

Module `java.base`**Package** `java.lang.constant`

Interface `MethodTypeDesc`

All Superinterfaces:`ConstantDesc, TypeDescriptor, TypeDescriptor.OfMethod<ClassDesc, MethodTypeDesc>`

```
public sealed interface MethodTypeDesc
extends ConstantDesc, TypeDescriptor.OfMethod<ClassDesc, MethodTypeDesc>
```

A nominal descriptor for a `MethodType` constant.

Since:

12

Nested Class Summary

Nested classes/interfaces declared in interface `java.lang.invoke.TypeDescriptor`

```
TypeDescriptor.OfField<F extends TypeDescriptor.OfField<F>>, TypeDescriptor.OfMethod<F extends
TypeDescriptor.OfField<F>, M extends TypeDescriptor.OfMethod<F, M>>
```

Method Summary

All Methods

Static Methods

Instance Methods

Abstract Methods

Default Methods

Modifier and Type

Method

Description

MethodTypeDesc

changeParameterType(int index,
ClassDesc paramType)

Returns a `MethodTypeDesc` that is identical to this one, except that a single parameter type has been

changed to the specified type.

MethodTypeDesc	changeReturnType (ClassDesc returnType)	Returns a MethodTypeDesc that is identical to this one, except with the specified return type.
String	descriptorString ()	Returns the method type descriptor string.
default String	displayDescriptor ()	Returns a human-readable descriptor for this method type, using the canonical names for parameter and return types.
MethodTypeDesc	dropParameterTypes (int start, int end)	Returns a MethodTypeDesc that is identical to this one, except that a range of parameter types have been removed.
boolean	equals (Object o)	Compares the specified object with this descriptor for equality.
MethodTypeDesc	insertParameterTypes (int pos, ClassDesc ... paramTypes)	Returns a MethodTypeDesc that is identical to this one, except that a range of additional parameter types have been inserted.
static MethodTypeDesc	of (ClassDesc returnDesc)	Returns a MethodTypeDesc with the given return type and no parameter types.
static MethodTypeDesc	of (ClassDesc returnDesc, ClassDesc ... paramDescs)	Returns a MethodTypeDesc given the return type and parameter types.
static MethodTypeDesc	of (ClassDesc returnDesc, List < ClassDesc > paramDescs)	Returns a MethodTypeDesc given the return type and a list of parameter types.
static MethodTypeDesc	ofDescriptor (String descriptor)	Creates a MethodTypeDesc given a method descriptor string.
ClassDesc []	parameterArray ()	Returns the parameter types as an array.
int	parameterCount ()	Returns the number of parameters of the method type described by this MethodTypeDesc .

List<ClassDesc>	parameterList()	Returns the parameter types as an immutable List .
ClassDesc	parameterType(int index)	Returns the parameter type of the index'th parameter of the method type described by this MethodTypeDesc .
MethodType	resolveConstantDesc (MethodHandles.Lookup lookup)	Resolves this descriptor reflectively, emulating the resolution behavior of JVMS 5.4.3 and the access control behavior of JVMS 5.4.4 .
ClassDesc	returnType()	Gets the return type of the method type described by this MethodTypeDesc .

Method Details

ofDescriptor

static [MethodTypeDesc](#) ofDescriptor([String](#) descriptor)

Creates a [MethodTypeDesc](#) given a method descriptor string.

Parameters:

descriptor - a method descriptor string

Returns:

a [MethodTypeDesc](#) describing the desired method type

Throws:

[NullPointerException](#) - if the argument is null

[IllegalArgumentException](#) - if the descriptor string is not a valid method descriptor

See *Java Virtual Machine Specification*:

[4.3.3 Method Descriptors](#)

of

```
static MethodTypeDesc of(ClassDesc returnDesc)
```

Returns a [MethodTypeDesc](#) with the given return type and no parameter types.

Parameters:

returnDesc - a [ClassDesc](#) describing the return type

Returns:

a [MethodTypeDesc](#) with the given return type and no parameter types

Throws:

[NullPointerException](#) - if returnDesc is null

Since:

21

of

```
static MethodTypeDesc of(ClassDesc returnDesc,  
                        List<ClassDesc> paramDescs)
```

Returns a [MethodTypeDesc](#) given the return type and a list of parameter types.

Parameters:

returnDesc - a [ClassDesc](#) describing the return type

paramDescs - a [List](#) of [ClassDescs](#) describing the parameter types

Returns:

a [MethodTypeDesc](#) given the return type and a list of parameter types

Throws:

[NullPointerException](#) - if any argument or its contents are null

[IllegalArgumentException](#) - if any element of paramDescs is a [ClassDesc](#) for void

Since:

of

```
static MethodTypeDesc of(ClassDesc returnDesc,  
                        ClassDesc... paramDescs)
```

Returns a `MethodTypeDesc` given the return type and parameter types.

Parameters:

`returnDesc` - a `ClassDesc` describing the return type

`paramDescs` - `ClassDescs` describing the argument types

Returns:

a `MethodTypeDesc` describing the desired method type

Throws:

`NullPointerException` - if any argument or its contents are null

`IllegalArgumentException` - if any element of `paramDescs` is a `ClassDesc` for void

returnType

```
ClassDesc returnType()
```

Gets the return type of the method type described by this `MethodTypeDesc`.

Specified by:

`returnType` in interface `TypeDescriptor.OfMethod<ClassDesc,MethodTypeDesc>`

Returns:

a `ClassDesc` describing the return type of the method type

parameterCount

```
int parameterCount()
```

Returns the number of parameters of the method type described by this `MethodTypeDesc`.

Specified by:

`parameterCount` in interface `TypeDescriptor.OfMethod<ClassDesc,MethodTypeDesc>`

Returns:

the number of parameters

parameterType

```
ClassDesc parameterType(int index)
```

Returns the parameter type of the `index`'th parameter of the method type described by this `MethodTypeDesc`.

Specified by:

`parameterType` in interface `TypeDescriptor.OfMethod<ClassDesc,MethodTypeDesc>`

Parameters:

`index` - the index of the parameter to retrieve

Returns:

a `ClassDesc` describing the desired parameter type

Throws:

`IndexOutOfBoundsException` - if the index is outside the half-open range `[0, parameterCount())`

parameterList

```
List<ClassDesc> parameterList()
```

Returns the parameter types as an immutable `List`.

Specified by:

`parameterList` in interface `TypeDescriptor.OfMethod<ClassDesc,MethodTypeDesc>`

Returns:

a [List](#) of [ClassDesc](#) describing the parameter types

parameterArray

```
ClassDesc[] parameterArray()
```

Returns the parameter types as an array.

Specified by:

[parameterArray](#) in interface [TypeDescriptor.OfMethod<ClassDesc,MethodTypeDesc>](#)

Returns:

an array of [ClassDesc](#) describing the parameter types

changeReturnType

```
MethodTypeDesc changeReturnType(ClassDesc returnType)
```

Returns a [MethodTypeDesc](#) that is identical to this one, except with the specified return type.

Specified by:

[changeReturnType](#) in interface [TypeDescriptor.OfMethod<ClassDesc,MethodTypeDesc>](#)

Parameters:

[returnType](#) - a [ClassDesc](#) describing the new return type

Returns:

a [MethodTypeDesc](#) describing the desired method type

Throws:

[NullPointerException](#) - if the argument is null

changeParameterType

```
MethodTypeDesc changeParameterType(int index,  
                                   ClassDesc paramType)
```

Returns a `MethodTypeDesc` that is identical to this one, except that a single parameter type has been changed to the specified type.

Specified by:

`changeParameterType` in interface `TypeDescriptor.OfMethod<ClassDesc,MethodTypeDesc>`

Parameters:

`index` - the index of the parameter to change

`paramType` - a `ClassDesc` describing the new parameter type

Returns:

a `MethodTypeDesc` describing the desired method type

Throws:

`NullPointerException` - if any argument is null

`IndexOutOfBoundsException` - if the index is outside the half-open range `[0, parameterCount)`

dropParameterTypes

```
MethodTypeDesc dropParameterTypes(int start,  
                                   int end)
```

Returns a `MethodTypeDesc` that is identical to this one, except that a range of parameter types have been removed.

Specified by:

`dropParameterTypes` in interface `TypeDescriptor.OfMethod<ClassDesc,MethodTypeDesc>`

Parameters:

`start` - the index of the first parameter to remove

`end` - the index after the last parameter to remove

Returns:

a `MethodTypeDesc` describing the desired method type

Throws:

[IndexOutOfBoundsException](#) - if start is outside the half-open range [0, parameterCount), or end is outside the closed range [0, parameterCount], or if start > end

insertParameterTypes

```
MethodTypeDesc insertParameterTypes(int pos,  
                                     ClassDesc... paramTypes)
```

Returns a [MethodTypeDesc](#) that is identical to this one, except that a range of additional parameter types have been inserted.

Specified by:

[insertParameterTypes](#) in interface [TypeDescriptor.OfMethod<ClassDesc,MethodTypeDesc>](#)

Parameters:

pos - the index at which to insert the first inserted parameter

paramTypes - [ClassDescs](#) describing the new parameter types to insert

Returns:

a [MethodTypeDesc](#) describing the desired method type

Throws:

[NullPointerException](#) - if any argument or its contents are null

[IndexOutOfBoundsException](#) - if pos is outside the closed range [0, parameterCount]

[IllegalArgumentException](#) - if any element of paramTypes is a [ClassDesc](#) for void

descriptorString

```
String descriptorString()
```

Returns the method type descriptor string.

Specified by:

[descriptorString](#) in interface [TypeDescriptor](#)

Returns:

the method type descriptor string

See *Java Virtual Machine Specification*:

[4.3.3 Method Descriptors](#) 

displayDescriptor

```
default String displayDescriptor()
```



Returns a human-readable descriptor for this method type, using the canonical names for parameter and return types.

Returns:

the human-readable descriptor for this method type

resolveConstantDesc

```
MethodType resolveConstantDesc(MethodHandles.Lookup lookup)  
            throws ReflectiveOperationException
```

Resolves this descriptor reflectively, emulating the resolution behavior of JVM 5.4.3  and the access control behavior of JVM 5.4.4 . The resolution and access control context is provided by the `MethodHandles.Lookup` parameter. No caching of the resulting value is performed.

Specified by:

`resolveConstantDesc` in interface `ConstantDesc`

API Note:

`MethodTypeDesc` can represent method type descriptors that are not representable by `MethodType`, such as methods with more than 255 parameter slots, so attempts to resolve these may result in errors.

Parameters:

`lookup` - The `MethodHandles.Lookup` to provide name resolution and access control context

Returns:

the resolved constant value

Throws:

[ReflectiveOperationException](#) - if a class, method, or field could not be reflectively resolved in the course of resolution

equals

```
boolean equals(Object o)
```

Compares the specified object with this descriptor for equality. Returns `true` if and only if the specified object is also a [MethodTypeDesc](#) both have the same arity, their return types are equal, and each pair of corresponding parameter types are equal.

Overrides:

`equals` in class [Object](#)

Parameters:

`o` - the other object

Returns:

whether this descriptor is equal to the other object

See Also:

[Object.hashCode\(\)](#), [HashMap](#)

[Report a bug or suggest an enhancement](#)

For further API reference and developer documentation see the [Java SE Documentation](#), which contains more detailed, developer-targeted descriptions with conceptual overviews, definitions of terms, workarounds, and working code examples. [Other versions](#).

Java is a trademark or registered trademark of Oracle and/or its affiliates in the US and other countries.

Copyright © 1993, 2024, Oracle and/or its affiliates, 500 Oracle Parkway, Redwood Shores, CA 94065 USA.

All rights reserved. Use is subject to [license terms](#) and the [documentation redistribution policy](#). [Modify Cookie Preferences](#). [Modify Ad Choices](#).