# FEM – Function Evaluation Model

Die «Function Evaluation Model» oder kurz «FEM» beschreibt ein generisches Datenmodell zur Auswertung abstrakter Funktionen im Rahmen gegebener Parameterwerte und Kontextinformationen.

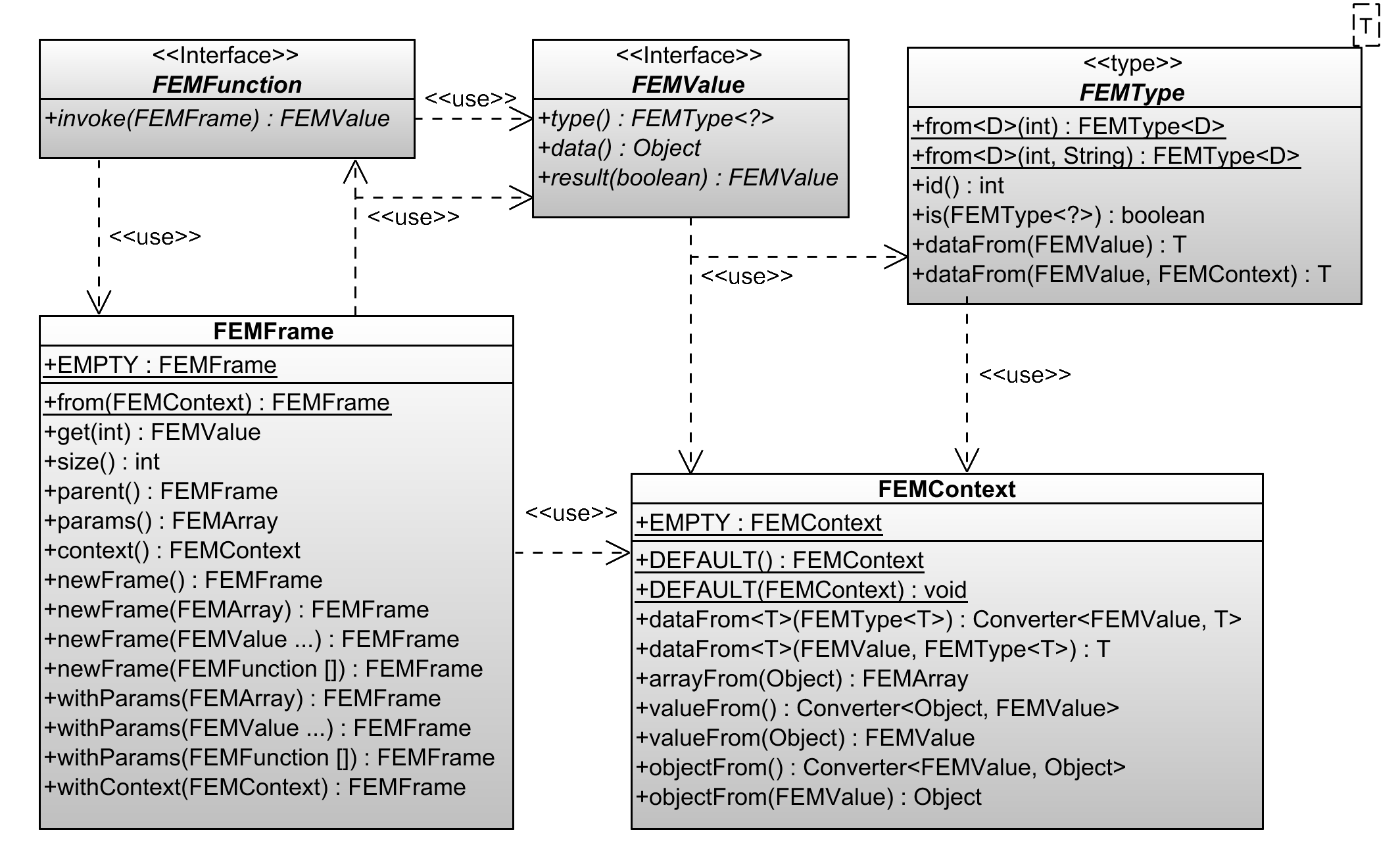


Abbildung 1 Function Evaluation Model

Eine Funktion («FEMFunction») wird mit einem Stapelrahmen («FEMFrame») aufgerufen und liefert einen Ergebniswert («FEMValue»). Werte besitzen einen Datentyp («FEMType») sowie Nutzdaten, welche über den impliziten bzw. ein expliziten Kontext («FEMContext») in andere Datenformate umgewandelt werden können.

## FEMFunction

+ bee.creative.fem.FEMFunction

Diese Schnittstelle definiert eine Funktion, deren Berechnungsmethode mit einem Stapelrahmen aufgerufen wird und einen Ergebniswert liefert. Aus den Parameterwerten sowie dem Kontextobjekt der Stapelrahmens können hierbei Informationen für die Berechnungen extrahiert werden. Der Zustand des Kontextobjekts kann auch modifiziert werden.

+ invoke(frame: FEMFrame): FEMValue

Diese Methode führt Berechnungen im gegebenen Stapelrahmen durch und gibt den ermittelten Ergebniswert zurück.

## FEMValue

+ bee.creative.fem.FEMValue

Diese Schnittstelle definiert einen Wert, der als Ergebnis der Auswertung einer Funktion oder als Parameter in einem Stapelrahmen zur Auswertung einer Funktion verwendet werden kann. Ein solcher Wert besitzt dazu Nutzdaten mit einem bestimmten Datentyp. Die Konvertierung der Nutzdaten in einen gegebenen Datentyp type kann im Rahmen eines gegebenen Kontextobjekts «context» über den Befehl «context.dataFrom(this, type)» erfolgen.

+ type(): FEMType<?>

Diese Methode gibt den Datentyp zurück.

+ data(): Object

Diese Methode gibt die Nutzdaten zurück.

+ result(recursive: boolean): FEMValue

Diese Methode gibt diesen Wert als ausgewerteten Ergebniswert zurück. Bei rekursiver Auswertung werden auch die in diesem Wert enthaltenen Werte ausgewertet, z.B. bei einem «FEMArray». Andernfalls wird nur dieser Wert ausgewertet, z.B. bei einem «FEMResult».

## FEMType

+ bee.creative.fem.FEMType<GData>

Diese Klasse implementiert den abstrakten Datentyp eines Werts («FEMValue»), analog zur Klasse eines Objekts. Ein solcher Datentyp besitzt Methoden zum Konvertieren der Nutzdaten eines gegebenen Werts sowie zur Prüfung der Kompatibilität zu anderen Datentypen. «GData» ist hierbei der Typ der von «FEMType.dataFrom(FEMValue)» bzw. «FEMType.dataFrom(FEMValue, FEMContext)» gelieferten Nutzdaten.

+ from<GData>(id: int): FEMType<GData>

Diese Methode gibt einen einfachen Datentyp mit dem gegebenen Identifikator zurück.

+ from<GData>(id: int, toString: String): FEMType<GData>

Diese Methode gibt einen einfachen Datentyp mit dem gegebenen Identifikator und der gegebenen Textdarstellung zurück.

+ id(): int

Diese Methode gibt den Identifikator dieses Datentyps zurück. Dieser sollte über eine statische Konstante definiert werden, um Fallunterscheidungen mit einem *switch*-Statement umsetzen zu können.

+ is(type: FEMType<?>): boolean

Diese Methode gibt nur dann «true» zurück, wenn ein *cast* in den gegebenen Datentyp zulässig ist. Dies kann der Fall sein, wenn der gegebene Datentyp gleich zu diesem oder ein Vorfahre dieses Datentyps ist. Wenn der gegebene Datentyp «null» ist, wird «false» geliefert.

+ dataFrom(value: FEMValue): GData

Diese Methode gibt die in den Datentyp «T» konvertierten Nutzdaten des gegebenen Werts zurück. Der Rückgabewert entspricht «Context.DEFAULT().dataFrom(value, this)».

+ dataFrom(value: FEMValue, context: FEMContext): GData

Diese Methode gibt die in den Datentyp «T» kontextsensitiv konvertierten Nutzdaten des gegebenen Werts zurück. Der Rückgabewert entspricht «context.dataFrom(value, this)».

## FEMFrame

+ bee.creative.fem.FEMFrame

Diese Klasse implementiert einen Stapelrahmen (*stack-frame*), über welchen einer Funktion eine Liste von Parameterwerten sowie ein Kontextobjekt zur Verfügung gestellt wird. Über die Anzahl der zugesicherten Parameterwerte hinaus können von der Methode «get(index)» auch zusätzliche Parameterwerte aus dem übergeordneten Stapelrahmen bereitgestellt werden. Diese Methode liefert für einen gegebenen «index» immer den gleichen Wert bzw. löst immer die gleiche Ausnahme aus. Analoges gilt für die Methoden «size()», «params()» und «context()».

+ EMPTY: FEMFrame

Dieses Feld speichert den leeren Stapelrahmen, der keine Parameterwerte bereitstellt, das Kontextobjekt «FEMContext.EMPTY» verwendet und sich selbst als «parent()» nutzt.

+ from(context: FEMContext): FEMFrame

Diese Methode gibt einen Stapelrahmen mit dem gegebenen Kontextobjekt zurück. Sie ist eine Abkürzung für «EMPTY.withContext(context)».

+ get(index: int): FEMValue

Diese Methode gibt den «index»-ten Parameterwert evaluiert zurück. Über die Anzahl der zugesicherten Parameterwerte hinaus, können auch zusätzliche Parameterwerte des übergeordneten Stapelrahmens bereitgestellt werden. Genauer wird für einen «index» größer oder gleich der Anzahl der zugesicherten Parameter der Parameterwert «parent().get(index - size())» des übergeordneten Stapelrahmens geliefert.

+ size(): int

Diese Methode gibt die Anzahl der Parameterwerte zurück, die zur Verwendung durch eine aufgerufene Funktion bestimmt sind. Über die «get(index)»-Methode werden mindestens so viele Parameterwerte bereitgestellt.

+ parent(): FEMFrame

Diese Methode gibt die übergeordneten Parameterdaten zurück.

+ params(): FEMArray

Diese Methode gibt eine Wertliste als Sicht auf die zugesicherten Parameterwerte zurück. Die Elemente dieser Wertliste können der *return-by-reference*-Semantik angehören.

+ context(): FEMContext

Diese Methode gibt das Kontextobjekt zurück. Funktionen können aus diesem Objekt Informationen für ihre Berechnungen extrahieren oder auch den Zustand dieses Objekts modifizieren. Das Kontextobjekt entspricht dem Kontext «this» in Java-Methoden.

+ newFrame(): FEMFrame

Diese Methode gibt einen neuen Stapelrahmen zurück, welcher keine zugesicherten Parameterwerte besitzt, das Kontextobjekt dieses Stapelrahmens übernimmt und diesen als übergeordneten Stapelrahmen nutzt.

+ newFrame(params: FEMArray): FEMFrame

Diese Methode gibt einen neuen Stapelrahmen zurück, welcher die gegebenen zugesicherten Parameterwerte besitzt, das Kontextobjekt dieses Stapelrahmens übernimmt und diesen als übergeordneten Stapelrahmen nutzt.

+ newFrame(params: FEMValue…): FEMFrame

Diese Methode gibt einen neuen Stapelrahmen zurück, welcher die gegebenen zugesicherten Parameterwerte besitzt, das Kontextobjekt dieses Stapelrahmens übernimmt und diesen als übergeordneten Stapelrahmen nutzt.

+ newFrame(functions: FEMFunction[]): FEMFrame

Diese Methode gibt einen neuen Stapelrahmen zurück, welcher die gegebenen Parameterfunktionen zur Berechnung der zugesicherten Parameterwerte verwendet, das Kontextobjekt dieses Stapelrahmens übernimmt und diesen als übergeordneten Stapelrahmen nutzt.

Die zugesicherten Parameterwerte werden mit Hilfe dieses Stapelrahmens und der gegebener Parameterfunktionen ermittelt. Eine Parameterfunktion wird zur Ermittlung eines Parameterwerts einmalig mit diesem Stapelrahmen ausgewertet. Genauer entspricht der «index»-te zugesicherte Parameterwert dem Ergebnis von «functions[index].invoke(this)». Der Ergebniswert wird zur Wiederverwendung zwischengespeichert. Die über «params()» bereitgestellte Liste der Parameterwerte des erzeugten Stapelrahmens liefert die noch nicht über die «get(index)»-Methode ermittelten Parameterwerte als «FEMResult».

withParams(  
  array: FEMArray)  
: FEMFrame

Diese Methode gibt diesen Stapelrahmen mit den gegebenen zugesicherten Parameterwerten zurück. Sie ist eine Abkürzung für «parent().newFrame(params).withContext(context())».

withParams(  
  values: FEMValue…)  
: FEMFrame

withParams(  
  functions: FEMFunction[])  
: FEMFrame

withContext(  
  context: FEMContext)  
: FEMFrame

Diese Methode gibt diesen Stapelrahmen mit dem gegebenen Kontextobjekt zurück.

Tabelle 1‑4 FEM-Schnittstelle «FEMFrame»

|  |  |
| --- | --- |
| FEMContect | |
| Diese Klasse implementiert ein abstraktes Kontextobjekt, das über einen Stapelrahmen der Auswertung von Funktionen bereitgestellt wird und in Funktionen zur Umwandlung von Werten genutzt werden kann. | |
| Mitglied | Beschreibung |
| EMPTY : FEMContect | Dieses Feld speichert das leere Kontextobjekt.  Die Methode «dataFrom(value, type)» dieses Kontextobjekts gibt die Nutzdaten des ihr übergebenen Werts «value» unverändert zurück, wenn sein Datentyp gleich oder einem Nachfahren des ihr übergebenen Datentyps type ist, d.h. wenn «value.type().is(type)». Andernfalls löst sie eine «IllegalArgumentException» aus.  Die «valueFrom(object)»-Methode dieses Kontextobjekts gibt einen gegebenen «FEMValue» unverändert zurück und konvertiert «null» zu «FEMVoid», «char[]» und «String» zu «FEMString», «byte[]» zu «FEMBinary», «Float», «Double» und «BigDecimal» zu «FEMDecimal», alle anderen «Number» zu «FEMInteger», «Boolean» zu «FEMBoolean», «Calendar» zu «FEMDatetime», «FEMFunction» zu «FEMHandler» und alle anderen Eingaben via «arrayFrom(object)» in ein «FEMArray». Im Fehlerfall löst sie eine «IllegalArgumentException» aus.  Die «objectFrom(value)»-Methode dieses Kontextobjekts konvertiert «FEMVoid» zu «null», «FEMArray» und die darin enthaltenen Werte rekursiv zu «Object[]», «FEMBinary» zu «byte[]», «FEMString» zu «String», «FEMInteger» und «FEMDecimal» zu «Number», «FEMDatetime» zu «Calendar», «FEMBoolean» zu «Boolean» und alle anderen Werte ihren Nutzdatensatz. |
| DEFAULT() : FEMContect | Diese Methode gibt das Kontextobjekt zurück, das als Rückfallebene für kontextfeie Datentypumwandlungen genutzt wird. Dieses Rückfallkontextobjekt wird u.a. in «FEMType.dataFrom(value)» verwendet. |
| DEFAULT(   context: FEMContect) : void | Diese Methode setzt den Rückfallkontextobjekt. Wenn das gegebene Kontextobjekt «null» ist, wird «EMPTY» verwendet. |
| dataFrom<D>(   type: FEMType<D>) : Converter<FEMValue, D> | Diese Methode gibt einen «Converter» zurück, der seine Eingabe «input» über den Aufruf «dataFrom(input, type)» in seine Ausgabe überführt. |
| dataFrom<D>(   value: FEMValue,   type: FEMType<D>) : D | Diese Methode gibt die in Nutzdaten des gegebenen Werts im gegebenen Datentyp «T» zurück. Hierbei werden die Nutzdaten «value.data()» in den geforderten Datentyp konvertiert. |
| arrayFrom(   object: Object) : FEMArray | Diese Methode konvertiert das gegebene Objekt in eine Wertliste und gibt diese zurück.  Wenn das Objekt ein «FEMArray» ist, wird es unverändert zurückgegeben. Wenn es ein natives Array ist, wird jedes seiner Elemente über die «valueFrom(object)»-Methode in einen Wert überführt und die so entstandene Wertliste geliefert. Wenn es eine «Collection» ist, wird diese in ein natives Array überführt, welches anschließend in eine Wertliste umgewandelt wird. Wenn es ein «Iterable» ist, wird dieses in eine «Collection» überführt, welche anschließend in eine Wertliste umgewandelt wird. Andernfalls wird eine «IllegalArgumentException» ausgelöst. |
| valueFrom() : Converter<Object, FEMValue> | Diese Methode gibt einen «Converter» zurück, der seine Eingabe «input» via «valueFrom(input)» in seine Ausgabe überführt. |
| valueFrom(   object: Object) : FEMValue | Diese Methode gibt einen Wert mit den gegebenen Nutzdaten zurück. Welcher Wert- und Datentyp hierfür verwendet wird, ist der Implementation überlassen. |
| objectFrom() : Converter<FEMValue, Object> | Diese Methode gibt einen «Converter» zurück, der seine Eingabe «input» via «objectFrom(input)» in seine Ausgabe überführt. |
| objectFrom(   value: FEMValue) : Object | Diese Methode gibt ein «Object» zurück, welches via «valueFrom(object)» in einen Wert überführt werden kann, der zum gegebenen Wert äquivalenten ist. |

Tabelle 1‑5 FEM-Schnittstelle «FEMContect»

# FEM-Werte

## BaseValue

|  |  |
| --- | --- |
| BaseValue | |
| Diese Klasse implementiert einen abstrakten Wert als «FEMFunction». Die «invoke(frame)»-Methode liefert «this», sodass Instanzen dieser Klassen das Einpacken in eine «ValueFunction» nicht benötigen. | |
| Mitglied | Beschreibung |
| data<D>(   type: FEMType<D>) : D | Diese Methode gibt die in den gegebenen Datentyp kontextfrei konvertierten Nutzdaten dieses Werts zurück. Der Rückgabewert entspricht «FEMContext.DEFAULT().dataFrom(this, type)». |
| data<D>(   type: FEMType<D>,   context: FEMContext) : D | Diese Methode gibt die in den gegebenen Datentyp kontextsensitiv konvertierten Nutzdaten dieses Werts zurück. Der Rückgabewert entspricht «context.dataFrom(this, type)». |
| result(   recursive: boolean) : FEMValue | Diese Methode gibt «this» zurück. |
| invoke(   frame: FEMFrame) : FEMValue |

Tabelle 1 FEM-Wert «BaseValue»

## FEMVoid

|  |  |
| --- | --- |
| FEMVoid | |
| Diese Klasse implementiert den unveränderlichen Leerwert. | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMVoid> | Dieses Feld speichert den Datentyp. |
| INSTANCE : FEMVoid | Dieses Feld speichert den Leerwert. |
| from(   string: String) : FEMVoid | Diese Methode gibt den Leerwert nur dann zurück, wenn die gegebene Zeichenkette gleich «"void"» ist. |
| from(   value: FEMValue) : FEMVoid | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMVoid | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| data() : FEMVoid | Diese Methode gibt «this» zurück. |
| type() : FEMType<FEMVoid> | Diese Methode «TYPE» zurück. |
| toString() : String | Diese Methode gibt «"void"» zurück. |

Tabelle 2 FEM-Wert «FEMVoid»

## FEMArray

|  |  |
| --- | --- |
| FEMArray | |
| Diese Klasse implementiert eine unveränderliche Liste von Werten sowie Methoden zur Erzeugung solcher Wertlisten aus nativen Arrays und «Iterable». | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMArray> | Dieses Feld speichert den Datentyp. |
| EMPTY  : FEMArray | Dieses Feld speichert eine leere Wertliste. |
| from(   values: FEMValue…) : FEMArray | Diese Methode konvertiert die gegebenen Werte in eine Wertliste und gibt diese zurück. Das gegebene Array wird Kopiert, sodass spätere Änderungen am gegebenen Array nicht auf die erzeugte Wertliste übertragen werden. |
| from(   values: FEMValue,   count: int) : FEMArray | Diese Methode gibt eine uniforme Wertliste mit der gegebenen Länge zurück, deren Werte alle gleich dem gegebenen sind. |
| from(   data: Iterable<FEMValue>) : FEMArray | Diese Methode konvertiert die gegebenen Werte in eine Wertliste und gibt diese zurück. |
| from(   data: Collection<FEMValue>) : FEMArray |
| from(   value: FEMValue) : FEMArray | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMArray | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| data() : FEMArray | Diese Methode gibt «this» zurück. |
| type() : FEMType<FEMArray> | Diese Methode «TYPE» zurück. |
| value() : FEMValue[] | Diese Methode konvertiert diese Wertliste in ein «FEMValue[]» und gibt dieses zurück. |
| get(   index: int) : FEMValue | Diese Methode gibt den «index»-ten Wert zurück.  Wenn «index» ungültig ist, wird eine «IndexOutOfBoundsException» ausgelöst. |
| length() : int | Diese Methode gibt die Länge, d.h. die Anzahl der Werte in der Wertliste zurück. |
| concat(   that: FEMArray) : FEMArray | Diese Methode gibt eine Sicht auf die Verkettung dieser Wertliste mit der gegebenen Wertliste zurück. |
| section(   offset: int,   length: int) : FEMArray | Diese Methode gibt eine Sicht auf einen Abschnitt dieser Wertliste zurück.  Wenn der Abschnitt nicht innerhalb dieser Wertliste liegt oder eine negative Länge hätte, wird eine «IllegalArgumentException» ausgelöst. |
| reverse() : FEMArray | Diese Methode gibt eine rückwärts geordnete Sicht auf diese Wertliste zurück. |
| compact() : FEMArray | Diese Methode gibt die Werte dieser Wertliste in einer performanteren oder zumindest gleichwertigen Wertliste zurück. |
| find(   that: FEMValue,   offset: int) : int | Diese Methode gibt die Position des ersten Vorkommens des gegebenen Werts «that» innerhalb dieser Wertliste zurück. Die Suche beginnt an der gegebenen Position «offset». Bei einer erfolglosen Suche wird «-1» geliefert.  Wenn «offset» ungültig ist, wird eine «IllegalArgumentException» ausgelöst. |
| find(   that: FEMArray,   offset: int) : int | Diese Methode gibt die Position des ersten Vorkommens der gegebenen Wertliste «that» innerhalb dieser Wertliste zurück. Die Suche beginnt an der gegebenen «offset». Bei einer erfolglosen Suche wird «-1» geliefert.  Wenn «offset» ungültig ist, wird eine «IllegalArgumentException» ausgelöst. |
| export(   target: FEMArray.Collector) : boolean | Diese Methode fügt alle Werte dieser Wertliste vom ersten zum letzten geordnet an den gegebenen «Collector» «target» an. Das Anfügen wird vorzeitig abgebrochen, wenn dessen «push(value)»-Methode «false» liefert. |
| hash() : int | Diese Methode gibt den Streuwert zurück. |
| equals(   that: FEMArray) : boolean | Diese Methode gibt nur dann «true» zurück, wenn diese Wertliste gleich der gegebenen ist. |
| compare(   that: FEMArray,   order: Comparator<FEMValue>) : int | Diese Methode gibt «-1», «0» bzw. «+1» zurück, wenn die lexikographische Ordnung dieser Wertliste kleiner, gleich oder größer als die der gegebenen Wertliste ist. Die Werte werden über den gegebenen «Comparator» «order» verglichen. |
| toList() : List<FEMValue> | Diese Methode gibt diese Wertliste als «List» zurück. |

Tabelle 3 FEM-Wert «FEMArray»

|  |  |
| --- | --- |
| FEMArray.Collector | |
| Diese Schnittstelle definiert ein Objekt zum geordneten Sammeln von Werten einer Wertliste in der Methode «FEMArray.export(target)». | |
| Mitglied | Beschreibung |
| push(   value: FEMValue) : boolean | Diese Methode fügt den gegebenen Wert an das Ende der Sammlung an und gibt nur dann «true» zurück, wenn das Sammeln fortgeführt werden soll. Der Rückgabewert ist «true», wenn das Sammeln fortgeführt werden soll, bzw. «false», wenn es abgebrochen werden soll. |

Tabelle 4 FEM-Wert «FEMArray.Collector»

## FEMBinary

|  |  |
| --- | --- |
| FEMBinary | |
| Diese Klasse implementiert eine unveränderliche Bytefolge, deren Verkettungen, Anschnitte und Umkehrungen als Sichten auf die grundlegenden Bytefolgen realisiert sind. | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMBinary> | Dieses Feld speichert den Datentyp. |
| EMPTY  : FEMBinary | Dieses Feld speichert eine leere Wertliste. |
| from(   values: byte[]) : FEMBinary | Diese Methode gibt eine Bytefolge mit den gegebenen Bytes zurück. Das gegebene Array wird hierbei kopiert. |
| from(   value: byte,    length: int) : FEMBinary | Diese Methode gibt eine uniforme Bytefolge mit der gegebenen Länge zurück, deren Bytes alle gleich dem gegebenen sind. |
| from(   string: String) : FEMBinary | Diese Methode gibt eine neue Bytefolge mit dem in der gegebenen Zeichenkette kodierten Wert zurück. Das Format der Zeichenkette entspricht dem der Textdarstellung. |
| from(   data: MMFArray) : FEMBinary | Diese Methode gibt eine Bytefolge mit den gegebenen Zahlen zurück.  Wenn diese Zahlen nicht als «UNI8» oder «UINT8» vorliegt, wird eine «IllegalArgumentException» ausgelöst. |
| from(   value: FEMValue) : FEMBinary | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMBinary | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| toChar(   hexDigit: int) : int | Diese Methode gibt das Zeichen («'0'…'9'», «'A'…'F'») zur gegebenen hexadezimalen Ziffer («0…15») zurück.  Wenn «hexDigit» ungültig ist, wird eine «IllegalArgumentException» ausgelöst. |
| toDigit(   hexChar: int) : int | Diese Methode gibt die hexadezimale Ziffer («0…15») zum gegebenen Zeichen («'0'…'9'», «'A'…'F'») zurück.  Wenn «hexChar» ungültig ist, wird eine «IllegalArgumentException» ausgelöst. |
| data() : FEMBinary | Diese Methode gibt «this» zurück. |
| type() : FEMType<FEMBinary> | Diese Methode «TYPE» zurück. |
| value() : byte[] | Diese Methode konvertiert diese Bytefolge in ein «byte[]» und gibt dieses zurück. |
| get(   index: int) : byte | Diese Methode gibt das «index»-te Byte zurück.  Wenn «index» ungültig ist, wird eine «IndexOutOfBoundsException» ausgelöst. |
| length() : int | Diese Methode gibt die Länge, d.h. die Anzahl der Bytes in der Bytefolge zurück. |
| concat(   that: FEMBinary) : FEMBinary | Diese Methode gibt eine Sicht auf die Verkettung dieser Bytefolge mit der gegebenen Bytefolge zurück. |
| section(   offset: int,   length: int) : FEMBinary | Diese Methode gibt eine Sicht auf einen Abschnitt dieser Bytefolge zurück.  Wenn der Abschnitt nicht innerhalb dieser Bytefolge liegt oder eine negative Länge hätte, wird eine «IllegalArgumentException» ausgelöst. |
| reverse() : FEMBinary | Diese Methode gibt eine rückwärts geordnete Sicht auf diese Bytefolge zurück. |
| compact() : FEMBinary | Diese Methode gibt die Bytes dieser Bytefolge in einer performanteren oder zumindest gleichwertigen Bytefolge zurück. |
| find(   that: FEMValue,   offset: int) : int | Diese Methode gibt die Position des ersten Vorkommens des gegebenen Bytewerts «that» innerhalb dieser Bytefolge zurück. Die Suche beginnt an der gegebenen Position «offset». Bei einer erfolglosen Suche wird «-1» geliefert.  Wenn «offset» ungültig ist, wird eine «IllegalArgumentException» ausgelöst. |
| find(   that: FEMBinary,   offset: int) : int | Diese Methode gibt die Position des ersten Vorkommens der gegebenen Bytefolge «that» innerhalb dieser Bytefolge zurück. Die Suche beginnt an der gegebenen «offset». Bei einer erfolglosen Suche wird «-1» geliefert.  Wenn «offset» ungültig ist, wird eine «IllegalArgumentException» ausgelöst. |
| export(   target: FEMBinary.Collector) : boolean | Diese Methode fügt alle Bytes dieser Bytefolge vom ersten zum letzten geordnet an den gegebenen «Collector» «target» an. Das Anfügen wird vorzeitig abgebrochen, wenn dessen «push(value)»-Methode «false» liefert. |
| hash() : int | Diese Methode gibt den Streuwert zurück. |
| equals(   that: FEMBinary) : boolean | Diese Methode gibt nur dann «true» zurück, wenn diese Wertliste gleich der gegebenen ist. |
| compare(   that: FEMBinary) : int | Diese Methode gibt «-1», «0» bzw. «+1» zurück, wenn die lexikographische Ordnung dieser Bytefolge kleiner, gleich oder größer als die der gegebenen Bytefolge ist. Die Bytewerte werden als «UINT8» verglichen. |
| toString() : String | Diese Methode gibt die Textdarstellung dieser Bytefolge zurück. Die Textdarstellung besteht aus der Zeichenkette «"0x"» und den Bytes dieser Bytefolge vom ersten zum letzten geordnet in hexadezimalen Ziffern («'0'…'9'», «'A'…'F'»). |

Tabelle 5 FEM-Wert «FEMBinary»

|  |  |
| --- | --- |
| FEMBinary.Collector | |
| Diese Schnittstelle definiert ein Objekt zum geordneten Sammeln von Bytes einer Bytefolge in der Methode «FEMBinary.export(target)». | |
| Mitglied | Beschreibung |
| push(   value: FEMValue) : boolean | Diese Methode fügt den gegebenen Bytewert an das Ende der Sammlung an und gibt nur dann «true» zurück, wenn das Sammeln fortgeführt werden soll. Der Rückgabewert ist «true», wenn das Sammeln fortgeführt werden soll, bzw. «false», wenn es abgebrochen werden soll. |

Tabelle 6 FEM-Wert «FEMBinary.Collector»

## FEMBoolean

|  |  |
| --- | --- |
| FEMBoolean | |
| Diese Klasse implementiert einen unveränderlichen Wahrheitswert. Intern wird der Wahrheitswert als «boolean» dargestellt. | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMBoolean> | Dieses Feld speichert den Datentyp. |
| TRUE : FEMBoolean | Dieses Feld speichert den Wahrheitswert «true». |
| FALSE : FEMBoolean | Dieses Feld speichert den Wahrheitswert «false». |
| from(   data: boolean) : FEMBoolean | Diese Methode gibt einen neuen Wahrheitswert mit dem gegebenen Wert zurück. |
| from(   data: Boolean) : FEMBoolean |
| from(   string: String) : FEMVoid | Diese Methode gibt einen neuen Wahrheitswert mit dem in der gegebenen kodierten Wert zurück. Das Format der Zeichenkette entspricht dem der Textdarstellung. |
| from(   value: FEMValue) : FEMVoid | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMVoid | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| FEMBoolean(   value: boolean) | Dieser Konstruktor initialisiert die interne Darstellung des Wahrheitswerts. |
| type() : FEMType<FEMVoid> | Diese Methode «TYPE» zurück. |
| data() : FEMVoid | Diese Methode gibt «this» zurück. |
| value() : boolean | Diese Methode gibt die interne Darstellung des Wahrheitswerts zurück. |
| hash() : int | Diese Methode gibt den Streuwert zurück. |
| equals(   that: FEMBoolean) : boolean | Diese Methode gibt nur dann «true» zurück, wenn dieser Wahrheitswert gleich dem gegebenen ist. |
| compare(   that: FEMBoolean) : int | Diese Methode gibt «-1», «0» bzw. «+1» zurück, wenn dieser Wahrheitswert kleiner, gleich oder größer als der gegebene Wahrheitswert ist. |
| toString() : String | Diese Methode gibt die Textdarstellung dieses Wahrheitswerts zurück. Für die Wahrheitswerte «true» und «false» sind die Textdarstellungen «"true"» und «"false"». |
| toBoolean() : Boolean | Diese Methode gibt diesen Wahrheitswert als «Boolean» zurück. |

Tabelle 7 FEM-Wert «FEMBoolean»

## FEMResult

|  |  |
| --- | --- |
| FEMResult | |
| Diese Klasse implementiert einen Wert, der als Ergebniswert einer Funktion mit *return-by-reference*-Semantik sowie als Parameterwert eines Aufrufs mit *call-by-reference*-Semantik eingesetzt werden kann.  Der Wert kapselt dazu eine gegebene Funktion sowie einen gegebenen Stapelrahmen und wertet diese Funktion erst dann mit dem diesem Stapelrahmen einmalig aus, wenn auf Datentyp oder Nutzdaten zugegriffen wird. Der von der Funktion berechnete Ergebniswert wird zur Wiederverwendung zwischengespeichert. Nach der einmaligen Auswertung der Funktion werden die Verweise auf Stapelrahmen und Funktion aufgelöst. | |
| Mitglied | Beschreibung |
| from(   frame: FEMFrame,   function: FEMFunction) : FEMResult | Diese Methode den Ergebniswert des Aufrufs der gegebenen Funktion mit dem gegebenen Stapelrahmen mit *call-by-reference*-Semantik zurück. Der gelieferte Ergebniswert verzögert die Auswertung der Funktion bis zum ersten Lesen seines Datentyp bzw. seiner Nutzdaten. |
| data() : Object | Diese Methode «TYPE» zurück. |
| type() : FEMType<?> | Diese Methode gibt «this» zurück. |
| result(   recursive: boolean) : FEMValue | Diese Methode gibt das Ergebnis der Auswertung der Funktion mit den Stapelrahmen zurück. Dieser Ergebniswert wird nur beim ersten Aufruf dieser Methode ermittelt und zur Wiederverwendung zwischengespeichert. Dabei werden die Verweise auf Stapelrahmen und Funktion aufgelöst. |
| frame() : FEMFrame | Diese Methode gibt die Stapelrahmen oder «null» zurück. Der erste Aufruf von «result(recursive)» setzt die Stapelrahmen auf «null». |
| function() : FEMFunction | Diese Methode gibt die Funktion oder «null» zurück. Der erste Aufruf von «result(recursive)» setzt die Funktion auf «null». |

Tabelle 8 FEM-Wert «FEMResult»

## FEMNative

|  |  |
| --- | --- |
| FEMNative | |
| Diese Klasse implementiert einen Wert mit einem beliebigen nativen Objekt als Nutzdaten. | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMNative> | Dieses Feld speichert den Datentyp. |
| NULL : FEMNative | Dieses Feld speichert den Wert zu «null». |
| TRUE : FEMNative | Dieses Feld speichert den Wahrheitswert «true». |
| FALSE : FEMNative | Dieses Feld speichert den Wahrheitswert «false». |
| from(   object: Object) : FEMNative | Diese Methode gibt das native Objekt als Wert zurück. Wenn das Objekt bereits ein «FEMNative» ist, wird dieses geliefert. |
| FEMNative(   data: Object) | Dieser Konstruktor initialisiert das native Objekt. |
| data() : Object | Diese Methode gibt die Nutzdaten zurück. |
| type() : FEMType<FEMNative> | Diese Methode «TYPE» zurück. |

Tabelle 9 FEM-Wert «FEMNative»

## FEMDecimal

|  |  |
| --- | --- |
| FEMDecimal | |
| Diese Klasse implementiert einen unveränderlichen Dezimalbruch. Intern wird der Dezimalbruch als «double» dargestellt. | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMDecimal> | Dieses Feld speichert den Datentyp. |
| EMPTY : FEMDecimal | Dieses Feld speichert den Dezimalbruch «NaN». |
| from(   data: double) : FEMDecimal | Diese Methode gibt einen neuen Dezimalbruch mit dem gegebenen Wert zurück. |
| from(   data: Number) : FEMDecimal |
| from(   string: String) : FEMDecimal | Diese Methode gibt einen neuen Dezimalbruch mit dem in der gegebenen Zeichenkette kodierten Wert zurück. Das Format der Zeichenkette entspricht dem der Textdarstellung. |
| from(   value: FEMValue) : FEMDecimal | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMDecimal | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| FEMDecimal(   value: double) | Dieser Konstruktor initialisiert die interne Darstellung des Dezimalbruchs. |
| type() : FEMType<FEMDecimal> | Diese Methode «TYPE» zurück. |
| data() : FEMDecimal | Diese Methode gibt «this» zurück. |
| value() : double | Diese Methode gibt die interne Darstellung des Dezimalbruchs zurück. |
| hash() : int | Diese Methode gibt den Streuwert zurück. |
| equals(   that: FEMDecimal) : boolean | Diese Methode gibt nur dann «true» zurück, wenn dieser Dezimalbruch gleich dem gegebenen ist. |
| compare(   that: FEMDecimal,   undefined: int) : int | Diese Methode gibt «-1», «0» bzw. «+1» zurück, wenn dieser Dezimalbruch kleiner, gleich oder größer als der gegebene Dezimalbruch ist. Wenn die Dezimalbrüche nicht vergleichbar sind, wird «undefined» geliefert. |
| toNumber() : Number | Diese Methode gibt diesen Dezimalbruch als «Number» zurück. |
| toString() : String | Diese Methode gibt die Textdarstellung dieses Dezimalbruchs zurück. |

Tabelle 10 FEM-Wert «FEMDecimal»

## FEMInteger

|  |  |
| --- | --- |
| FEMInteger | |
| Diese Klasse implementiert eine unveränderliche Dezimalzahl. Intern wird die Dezimalzahl als «long» dargestellt. | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMInteger> | Dieses Feld speichert den Datentyp. |
| EMPTY : FEMInteger | Dieses Feld speichert die Dezimalzahl «0». |
| from(   data: long) : FEMInteger | Diese Methode gibt eine neue Dezimalzahl mit dem gegebenen Wert zurück. |
| from(   data: Number) : FEMInteger |
| from(   string: String) : FEMInteger | Diese Methode gibt eine neue Dezimalzahl mit dem in der gegebenen Zeichenkette kodierten Wert zurück. Das Format der Zeichenkette entspricht dem der Textdarstellung. |
| from(   value: FEMValue) : FEMInteger | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMInteger | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| FEMInteger(   value: long) | Dieser Konstruktor initialisiert die interne Darstellung der Dezimalzahl. |
| type() : FEMType<FEMInteger> | Diese Methode «TYPE» zurück. |
| data() : FEMInteger | Diese Methode gibt «this» zurück. |
| value() : long | Diese Methode gibt die interne Darstellung der Dezimalzahl zurück. |
| hash() : int | Diese Methode gibt den Streuwert zurück. |
| equals(   that: FEMDecimal) : boolean | Diese Methode gibt nur dann «true» zurück, wenn diese Dezimalzahl gleich der gegebenen ist. |
| compare(   that: FEMDecimal) : int | Diese Methode gibt «-1», «0» bzw. «+1» zurück, wenn diese Dezimalzahl kleiner, gleich oder größer als die gegebene Dezimalzahl ist. |
| toNumber() : Number | Diese Methode gibt diese Dezimalzahl als «Number» zurück. |
| toString() : String | Diese Methode gibt die Textdarstellung dieser Dezimalzahl zurück. |

Tabelle 11 FEM-Wert «FEMInteger»

## FEMObject

|  |  |
| --- | --- |
| FEMObject | |
| Diese Klasse implementiert eine unveränderliche Referenz auf ein logisches Objekt, welches im Rahmen seines Besitzers über einen Objektschlüssel identifiziert wird. Datentyp und Besitzer des Objekts werden über eine Typkennung bzw. Besitzerkennung angegeben. Die Besitzerkennung kann beispielsweise eine über den «FEMContext» erreichbare Objektliste identifizieren, deren Elemente die referenzierten Objekte darstellen. Der Objektschlüssel könnte hierbei der Position eines Objekts in solch einer Liste entsprechen. Alternativ zur Besitzerkennung könnte hierbei auch die Typkennung genutzt werden. | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMObject> | Dieses Feld speichert den Datentyp. |
| EMPTY : FEMObject | Dieses Feld speichert die Referenz, deren Komponenten alle «0» sind. |
| from(   string: String) : FEMObject | Diese Methode gibt eine neue Referenz mit dem in der gegebenen Zeichenkette kodierten Wert zurück. Das Format der Zeichenkette entspricht dem der Textdarstellung. |
| from(   ref: int,   type: int,   owner: int) : FEMObject | Diese Methode gibt eine neue Referenz mit den gegebenen Eigenschaften zurück. |
| from(   value: FEMValue) : FEMObject | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMObject | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| FEMObject(   value: long) | Dieser Konstruktor initialisiert die interne Darstellung der Referenz. |
| type() : FEMType<FEMObject> | Diese Methode «TYPE» zurück. |
| data() : FEMObject | Diese Methode gibt «this» zurück. |
| value() : long | Diese Methode gibt die interne Darstellung der Referenz zurück. Die 64 Bit von MBS zum LSB sind:  «0» - 1 Bit  «refValue» - 31 Bit  «ownerValue» - 16 Bit  «typeValue» - 16 Bit |
| refValue() : int | Diese Methode gibt den Objektschlüssel zurück («0…2147483647»). |
| typeValue() : int | Diese Methode gibt die Typkennung zurück («0…65535»). |
| ownerValue() : int | Diese Methode gibt die Besitzerkennung zurück («0…65535»). |
| withRef(   ref: int) : FEMObject | Diese Methode gibt diese Referenz mit dem gegebenen Objektschlüssel zurück. |
| withType(   type: int) : FEMObject | Diese Methode gibt diese Referenz mit der gegebenen Typkennung zurück. |
| withOwner(   owner: int) : FEMObject | Diese Methode gibt diese Referenz mit der gegebenen Besitzerkennung zurück. |
| hash() : int | Diese Methode gibt den Streuwert zurück. |
| equals(   that: FEMObject) : boolean | Diese Methode gibt nur dann «true» zurück, wenn diese Referenz gleich der gegebenen ist. |
| compare(   that: FEMObject) : int | Diese Methode gibt «-1», «0» bzw. «+1» zurück, wenn die Ordnung dieser Referenz kleiner, gleich bzw. größer als die der gegebenen Referenz ist. |
| toString() : String | Diese Methode gibt die Textdarstellung dieser Referenz zurück. Das Format der Textdarstellung ist «#REF.TYPE:OWNER». |

Tabelle 12 FEM-Wert «FEMObject»

## FEMHandler

|  |  |
| --- | --- |
| FEMHandler | |
| Diese Klasse implementiert einen unveränderlichen Funktionszeiger, d.h. eine als «FEMValue» verpackte Funktion. Intern wird der Funktionszeiger als «FEMFunction» dargestellt. | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMInteger> | Dieses Feld speichert den Datentyp. |
| from(   function: FEMFunction) : FEMInteger | Diese Methode gibt die gegebene Funktion als Funktionszeiger zurück. |
| from(   value: FEMValue) : FEMInteger | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMInteger | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| FEMHandler(   value: FEMFunction) | Dieser Konstruktor initialisiert die Funktion. |
| type() : FEMType<FEMHandler> | Diese Methode «TYPE» zurück. |
| data() : FEMHandler | Diese Methode gibt «this» zurück. |
| value() : FEMFunction | Diese Methode gibt die Funktion zurück. |
| hash() : int | Diese Methode gibt den Streuwert zurück. |
| equals(   that: FEMHandler) : boolean | Diese Methode gibt nur dann «true» zurück, wenn dieser Funktionszeiger gleich der gegebenen ist. |

Tabelle 13 FEM-Wert «FEMHandler»

## FEMDuration

|  |  |
| --- | --- |
| FEMDuration | |
| Diese Klasse implementiert eine unveränderliche Zeitspanne aus Jahren, Monaten, Tagen, Stunden, Minuten, Sekunden und Millisekunden. Intern wird die Zeitspanne als ein «long» dargestellt. | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMDuration> | Dieses Feld speichert den Datentyp. |
| EMPTY : FEMDuration | Dieses Feld speichert die leere Zeitspanne, deren Komponenten «0» sind. |
| from(   string: String) : FEMDuration | Diese Methode gibt eine neue Zeitspanne mit dem in der gegebenen Zeichenkette kodierten Wert zurück. Das Format der Zeichenkette entspricht dem der Textdarstellung. |
| from(   durationmonths: int,   durationmillis: long) : FEMDuration | Diese Methode gibt eine Zeitspanne mit den gegebenen Gesamtanzahlen an Monaten und Millisekunden zurück. |
| from(   years: int,   months: int,   days: int,   hours: int,   minutes: long,   seconds: long,   milliseconds: long) : FEMDuration | Diese Methode gibt eine Zeitspanne mit den Gesamtanzahlen an Monaten und Millisekunden zurück, die sich aus den gegebenen Anzahlen ergeben. |
| from(   value: FEMValue) : FEMDuration | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMDuration | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| between(   datetime1: FEMDatetime,   datetime2: FEMDatetime) : FEMDuration | Diese Methode gibt die Zeitspanne zwischen den gegebenen Zeitangaben in Zeitzone «00:00» zurück.  Wenn nur eine der Zeitangaben ein Datum bzw. eine Uhrzeit besitzt, wird eine «IllegalArgumentException» ausgelöst. |
| minLengthOf(   months: int) : int | Diese Methode gibt die minimale Anzahl an Tagen zurück, die durch die gegebene Anzahl an Monaten ausgedrückt werden kann. |
| maxLengthOf(   months: int) : int | Diese Methode gibt die maximale Anzahl an Tagen zurück, die durch die gegebene Anzahl an Monaten ausgedrückt werden kann. |
| durationmillisOf(   days: int,   hours: int,   minutes: long,   seconds: long,   milliseconds: long) : long | Diese Methode gibt die Gesamtanzahl der Millisekunden zurück, die sich aus den gegebenen Anzahlen ergeben («-265621593600000…265621593600000»). |
| durationmonthsOf(   years: int,   months: int) : int | Diese Methode gibt die Gesamtanzahl der Monate zurück, die sich aus den gegebenen Anzahlen ergeben («-101006…101006»). |
| FEMDuration(   value: long) | Dieser Konstruktor initialisiert die interne Darstellung der Zeitspanne. |
| type() : FEMType<FEMDuration> | Diese Methode «TYPE» zurück. |
| data() : FEMDuration | Diese Methode gibt «this» zurück. |
| value() : long | Diese Methode gibt die interne Darstellung der Zeitspanne zurück. Die 64 Bit von MBS zum LSB sind:  «yearsValue» - 14 Bit  «signValue» - 1 Bit  «hoursValue» - 5 Bit  «minutesValue» - 6 Bit  «secondsValue» - 6 Bit  «daysValue» - 18 Bit  «monthsValue» - 4 Bit  «millisecondsValue» - 10 Bit |
| signValue() : int | Diese Methode gibt das Vorzeichen dieser Zeitspanne zurück. Das Vorzeichen ist «-1», «0» oder «+1», wenn alle Komponenten der Zeitspanne kleiner als, gleich bzw. größer als «0» sind. |
| yearsValue() : int | Diese Methode gibt die Anzahl der Jahre zurück («0…8417»). |
| monthsValue() : int | Diese Methode gibt die Anzahl der Monate zurück («0…11»). |
| daysValue() : int | Diese Methode gibt die Anzahl der Tage zurück («0…146096»). |
| hoursValue() : int | Diese Methode gibt die Anzahl der Stunden zurück («0…23»). |
| minutesValue() : int | Diese Methode gibt die Anzahl der Minuten zurück («0…59»). |
| secondsValue() : int | Diese Methode gibt die Anzahl der Sekunden zurück («0…59»). |
| millisecondsValue() : int | Diese Methode gibt die Anzahl der Millisekunden zurück («0…999»). |
| durationmillisValue() : long | Diese Methode gibt die Gesamtanzahl der Millisekunden zurück. Diese fassen «daysValue», «hoursValue», «minutesValue», «secondsValue» und «millisecondsValue» zusammen. |
| durationmonthsValue() : int | Diese Methode gibt die Gesamtanzahl der Monate zurück. Diese fassen «yearsValue» und «monthsValue» zusammen. |
| negate() : FEMDuration | Diese Methode gibt diese Zeitspanne mit umgekehrten Vorzeichen zurück. |
| move(   durationmonths: int,   durationmillis: long) : FEMDuration | Diese Methode gibt diese Zeitspanne verschoben um die gegebenen Gesamtanzahlen an Monate und Millisekunden zurück. |
| move(   years: int,   months: int,   days: int,   hours: int,   minutes: long,   seconds: long,   milliseconds: long) : FEMDuration | Diese Methode gibt diese Zeitspanne verschoben um die Gesamtanzahlen an Monaten und Millisekunden zurück, die sich aus den gegebenen Anzahlen ergeben. |
| move(   duration: FEMDuration)  : FEMDuration | Diese Methode gibt diese Zeitspanne verschoben um die Gesamtanzahlen an Monate und Millisekunden der gegebenen Zeitspanne zurück. |
| hash() : int | Diese Methode gibt den Streuwert zurück. |
| equals(   that: FEMDuration) : boolean | Diese Methode gibt nur dann «true» zurück, wenn diese Zeitspanne effektiv gleich der gegebenen ist. |
| compare(   that: FEMDuration,   undefined: int) : int | Diese Methode gibt «-1», «0» bzw. «+1» zurück, wenn diese Zeitspanne kürzer, gleich bzw. länger als die gegebene Zeitspanne ist. Wenn die Zeitspannen nicht vergleichbar sind, wird «undefined» geliefert. |
| toString() : String | Diese Methode gibt die Textdarstellung dieser Zeitspanne zurück. Diese Textdarstellung entspricht der des Datentyps «xsd:duration» aus «XML Schema Part 2: Datatypes Second Edition», beschränkt auf maximal drei Nachkommastellen für die Sekunden. |

Tabelle 14 FEMDuration

## FEMDatetime

|  |  |
| --- | --- |
| FEMDatetime | |
| Diese Klasse implementiert eine Zeitangabe mit Datum, Uhrzeit und/oder Zeitzone im Gregorianischen Kalender. Intern wird die Zeitangabe als ein «long» dargestellt.  Das Datum kann unspezifiziert sein oder aus Jahr, Monat sowie Tag bestehen und im Bereich «15.10.1582…31.12.9999» liegen. Die Uhrzeit kann unspezifiziert sein oder aus Stunde, Minute, Sekunde sowie Millisekunde bestehen und im Bereich «00:00:00.000…24:00:00.000» liegen. Die Zeitzone kann unspezifiziert sein oder aus Stunde sowie Minute bestehen und im Bereich «-14:00…+14:00» liegen.  Jahr  Der Zahlenwert für das Jahr entspricht der Anzahl der Jahre seit dem Beginn des Gregorianischen Kalenders erhöht um «1582». Unterstützte Zahlenwerte für das Jahr sind «1582…9999». Ein reguläres Jahr hat «365» Tage, ein Schaltjahr hat «366» Tage.  Monat  Der Zahlenwert für den Monat entspricht der Anzahl der Monate seit Beginn des Jahres erhöht um «1». Unterstützte Zahlenwerte für den Monat sind «1…12».  Januar = «Calendar.JANUARY + 1»  Februar = «Calendar.FEBRUARY + 1»  März = «Calendar.MARCH + 1»  April = «Calendar.APRIL + 1»  Mai = «Calendar.MAY + 1»  Juni = «Calendar.JUNE + 1»  Juli = «Calendar.JULY + 1»  August = «Calendar.AUGUST + 1»  September = «Calendar.SEPTEMBER + 1»  Oktober = «Calendar.OCTOBER + 1»  November = «Calendar.NOVEMBER + 1»  Dezember = «Calendar.DECEMBER + 1»  Tag (Tag in einem Monat)  Der Zahlenwert für den Tag entspricht der Anzahl der Tage seit Beginn des Monats erhöht um «1». Unterstützte Zahlenwerte für den Monat sind «1…31», wobei einige Monate auch abhängig von Schaltjahren geringere Obergrenzen besitzen.  Jahrestag (Tag in einem Jahr)  Der Zahlenwert für den Jahrestag entspricht der Anzahl der Tage seit dem Beginn des Jahres erhöht um «1». Unterstützte Zahlenwerte für den Jahrestag sind «1…366».  Wochentag (Tag in einer Woche)  Der Zahlenwert für den Wochentag entspricht der Anzahl der Tage seit Beginn der Woche erhöht um «1». Unterstützte Zahlenwerte für den Wochentag sind «1…12».  Sonntag = «Calendar.SUNDAY»  Montag = «Calendar.MONDAY»  Dienstag = «Calendar.TUESDAY»  Mittwoch = «Calendar.WEDNESDAY»  Donnerstag = «Calendar.THURSDAY»  Freitag = «Calendar.FRIDAY»  Samstag = «Calendar.SATURDAY»  Kalendertag (Tag im Kalender)  Der Zahlenwert für den Kalendertag entspricht der Anzahl der Tage seit dem Beginn des Gregorianischen Kalenders am Freitag dem «15.10.1582». Unterstützte Zahlenwerte für den Kalendertag sind «0…3074323».  Tagesmillis (Millisekunden am Tag)  Der Zahlenwert für die Tagesmillis entspricht der Anzahl der Millisekunden seit «00:00:00.000». Unterstützte Zahlenwerte für die Tagesmillis sind «0…86400000».  Zeitzone (Zeitzonenverschiebung)  Der Zahlenwert für die Zeitzone entspricht der Zeitzonenverschiebung gegenüber UTC in Minuten. Unterstützte Zahlenwerte für die Zeitzone sind «-840…840». | |
| Mitglied | Beschreibung |
| ID : int | Dieses Feld speichert den Identifikator von «TYPE». |
| TYPE : FEMType<FEMDatetime> | Dieses Feld speichert den Datentyp. |
| EMPTY : FEMDatetime | Dieses Feld speichert die leere Zeitangabe ohne Datum, ohne Uhrzeit und ohne Zeitzone. |
| now() : FEMDatetime | Diese Methode gibt den aktuellen Zeitpunkt zurück. |
| from(   millis: long) : FEMDatetime | Diese Methode gibt einen Zeitpunkt zurück, der die gegebene Anzahl an Millisekunden nach dem Zeitpunkt «1970-01-01T00:00:00Z» liegt. |
| from(   string: String) : FEMDatetime | Diese Methode gibt eine Zeitangabe mit den in der gegebenen Zeitangabe kodierten Komponenten zurück. Das Format der Zeichenkette entspricht dem der Textdarstellung. |
| from(   calendar: Calendar) : FEMDatetime | Diese Methode gibt eine Zeitangabe mit dem Datum, der Uhrzeit und der Zeitzone des gegebenen «Calendar» zurück und ist eine Abkürzung für «FEE\_Datetime.EMPTY.withDate(calendar). withTime(calendar).withZone(calendar)». |
| from(   value: FEMValue) : FEMDatetime | Diese Methode ist eine Abkürzung für «FEMContext.DEFAULT().dataFrom(value, TYPE)». |
| from(   value: FEMValue,   context: FEMContext) : FEMDatetime | Diese Methode ist eine Abkürzung für «context.dataFrom(value, TYPE)». |
| fromDate(   calendarday: int)  : FEMDatetime | Diese Methode gibt eine Zeitangabe mit dem Datum zum gegebenen Kalendertag zurück und ist eine Abkürzung für «FEE\_Datetime.EMPTY.withDate(calendarday)». |
| fromDate(   year: int,   month: int,   date: int)  : FEMDatetime | Diese Methode gibt eine Zeitangabe mit dem gegebenen Datum zurück und ist eine Abkürzung für «FEE\_Datetime.EMPTY.withDate(year, month, date)». |
| fromTime(   daymillis: int)  : FEMDatetime | Diese Methode gibt eine Zeitangabe mit der Uhrzeit zu den gegebenen Tagesmillis zurück und ist eine Abkürzung für «FEE\_Datetime.EMPTY.withTime(daymillis)». |
| fromTime(   hour: int,   minute: int,   second: int,   millisecond: int)  : FEMDatetime | Diese Methode gibt eine Zeitangabe mit der gegebenen Uhrzeit zurück und ist eine Abkürzung für «FEE\_Datetime.EMPTY.withTime(hour, minute, second, millisecond)». |
| fromZone(   zone: int)  : FEMDatetime | Diese Methode gibt eine Zeitangabe mit der gegebenen Zeitzone zurück und ist eine Abkürzung für «FEE\_Datetime.EMPTY.withZone(zone)». |
| fromZone(   zoneHour: int,   zoneMinute: int)  : FEMDatetime | Diese Methode gibt eine Zeitangabe mit der gegebenen Zeitzone zurück und ist eine Abkürzung für «FEE\_Datetime.EMPTY.withZone(zoneHour, zoneMinute)». |
| leapOf(   year: int) : boolean | Diese Methode gibt nur dann «true» zurück, wenn das gegebene Jahr ein Schaltjahr im Gregorianischen Kalender ist. |
| lengthOf(   month: int,   year: int) : int | Diese Methode gibt die Länge des gegebenen Monats im gegebenen Jahr zurück, d.h. die Anzahl der Tage im Monat. |
| lengthOf(   month: int,   leap: boolean) : int | Diese Methode gibt die Länge des gegebenen Monats zurück, d.h. die Anzahl der Tage im Monat. |
| yeardayOf(   calendarday: int) : int | Diese Methode gibt den Jahrestag zum gegebenen Kalendertag zurück («1…366»). |
| weekdayOf(   calendarday: int) : int | Diese Methode gibt den Wochentag zum gegebenen Kalendertag zurück («1…7»). |
| daymillisOf(   hour: int,   minute: int,   second: int,   millisecond: int) : int | Diese Methode gibt die Tagesmillis zur gegebenen Uhrzeit zurück («0…86400000»). |
| calendardayOf(   year: int,   month: int,   date: int) : int | Diese Methode gibt den Kalendertag zum gegebenen Datum zurück («0…3074323»). |
| FEMDatetime(   value: long) | Dieser Konstruktor initialisiert die interne Darstellung. |
| type() : FEMType<FEMDatetime> | Diese Methode «TYPE» zurück. |
| data() : FEMDatetime | Diese Methode gibt «this» zurück. |
| value() : long | Diese Methode gibt die interne Darstellung der Zeitangabe zurück. Die 64 Bit von MBS zum LSB sind:  «yearValue» - 14 Bit  «monthValue» - 4 Bit  «minuteValue» - 6 Bit  «secondValue» - 6 Bit  «hasDate» - 1 Bit  «hasTime» - 1 Bit  «hasZone» - 1 Bit  «zoneValue» - 11 Bit  «dateValue» - 5 Bit  «hourValue» - 5 Bit  «millisecondValue» - 10 Bit |
| yearValue() : int | Diese Methode gibt das Jahr zurück («1582…9999»). |
| dateValue() : int | Diese Methode gibt den Tag zurück («1…31»). |
| monthValue() : int | Diese Methode gibt den Monat zurück («1…12»). |
| hourValue() : int | Diese Methode gibt die Stunde zurück («0…24»). |
| minuteValue() : int | Diese Methode gibt die Minute zurück («0…59»). |
| secondValue() : int | Diese Methode gibt die Sekunde zurück («0…59»). |
| millisecondValue() : int | Diese Methode gibt die Millisekunde zurück («0…999»). |
| zoneValue() : int | Diese Methode gibt die Zeitzonenverschiebung zur UTC in Minuten zurück («-840…840»). |
| hasDate() : boolean | Diese Methode gibt nur dann «true» zurück, wenn diese Zeitangabe ein Datum besitzt. |
| hasTime() : boolean | Diese Methode gibt nur dann «true» zurück, wenn diese Zeitangabe eine Uhrzeit besitzt. |
| hasZone() : boolean | Diese Methode gibt nur dann «true» zurück, wenn diese Zeitangabe eine Zeitzone besitzt. |
| yeardayValue() : int | Diese Methode gibt den Jahrestag zurück («1…366»). |
| weekdayValue() : int | Diese Methode gibt den Wochentag zurück («1…7»). |
| daymillisValue() : int | Diese Methode gibt die Tagesmillis zurück («0…86400000»). |
| calendardayValue() : int | Diese Methode gibt den Kalendertag zurück («0…3074323»). |
| withDate(int) |  |
| withDate(int, int, int) |  |
| withDate(Calendar) |  |
| withDate(FEMDatetime) |  |
| withTime(int) |  |
| withTime(int, int, int, int) |  |
| withTime(Calendar) |  |
| withTime(FEMDatetime) |  |
| withZone(int) |  |
| withZone(int, int) |  |
| withZone(Calendar) |  |
| withZone(FEMDatetime) |  |
| withoutDate() |  |
| withoutTime() |  |
| withoutZone() |  |
| move(FEMDuration) |  |
| moveDate(int, int, int) |  |
| moveDate(FEMDuration) |  |
| moveTime(int, int, int, int) |  |
| moveTime(int, long, long, long) |  |
| moveTime(FEMDuration) |  |
| moveZone(int, int) |  |
| hash() |  |
| equals(FEMDatetime) |  |
| compare(   that: FEMDatetime,   undefined: int) : int |  |
| toCalendar() : GregorianCalendar | Diese Methode gibt diese Zeitangabe als «Calendar» zurück. |
| toString() : String | Diese Methode gibt die Textdarstellung dieser Zeitangabe zurück. Diese Textdarstellung entspricht der des Datentyps «xs:dateTime» aus «XML Schema Part 2: Datatypes Second Edition», beschränkt auf maximal drei Nachkommastellen für die Sekunden. |

Tabelle 16

|  |  |
| --- | --- |
|  | |
|  | |
| Mitglied | Beschreibung |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Tabelle 17

|  |  |
| --- | --- |
|  | |
|  | |
| Mitglied | Beschreibung |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Tabelle 18

# FEM-Funktionen

Mitgelieferte funktionen für operationen

TODO

# FEM-Datenformat

Auf IAM aufbauende Kodierung von Werten, Funktionen usw

TODO

# FEM-Skriptsprache

TODO