZ

Name: impedanz
Eigenschaften: public
Parameter: void
Rückgabe: double Impedanz
Beschreibung: Rückgabe der Impedanz, der aus dem Aufruf der eigenen Methode „this.impedanz(0.0)“ erfolgt.

IMPEDANZMAX

Name: impedanzMax
Eigenschaften: public
Parameter: void
Rückgabe: double maximale Impedanz
Beschreibung: Rückgabe der Impedanz, der aus dem Aufruf der eigenen Methode „this.impedanzMax(0.0)“ erfolgt.

IMPEDANZMIN

Name: impedanzMin
Eigenschaften: public
Parameter: void
Rückgabe: double minimale Impedanz
Beschreibung: Rückgabe der Impedanz, der aus dem Aufruf der eigenen Methode „this.impedanzMin(0.0)“ erfolgt.

L_IDEAL

KLASSENEIGENSCHAFTEN

Name: L_ideal
Eigenschaften: public, extends Bauteil

ATTRIBUTBESCHREIBUNG

-

OPERATIONSBESCHREIBUNG

KONSTRUKTOR

Name: L_ideal
Eigenschaften: public
Parameter: double Induktivität, double Toleranz
Rückgabe: -
Beschreibung: Instanziert die Klasse L_ideal.
Nutzt die Set Methode des Attributs „wert“ aus der Klasse „Bauteil“, um dieses mit der Induktivität zu initialisieren.
Nutzt die Set Methode des Attributs „toleranz“ aus der Klasse „Bauteil“, um dieses mit der Toleranz zu initialisieren.

IMPEDANZ

Name: impedanz
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double Impedanz
Beschreibung: Überschreiben der Funktion aus „Bauteil“.
Rückgabe der Impedanz der Spule bei einer Frequenz, die als Parameter übergeben wurde. Zum Berechnen der Impedanz die Methode „LMath.x(f, this.wert)“ nutzen.

IMPEDANZMAX

Name: impedanzMax
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double maximale Impedanz
Beschreibung: Überschreiben der Funktion aus „Bauteil“.
Rückgabe der maximalen Impedanz der Spule bei einer Frequenz, die als Parameter übergeben wurde. Zum Berechnen der Impedanz die Methode „LMath.x(f, this.wertMax())“ nutzen.

IMPEDANZMIN

Name: impedanzMin
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double minimale Impedanz
Beschreibung: Überschreiben der Funktion aus „Bauteil“.
Rückgabe der minimalen Impedanz der Spule bei einer Frequenz, die als Parameter übergeben wurde. Zum Berechnen der Impedanz die Methode „LMath.x(f, this.wertMin())“ nutzen.

C_IDEAL

KLASSENEIGENSCHAFTEN

Name: C_ideal
Eigenschaften: public, extends Bauteil

ATTRIBUTBESCHREIBUNG

-

OPERATIONSBESCHREIBUNG

KONSTRUKTOR

Name: C_ideal
Eigenschaften: public
Parameter: double Kapazität, double Toleranz
Rückgabe: -
Beschreibung: Instanziert die Klasse C_ideal.
Nutzt die Set Methode des Attributs „wert“ aus der Klasse „Bauteil“, um dieses mit der Kapazität zu initialisieren.
Nutzt die Set Methode des Attributs „toleranz“ aus der Klasse „Bauteil“, um dieses mit der Toleranz zu initialisieren.

IMPEDANZ

Name: impedanz
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double Impedanz
Beschreibung: Überschreiben der Funktion aus „Bauteil“.
Rückgabe der Impedanz des Kondensators bei einer Frequenz, die als Parameter übergeben wurde. Zum Berechnen der Impedanz die Methode „LMath.x(f, this.wert)“ nutzen.

IMPEDANZMAX

Name: impedanzMax
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double maximale Impedanz
Beschreibung: Überschreiben der Funktion aus „Bauteil“.
Rückgabe der maximalen Impedanz des Kondensators bei einer Frequenz, die als Parameter übergeben wurde. Zum Berechnen der Impedanz die Methode „LMath.x(f, this.wertMin())“ nutzen.

IMPEDANZMIN

Name: impedanzMin
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double minimale Impedanz
Beschreibung: Überschreiben der Funktion aus „Bauteil“.
Rückgabe der minimalen Impedanz des Kondensators bei einer Frequenz, die als Parameter übergeben wurde. Zum Berechnen der Impedanz die Methode „LMath.x(f, this.wertMax())“ nutzen.

FILTER**KLASSENEIGENSCHAFTEN**

Name: Filter
Eigenschaften: public

ATTRIBUTBESCHREIBUNG**BAUTEIL1**

Name: bauteil1
Datentyp: Bauteil
Eigenschaften: protected
Set Methode: setzeBauteil1 public anbieten
Get Methode: holeBauteil1 public anbieten
Beschreibung: „bauteil1“ repräsentiert das Bauteil, welches sich im Signalpfad des Filters befindet. Daraus folgt: Steigt die Impedanz von „bauteil1“ so steigt die Dämpfung und vice versa.

BAUTEIL2

Name: bauteil2
Datentyp: Bauteil
Eigenschaften: protected
Set Methode: setzeBauteil2 public anbieten
Get Methode: holeBauteil2 public anbieten
Beschreibung: „bauteil2“ repräsentiert das Bauteil, welches das Ausgangssignal mit Masse verbindet. Daraus folgt: Steigt die Impedanz von „bauteil2“ so fällt die Dämpfung und vice versa.

OPERATIONSBESCHREIBUNG**KONSTRUKTOR**

Name: Filter
Eigenschaften: public
Parameter: Bauteil Bauteil1, Bauteil Bauteil2
Rückgabe: -
Beschreibung: Instanziert die Klasse Filter.
Nutzt die Set Methoden der Attribute um die Bauteile zu setzen.

GRENZFREQUENZ

Name: grenzfrequenz
Eigenschaften: public
Parameter: void
Rückgabe: double Grenzfrequenz
Beschreibung: Die Methode gibt die nominelle Grenzfrequenz des Filters zurück.
Vorgehen zur Berechnung:

1. Das Bauteil des Typ „R_ideal“ mit dem Keyword „instanceof“ finden.
2. Impedanz des Widerstands lesen.
3. Das andere Bauteil nimmt und dessen Wert lesen.
4. Bestimmen ob das andere Bauteil ein „C_ideal“ oder „L_ideal“ ist.

- a. Wenn Typ gleich „C_ideal“ ist, die Methode „CMath.f(Widerstandwert, Kondensatorwert)“ nutzen, um Grenzfrequenz zu bestimmen.
- b. Wenn Typ gleich „L_ideal“ ist, die Methode „LMath.f(Widerstandwert, Kondensatorwert)“ nutzen, um Grenzfrequenz zu bestimmen.

GRENZFREQUENZMAX

Name: grenzfrequenzMax
Eigenschaften: public
Parameter: void
Rückgabe: double maximale Grenzfrequenz
Beschreibung: Die Methode gibt die maximale Grenzfrequenz des Filters in der Toleranz zurück.
Vorgehen zur Berechnung:

1. Vorgehen generell wie bei der Methode „grenzfrequenz()“. Mit der Ausnahme, dass die Prozedur 4-mal ausgeführt wird und die maximal und minimal Werte der beiden Bauteile in jeder Kombination einsetzt, statt der nominellen Werte.
2. Der größte der 4 so errechneten Werte wird zurückgegeben.

GRENZFREQUENZMIN

Name: grenzfrequenzMin
Eigenschaften: public
Parameter: void
Rückgabe: double minimale Grenzfrequenz
Beschreibung: Die Methode gibt die minimale Grenzfrequenz des Filters in der Toleranz zurück.
Vorgehen zur Berechnung:

1. Vorgehen generell wie bei der Methode „grenzfrequenz()“. Mit der Ausnahme, dass die Prozedur 4-mal ausgeführt wird und die maximal und minimal Werte der beiden Bauteile in jeder Kombination einsetzt, statt der nominellen Werte.
2. Der kleinste der 4 so errechneten Werte wird zurückgegeben.

IMPEDANZ

Name: impedanz
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double Eingangsimpedanz
Beschreibung: Die Methode gibt die nominelle Eingangsimpedanz des Filters bei der übergebenen Frequenz zurück.

$$Z_{ein} = \sqrt{Z_{bauteil1}^2 + Z_{bauteil2}^2}$$

IMPEDANZMAX

Name: impedanzMax
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double maximale Eingangsimpedanz
Beschreibung: Die Methode gibt die maximale Eingangsimpedanz des Filters bei der übergebenen Frequenz zurück.

$$Z_{einMAX} = \sqrt{Z_{bauteil1MA}^2 + Z_{bauteil2MA}^2}$$

IMPEDANZMIN

Name: impedanzMin
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double minimale Eingangsimpedanz
Beschreibung: Die Methode gibt die minimale Eingangsimpedanz des Filters bei der übergebenen Frequenz zurück.

$$Z_{einMIN} = \sqrt{Z_{bauteil1MIN}^2 + Z_{bauteil2MIN}^2}$$

DAEMPfung

Name: daempfung
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double logarithmisches Dämpfungsmaß
Beschreibung: Berechnet das nominelle logarithmische Dämpfungsmaß „a“ des Filters bei der übergebenen Frequenz.

$$a = 20 * \lg(D)$$
$$D = \frac{U_e}{U_a} = \frac{Z_1 + Z_2}{Z_2}$$

DAEMPfungMAX

Name: daempfungMax
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double maximales logarithmisches Dämpfungsmaß
Beschreibung: Berechnet das maximale logarithmische Dämpfungsmaß „a“ des Filters bei der übergebenen Frequenz.

$$a_{max} = 20 * \lg(D_{max})$$
$$D_{max} = \frac{U_{e_{max}}}{U_{a_{min}}} = \frac{Z_{1_{max}} + Z_{2_{min}}}{Z_{2_{min}}}$$

DAEMPfungMIN

Name: daempfungMin
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double minimales logarithmisches Dämpfungsmaß
Beschreibung: Berechnet das minimale logarithmische Dämpfungsmaß „a“ des Filters bei der übergebenen Frequenz.

$$a_{min} = 20 * \lg(D_{min})$$
$$D_{min} = \frac{U_{e_{min}}}{U_{a_{max}}} = \frac{Z_{1_{min}} + Z_{2_{max}}}{Z_{2_{max}}}$$

PHASENVERSCHIEBUNG

Name: phasenverschiebung
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double Phasenverschiebung des Ausgangs- zum Eingangssignal
Beschreibung: Berechnet die Phasenverschiebung der Ausgangsspannung gegenüber der Eingangsspannung in Grad, bei der gegebenen Frequenz.

$$\varphi_{TP} = - \operatorname{atan} \left(\frac{Z_1}{Z_2} \right)$$
$$\varphi_{HP} = + \operatorname{atan} \left(\frac{Z_1}{Z_2} \right)$$

PHASENVERSCHIEBUNGMAX

Name: phasenverschiebungMax
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double maximale Phasenverschiebung des Ausgangs- zum Eingangssignal
Beschreibung: Berechnet die maximale (positivste) Phasenverschiebung der Ausgangsspannung gegenüber der Eingangsspannung in Grad, bei der gegebenen Frequenz.

$$\varphi_{TP \text{ Max}} = - \operatorname{atan} \left(\frac{Z_{1 \text{ Max}}}{Z_{2 \text{ Min}}} \right)$$
$$\varphi_{HP \text{ Max}} = + \operatorname{atan} \left(\frac{Z_{1 \text{ Max}}}{Z_{2 \text{ Min}}} \right)$$

PHASENVERSCHIEBUNGMIN

Name: phasenverschiebungMin
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double minimale Phasenverschiebung des Ausgangs- zum Eingangssignal
Beschreibung: Berechnet die minimale (negativste) Phasenverschiebung der Ausgangsspannung gegenüber der Eingangsspannung in Grad, bei der gegebenen Frequenz.

$$\varphi_{TP \text{ Min}} = - \operatorname{atan} \left(\frac{Z_{1 \text{ Min}}}{Z_{2 \text{ Max}}} \right)$$
$$\varphi_{HP \text{ Min}} = + \operatorname{atan} \left(\frac{Z_{1 \text{ Min}}}{Z_{2 \text{ Max}}} \right)$$

RC_TP

KLASSENEIGENSCHAFTEN

Name: RC_TP
Eigenschaften: public, extends Filter

ATTRIBUTBESCHREIBUNG

-

OPERATIONSBESCHREIBUNG

KONSTRUKTOR

Name: RC_TP
Eigenschaften: public
Parameter: double Widerstand, double Toleranz des Widerstands, double Kapazität, double Toleranz der Kapazität
Rückgabe: -
Beschreibung: Instanziert die Klasse „RC_TP“.
Instanziert einen „R_ideal“ mit dem Widerstand und der Toleranz des Widerstands. Gibt diesen als „bauteil1“ an.
Instanziert einen „C_ideal“ mit der Kapazität und der Toleranz der Kapazität. Gibt diesen als „bauteil2“ an.

RC_HP

KLASSENEIGENSCHAFTEN

Name: RC_HP
Eigenschaften: public, extends Filter

ATTRIBUTBESCHREIBUNG

-

OPERATIONSBESCHREIBUNG

KONSTRUKTOR

Name: RC_HP
Eigenschaften: public
Parameter: double Widerstand, double Toleranz des Widerstands, double Kapazität, double Toleranz der Kapazität
Rückgabe: -
Beschreibung: Instanziert die Klasse „RC_HP“.
Instanziert einen „C_ideal“ mit der Kapazität und der Toleranz der Kapazität. Gibt diesen als „bauteil1“ an.
Instanziert einen „R_ideal“ mit dem Widerstand und der Toleranz des Widerstands. Gibt diesen als „bauteil2“ an.

RL_TP**KLASSENEIGENSCHAFTEN**

Name: RL_TP
Eigenschaften: public, extends Filter

ATTRIBUTBESCHREIBUNG

-

OPERATIONSBESCHREIBUNG**KONSTRUKTOR**

Name: RL_TP
Eigenschaften: public
Parameter: double Widerstand, double Toleranz des Widerstands, double Induktivität, double Toleranz der Induktivität
Rückgabe: -
Beschreibung: Instanziert die Klasse „RL_TP“.
Instanziert einen „L_ideal“ mit der Induktivität und der Toleranz der Induktivität. Gibt diesen als „bauteil1“ an.
Instanziert einen „R_ideal“ mit dem Widerstand und der Toleranz des Widerstands. Gibt diesen als „bauteil2“ an.

RL_HP

KLASSENEIGENSCHAFTEN

Name: RL_HP
Eigenschaften: public, extends Filter

ATTRIBUTBESCHREIBUNG

-

OPERATIONSBESCHREIBUNG

KONSTRUKTOR

Name: RL_HP
Eigenschaften: public
Parameter: double Widerstand, double Toleranz des Widerstands, double Induktivität, double Toleranz der Induktivität
Rückgabe: -
Beschreibung: Instanziert die Klasse „RL_HP“.
Instanziert einen „R_ideal“ mit dem Widerstand und der Toleranz des Widerstands. Gibt diesen als „bauteil1“ an.
Instanziert einen „L_ideal“ mit der Induktivität und der Toleranz der Induktivität. Gibt diesen als „bauteil2“ an.

E_REIHE

KLASSENEIGENSCHAFTEN

Name: E_Reihe
Eigenschaften: public

ATTRIBUTBESCHREIBUNG

NUMMER

Name: nummer
Datentyp: int
Eigenschaften: private
Set Methode: Nur positive Werte.
Get Methode: „holeNummer()“ als public anbieten.
Beschreibung: Die Nummer der E-Reihe gibt an, wie viele Grundwerte es pro Dekade gibt. (zB 12 für E12)

TOLERANZ

Name: toleranz
Datentyp: double
Eigenschaften: private
Set Methode: Nur positive Werte und null.
Get Methode: „holeToleranz()“ als public anbieten.
Beschreibung: Die Toleranz der E-Reihe bestimmt, die Toleranz der Werte, die in der E-Reihe vorhanden sind.

STDTOLERANZEN

Name: STDTOLERANZEN
Datentyp: hashMap
Eigenschaften: private, static
Set Methode: -
Get Methode: „holeStdToleranz(int nr)“ als public anbieten. Diese gibt die Standardtoleranz als double zurück, die mit der entsprechenden Nummer „nr“ verknüpft ist. Falls kein passender Eintrag zur Nummer „nr“ vorhanden ist, wird eine Exception geworfen.
Beschreibung: Die hashMap enthält die Standard Toleranzen der Standard E-Reihen.

E-Reihen Nr	Standard Toleranz
3	20%
6	20%
12	10%
24	5%
48	2%
96	1%
192	0,5%

OPERATIONSBESCHREIBUNG**KONSTRUKTOR**

Name: E_Reihe
Eigenschaften: public
Parameter: int Kennzahl der E-Reihe
Rückgabe: -
Beschreibung: Instanziert die Klasse „E_Reihe“. Setzt die „nummer“ auf die Kennzahl der E-Reihe. Verwendet (falls vorhanden) die Standard Toleranz aus der hashMap. Wenn nicht vorhanden wird die Toleranz auf 0 gesetzt.

WERT

Name: wert
Eigenschaften: public
Parameter: int index
Rückgabe: double Wert aus E-Reihe
Beschreibung: Gibt einen Wert aus der entsprechenden E_Reihe, die durch „nummer“ bestimmt ist zurück.

$$\text{Wert} = \text{nummer} \sqrt{10^{\text{index}}}$$

NAECHSTERWERT

Name: naechsterWert
Eigenschaften: public
Parameter: double Wert
Rückgabe: double nächster Wert aus E-Reihe
Beschreibung: Gibt einen Wert aus der entsprechenden E_Reihe, die durch „nummer“ bestimmt ist zurück.
Der zurückgegebene Wert hat die geringste Differenz zum eingegebenen Wert und ist größer als der eingegebenen Wert.

$$\text{Werte aus E_Reihe} = \text{nummer} \sqrt{10^{\text{index}}}$$

VORHERIGERWERT

Name: vorherigerWert
Eigenschaften: public
Parameter: double Wert
Rückgabe: double vorheriger Wert aus E-Reihe
Beschreibung: Gibt Wert aus der entsprechenden E_Reihe, die durch „nummer“ bestimmt ist zurück.
Der zurückgegebene Wert hat die geringste Differenz zum eingegebenen Wert und ist kleiner als der eingegebenen Wert.

$$\text{Werte aus E_Reihe} = \text{nummer} \sqrt{10^{\text{index}}}$$

RUNDEWERT

Name: rundeWert

Eigenschaften: public

Parameter: double Wert

Rückgabe: double Wert aus E-Reihe mit geringster Differenz zum angegebenen Wert

Beschreibung: Gibt Wert aus der entsprechenden E_Reihe, die durch „nummer“ bestimmt ist zurück.

Der zurückgegebene Wert hat die geringste Differenz zum eingegebenen Wert.

$$\text{Werte aus } E_Reihe = \overset{\text{nummer}}{\sqrt{10^{\text{index}}}}$$

RECHNER

KLASSENEIGENSCHAFTEN

Name: Rechner
Eigenschaften: public

ATTRIBUTBESCHREIBUNG

SCHALTUNG

Name: schaltung
Datentyp: int
Eigenschaften: private
Set Methode: Nur die Werte „0“ bis „3“. Diese „Set Methode“ muss dem Attribut „this.filter“ „NULL“ zugewiesen werden.
Get Methode: „holeSchaltung()“ als public anbieten.
Beschreibung: Der Integer kodiert, welche Filterschaltung im Rechner benutzt wird.

Schaltung Nr	„Name“
0	RC_TP
1	RC_HP
2	RL_TP
3	RL_HP

FILTER

Name: filter
Datentyp: Filter
Eigenschaften: private
Set Methode: -
Get Methode: -
Beschreibung: Die Instanz vom Typ Filter, die zur Berechnung der Werte genutzt wird. Bevor eine Berechnung durchgeführt werden kann, muss der Filter mit den vom Benutzer eingegebenen Daten durch eine der „erzeugeFilter Methoden“ instanziiert werden.

E_REIHE

Name: e_Reihe
Datentyp: E_Reihe
Eigenschaften: private
Set Methode: Ersetzt bisherige Instanz von E_Reihe mit einer Neuen. Und verwendet die Standardtoleranz aus „STDTOLERANZ“, wenn diese nicht angegeben ist, wird die Toleranz auf null gesetzt.
Get Methode: -
Beschreibung: Die E-Reihe, die im Rechner verwendet werden kann, um Bauteilwerte zu runden und auszuwählen.

OPERATIONSBESCHREIBUNG

KONSTRUKTOR

Name: Rechner
 Eigenschaften: public
 Parameter: int Kennnummer der Schaltung, int Kennzahl der E-Reihe
 Rückgabe: -
 Beschreibung: Instanziert die Klasse Rechner.
 Dabei werden die Kennnummer der Schaltung und die Kennzahl der E-Reihe übergeben.
 Die Kennnummer der Schaltung wird direkt durch die Set Methode gesetzt.
 Die e_Reihe wird durch ihren Konstruktor mit der Kennzahl der E-Reihe erzeugt.

SETZETOLERANZ

Name: setzeToleranz
 Eigenschaften: public
 Parameter: double Toleranz
 Rückgabe: void
 Beschreibung: Diese Methode setzt die Toleranz der e_Reihe.

ERZUEGEFILTER_RCL

Name: erzeugeFilter_rcl
 Eigenschaften: public
 Parameter: double Widerstand, double Kapazität oder Induktivität
 Rückgabe: void
 Beschreibung: Erzeugt eine Instanz von Filter, die im Attribut „filter“ gespeichert wird.
 Erzeugt den richtigen Subtyp von „Filter“ entsprechend der in „schaltung“ abgelegten Kennzahl der Schaltung.
 Übergibt zur Instanziierung von „Filter“ den Widerstandswert, die Kapazität oder Induktivität als Spulen- oder Kondensatorwert und die Toleranzen entsprechend der „toleranz“ von „e_Reihe“.

ERZUEGEFILTER_RF

Name: erzeugeFilter_rf
 Eigenschaften: public
 Parameter: double Widerstand, double Frequenz
 Rückgabe: void
 Beschreibung: Berechnet Kondensator- oder Spulenwert, rundet diesen und ruft „erzeugeFilter_rcl(double, double)“
 „Kapazität oder Induktivität“ berechnen:

„Kennzahl der Schaltung“ == „0“ „1“	„Kennzahl der Schaltung“ == „2“ „3“
CMath.c(r,f)	LMath.l(r,f)

Benutzt „e_Reihe.rundeWert(double)“ um den so errechneten Wert entsprechend aus der E-Reihe zu wählen.
 Ruft mit dem gerundeten Wert und dem Widerstand die Methode:
 „this.erzeugeFilter_rcl(r, cl)“.

ERZUEGEFILTER_ZF

Name: `erzeugeFilter_zf`
 Eigenschaften: `public`
 Parameter: `double` Eingangsimpedanz bei Grenzfrequenz, `double` Grenzfrequenz
 Rückgabe: `void`
 Beschreibung: Berechnet den Widerstandswert, rundet diesen und ruft „`erzeugeFilter_rf(double, double)`“
 Widerstand berechnen:

$$R = \sqrt{\frac{Z^2}{2}}$$

Benutzt „`e_Reihe.rundeWert(double)`“ um den so errechneten Wert entsprechend aus der E-Reihe zu wählen.

Ruft mit dem gerundeten Widerstand und der Grenzfrequenz die Methode: „`this.erzeugeFilter_rcl(r, f)`“.

ERZUEGEFILTER_CLF

Name: `erzeugeFilter_clf`
 Eigenschaften: `public`
 Parameter: `double` Kapazität oder Induktivität, `double` Frequenz
 Rückgabe: `void`
 Beschreibung: Berechnet den Widerstandswert, rundet diesen und ruft „`erzeugeFilter_rcl(double, double)`“
 „Widerstand berechnen:

„Kennzahl der Schaltung“ == „0“ „1“	„Kennzahl der Schaltung“ == „2“ „3“
<code>CMath.x(f,c)</code>	<code>LMath.x(f,l)</code>

Benutzt „`e_Reihe.rundeWert(double)`“ um den so errechneten Wert entsprechend aus der E-Reihe zu wählen.

Ruft mit dem gerundeten Widerstand und der Kapazität oder Induktivität die Methode: „`this.erzeugeFilter_rcl(r, cl)`“.

ERZUEGEFILTER_ZCL

Name: `erzeugeFilter_zcl`
 Eigenschaften: `public`
 Parameter: `double` Eingangsimpedanz bei Grenzfrequenz, `double` Kapazität oder Induktivität
 Rückgabe: `void`
 Beschreibung: Berechnet den Widerstandswert, rundet diesen und ruft „`erzeugeFilter_rcl(double, double)`“
 Widerstand berechnen:

$$R = \sqrt{\frac{Z^2}{2}}$$

Benutzt „`e_Reihe.rundeWert(double)`“ um den so errechneten Widerstand entsprechend aus der E-Reihe zu wählen.

Ruft mit dem gerundeten Widerstand und der Kapazität oder Induktivität die Methode: „`this.erzeugeFilter_rcl(r, cl)`“.

HOLEGRENZFREQUENZ

Name: holeGrenzfrequenz
Eigenschaften: public
Parameter: void
Rückgabe: double Grenzfrequenz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.grenzfrequenz()“ zurück.

HOLEGRENZFREQUENZMAX

Name: holeGrenzfrequenzMax
Eigenschaften: public
Parameter: void
Rückgabe: double maximale Grenzfrequenz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.grenzfrequenzMax()“ zurück.

HOLEGRENZFREQUENZMIN

Name: holeGrenzfrequenzMin
Eigenschaften: public
Parameter: void
Rückgabe: double minimale Grenzfrequenz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.grenzfrequenzMin()“ zurück.

HOLEIMPEDANZ

Name: holeImpedanz
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double Impedanz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter. impedanz (f)“ zurück.

HOLEIMPEDANZMAX

Name: holeImpedanzMax
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double maximale Impedanz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter. impedanzMax(f)“ zurück.

HOLEIMPEDANZMIN

Name: holeImpedanzMin
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double minimale Impedanz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.impedanzMin(f)“ zurück.

HOLEDAEMPfung

Name: holeDaempfung
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double logarithmisches Dämpfungsmaß bei einer Frequenz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.daempfung(f)“ zurück.

HOLEDAEMPfungMAX

Name: holeDaempfungMax
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double maximales logarithmisches Dämpfungsmaß bei einer Frequenz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.daempfungMax(f)“ zurück.

HOLEDAEMPfungMIN

Name: holeDaempfungMin
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double minimales logarithmisches Dämpfungsmaß bei einer Frequenz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.daempfungMin(f)“ zurück.

HOLEPHASENVERSCHIEBUNG

Name: holePhasenverschiebung
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double Phasenverschiebung bei einer Frequenz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.Phasenverschiebung(f)“ zurück.

HOLEPHASENVERSCHIEBUNGMAX

Name: holePhasenverschiebungMax
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double maximale Phasenverschiebung bei einer Frequenz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.PhasenverschiebungMax(f)“ zurück.

HOLEPHASENVERSCHIEBUNGMIN

Name: holePhasenverschiebungMin
Eigenschaften: public
Parameter: double Frequenz
Rückgabe: double minimale Phasenverschiebung bei einer Frequenz
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt diese Methode den Wert aus dem Aufruf von „this.filter.PhasenverschiebungMin(f)“ zurück.

HOLEWERTB1

Name: holeWertB1
Eigenschaften: public
Parameter: void
Rückgabe: double Wert des ersten Bauteils
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt den Wert des Bauteils „bauteil1“ zurück.
„this.filter.bauteil1.holeWert()“

HOLEWERTB1MAX

Name: holeWertB1Max
Eigenschaften: public
Parameter: void
Rückgabe: double maximaler Wert des ersten Bauteils
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt den maximalen Wert des Bauteils „bauteil1“ zurück.
„this.filter.bauteil1.holeWertMax()“

HOLEWERTB1MIN

Name: holeWertB1Min
Eigenschaften: public
Parameter: void
Rückgabe: double minimaler Wert des ersten Bauteils
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt den minimalen Wert des Bauteils „bauteil1“ zurück.
„this.filter.bauteil1.holeWertMin()“

HOLEWERTB2

Name: holeWertB2
Eigenschaften: public
Parameter: void
Rückgabe: double Wert des zweiten Bauteils
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt den Wert des Bauteils „bauteil2“ zurück.
„this.filter.bauteil2.holeWert()“

HOLEWERTB2MAX

Name: holeWertB2Max
Eigenschaften: public
Parameter: void
Rückgabe: double maximaler Wert des zweiten Bauteils
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt den maximalen Wert des Bauteils „bauteil2“ zurück.
„this.filter.bauteil2.holeWertMax()“

HOLEWERTB2MIN

Name: holeWertB2Min
Eigenschaften: public
Parameter: void
Rückgabe: double minimaler Wert des zweiten Bauteils
Beschreibung: Wenn „filter“ == „NULL“, dann Exception.
Sonst: Gibt den minimalen Wert des Bauteils „bauteil2“ zurück.
„this.filter.bauteil2.holeWertMin()“