Referenz für Integer

Konstruktoren

`Integer()`
Konstruktor für die Klasse `Integer`. Der Wert des Objektes ist null (=0).

`Integer(Integer i)`
Konstruktor für die Klasse `Integer`. Der Wert des Objektes ist gleich dem des Objektes `i`.

`Integer(String str)`
Konstruktor für die Klasse `Integer`. Enthält die übergebene Zeichenkette `str` eine gültige Ganzzahl, so wird sie konvertiert und als initialer Wert für das `Integer` Objekt verwendet. Enthält `str` eine ungültige Zeichenkette, so wird das `Integer` Objekt mit 0 initialisiert.

Methoden für arithmetische Operationen

`Integer add(Integer i)`
Addiert den Wert von `i` zu dem Wert des aufrufenden Objektes und gibt die Summe zurück.

`Integer add(Float f)`
Rundet den Wert von `f` zu einem Integer-Wert und addiert ihn zu dem Wert des aufrufenden Objektes. Als Ergebnis wird die Summe zurück gegeben.

`Integer sub(Integer i)`
Subtrahiert den Wert von `i` von dem Wert des aufrufenden Objektes und gibt die Differenz zurück.

`Integer sub(Float f)`
Rundet den Wert von `f` zu einem Integer-Wert und subtrahiert ihn von dem Wert des aufrufenden Objektes. Als Ergebnis wird die Differenz zurück gegeben.

`Integer mul(Integer i)`
Multipliziert den Wert von `i` mit dem Wert des aufrufenden Objektes und liefert das Produkt zurück.

`Integer mul(Float f)`
Rundet den Wert von `f` zu einem Integer-Wert und multipliziert ihn mit dem Wert des aufrufenden Objektes. Als Ergebnis wird das Produkt zurück gegeben.

`Integer div(Integer i)`
Dividiert den Wert des aufrufenden Objektes durch den Wert von `i` und liefert den Quotienten zurück.

`Integer div(Float f)`
Rundet den Wert von `f` zu einem Integer-Wert und dividiert den Wert des aufrufenden Objektes durch den gerundeten Wert. Als Ergebnis wird der Quotient zurück gegeben.

`Integer mod(Integer i)`
Dividiert den Wert des aufrufenden Objektes durch den Wert von `i`. Der Rest, der bei der Ganzzahldivision entsteht, wird zurück gegeben.

Methoden zum konvertieren

`String toString()`
Gibt den Wert des aufrufenden Objektes als Zeichenkette zurück.

Methoden zum Vergleichen

`Boolean equal(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes und der von `i` gleich sind, ansonsten `false`.

`Boolean equal(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes und der von `f` gleich sind, ansonsten `false`.

`Boolean greaterThan(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes größer als der von `i` ist, ansonsten `false`.

`Boolean greaterThan(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes größer als der von `f` ist, ansonsten `false`.

`Boolean greaterThanOrEqual(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes größer oder gleich dem von `i` ist, ansonsten `false`.

`Boolean greaterThanOrEqual(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes größer oder gleich dem von `f` ist, ansonsten `false`.

`Boolean lowerThan(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes kleiner als der von `i` ist, ansonsten `false`.

`Boolean lowerThan(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes kleiner als der von `f` ist, ansonsten `false`.

`Boolean lowerThanOrEqual(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes kleiner oder gleich dem von `i` ist, ansonsten `false`.

`Boolean lowerThanOrEqual(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes kleiner oder gleich dem von `f` ist, ansonsten `false`.

Referenz für Float

Konstruktoren

`Float()`
Konstruktor für die Klasse `Float`. Der Wert des Objektes ist null (=0).

`Flaot(Float f)`
Konstruktor für die Klasse `Float`. Der Wert des Objektes ist gleich dem des Objektes `f`.

`Float(String str)`
Konstruktor für die Klasse `Float`. Enthält die übergebene Zeichenkette `str` eine gültige Fließkommazahl, so wird sie konvertiert und als initialer Wert für das `Float` Objekt verwendet. Enthält `str` eine ungültige Zeichenkette, so wird das `Float` Objekt mit 0 initialisiert.

`Flaot(Integer i)`
Konstruktor für die Klasse `Float`. Der Wert des Objektes ist gleich dem des Objektes `i`.

Methoden für arithmetische Operationen

`Float add(Float f)`
Addiert den Wert von `f` zu dem Wert des aufrufenden Objektes und gibt die Summe zurück.

`Float add(Integer i)`
Addiert den Wert von `i` zu dem Wert des aufrufenden Objektes und gibt die Summe zurück.

`Float sub(Float f)`
Subtrahiert den Wert von `f` von dem Wert des aufrufenden Objektes und gibt die Differenz zurück.

`Float sub(Integer i)`
Subtrahiert den Wert von `i` von dem Wert des aufrufenden Objektes und gibt die Differenz zurück.

`Float mul(Float f)`
Multipliziert den Wert von `f` mit dem Wert des aufrufenden Objektes und liefert das Produkt zurück.

`Float mul(Integer i)`
Multipliziert den Wert von `i` mit dem Wert des aufrufenden Objektes und liefert das Produkt zurück.

`Float div(Float f)`
Dividiert den Wert des aufrufenden Objektes durch den Wert von `f` und liefert den Quotienten zurück.

`Float div(Integer i)`
Dividiert den Wert des aufrufenden Objektes durch den Wert von `i` und liefert den Quotienten zurück.

`Float sqrt()`
Berechnet die Wurzel des aufrufenden Objektes und gibt das Ergebnis zurück.

Methoden zum konvertieren

`Integer round()`
Konvertiert die Fließkommazahl in eine Ganzzahl, in dem die Fließkommazahl entsprechend gerundet wird.

`String toString()`
Gibt den Wert des aufrufenden Objektes als Zeichenkette zurück.

Methoden zum Vergleichen

`Boolean equal(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes und der von `f` gleich sind, ansonsten `false`.

`Boolean equal(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes und der von `i` gleich sind, ansonsten `false`.

`Boolean greaterThan(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes größer als der von `f` ist, ansonsten `false`.

`Boolean greaterThan(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes größer als der von `i` ist, ansonsten `false`.

`Boolean greaterThanOrEqual(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes größer oder gleich dem von `f` ist, ansonsten `false`.

`Boolean greaterThanOrEqual(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes größer oder gleich dem von `i` ist, ansonsten `false`.

`Boolean lowerThan(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes kleiner als der von `f` ist, ansonsten `false`.

`Boolean lowerThan(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes kleiner als der von `i` ist, ansonsten `false`.

`Boolean lowerThanOrEqual(Float f)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes kleiner oder gleich dem von `f` ist, ansonsten `false`.

`Boolean lowerThanOrEqual(Integer i)`
Liefert `true` zurück, falls der Wert des aufrufenden Objektes kleiner oder gleich dem von `i` ist, ansonsten `false`.

Referenz für String

Konstruktoren

`String()`
Konstruktor für die Klasse `String`. Der Wert des Objektes ist eine leere Zeichenkette.

`String(String str)`
Konstruktor für die Klasse `String`. Die Zeichenkette des Objektes ist gleich dem des Objektes `str`.

Methoden zum manipulieren der Zeichenkette

`void append(String str)`
Hängt an die Zeichenkette des aufrufenden Objektes die Zeichenkette von `str` hinten an.

`void append(Integer i)`
Hängt an die Zeichenkette des aufrufenden Objektes die Zahl von `i` hinten an.

`void append(Float f)`
Hängt an die Zeichenkette des aufrufenden Objektes die Zahl von `f` hinten an.

`String toLowerCase()`
Gibt eine Zeichenkette zurück, bei der alle Großbuchstaben des aufrufenden Objektes in Kleinbuchstaben umgewandelt wurden.

`String toUpperCase()`
Gibt eine Zeichenkette zurück, bei der alle Kleinbuchstaben des aufrufenden Objektes in Großbuchstaben umgewandelt wurden.

`String charAt(Integer i)`
Gibt eine Zeichenkette zurück, die nur das `i`-te Zeichen der Zeichenkette des aufrufenden Objektes enthält.

`String substring(Integer beginIndex, Integer endIndex)`
Gibt eine Teilzeichenkette zurück, vom `beginIndex`-ten Zeichen bis zum `endIndex`-ten Zeichen. Hinweis: Das erste Zeichen ist an 0-ten Position, das letzte an der (Länge-1)-ten Position.

Methoden zur Analyse

`Integer length()`
Gibt die Länge der Zeichenkette zurück.

`Integer indexOf(String str)`
Sucht in der Zeichenkette die Zeichenkette `str`. Der Rückgabewert ist die Postion, an der `str` erstmals vorkommt. Kommt `str` in der Zeichenkette nicht vor, so ist der Rückgabewert -1.

Methoden zum Vergleichen

`Boolean equal(String str)`
Liefert `true` zurück, falls die Zeichenkette des aufrufenden Objektes und der von `str` gleich sind, ansonsten `false`.

Referenz für Boolean

Konstruktoren

`Boolean()`
Konstruktor für die Klasse `Boolean`. Der Wert des Objektes ist `false`.

`Boolean(Boolean b)`
Konstruktor für die Klasse `Boolean`. Der Wert des Objektes ist gleich dem des Objektes `b`.

Methoden für logische Operationen

`Boolean or(Boolean b)`
Gibt den Wert `true` zurück, wenn entweder der Wert des aufrufenden Objektes oder `b` oder beide gleich `true` sind. Ansonsten wird `false` zurück gegeben.

`Boolean and(Boolean b)`
Gibt den Wert `true` zurück, wenn der Wert des aufrufenden Objektes und `b` `true` sind. Ansonsten wird `false` zurück gegeben.

`Boolean not()`
Gibt den Wert `true` zurück, wenn der Wert des aufrufenden Objektes und `false` ist. Ansonsten wird `false` zurück gegeben.

Referenz für Computer

Konstruktor

`Computer()`
Konstruktor für die Klasse `Computer`.

Methoden zur Ein- und Ausgabe

`String read(String label)`
Öffnet ein Dialogfenster zur Eingabe von Zeichenketten über die Tastatur. Das Eingabefeld wird mit dem Text aus `label` versehen. Die Benutzereingabe wird als `String` zurück gegeben.

`void print(String str)`
Gibt die Zeichenkette `str` auf der Konsole aus.

`void print(Integer i)`
Gibt die Ganzzahl `i` auf der Konsole aus.

`void print(Float f)`
Gibt die Fließkommazahl `f` auf der Konsole aus.

`void println()`
Gibt einen Zeilenvorschub auf der Konsole aus..

Methoden zur Steuerung des Programms

`void wait(Integer msec)`
Unterbricht die Ausführung des Programms um `msec` Millisekunden.

Methoden zur Erzeugung von Zufallszahlen

`Integer rand(Integer min, Integer max)`
Gibt eine Zufallszahl größer oder gleich `min` und kleiner oder gleich `max` zurück.