

**Module** `jdk.incubator.foreign`  
**Package** `jdk.incubator.foreign`

## Class `FunctionDescriptor`

`java.lang.Object`  
`jdk.incubator.foreign.FunctionDescriptor`

**All Implemented Interfaces:**  
`Constable`

`public sealed class FunctionDescriptor`  
`extends Object`  
`implements Constable`

A function descriptor is made up of zero or more argument layouts and zero or one return layout. A function descriptor is used to model the signature of foreign functions.

Unless otherwise specified, passing a `null` argument, or an array argument containing one or more `null` elements to a method in this class causes a `NullPointerException` to be thrown.

### Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type		Method	Description
<code>FunctionDescriptor</code>		<code>appendArgumentLayouts</code> ( <code>MemoryLayout</code> ... <code>addedLayouts</code> )	Create a new function descriptor with the given argument layouts appended to the argument layout array of this function descriptor.
<code>List&lt;MemoryLayout&gt;</code>		<code>argumentLayouts()</code>	Returns the argument layouts associated with this function descriptor.
<code>FunctionDescriptor</code>		<code>asVariadic</code> ( <code>MemoryLayout</code> ... <code>variadicLayouts</code> )	Obtain a specialized variadic function descriptor, by appending given variadic layouts to this function descriptor argument layouts.
<code>FunctionDescriptor</code>		<code>changeReturnLayout</code> ( <code>MemoryLayout</code> <code>newReturn</code> )	Create a new function descriptor with the given memory layout as the new return layout.
<code>Optional&lt;DynamicConstantDesc&lt;FunctionDescriptor&gt;&gt;</code>		<code>describeConstable()</code>	Returns an <code>Optional</code> containing the nominal descriptor for this function descriptor, if one can be constructed, or an empty <code>Optional</code> if one cannot be constructed.
<code>FunctionDescriptor</code>		<code>dropReturnLayout()</code>	Create a new function descriptor with the return layout dropped.
<code>boolean</code>		<code>equals(Object other)</code>	Compares the specified object with this function descriptor for equality.
<code>int</code>		<code>firstVariadicArgumentIndex()</code>	The index of the first variadic argument layout (where defined).
<code>int</code>		<code>hashCode()</code>	Returns the hash code value for this function descriptor.
<code>FunctionDescriptor</code>		<code>insertArgumentLayouts</code> ( <code>int</code> <code>index</code> , <code>MemoryLayout</code> ... <code>addedLayouts</code> )	Create a new function descriptor with the given argument layouts inserted at the given index, into the argument layout array of this function descriptor.
<code>static FunctionDescriptor</code>		<code>of</code> ( <code>MemoryLayout</code> <code>resLayout</code> , <code>MemoryLayout</code> ... <code>argLayouts</code> )	Create a function descriptor with given return and argument layouts.
<code>static FunctionDescriptor</code>		<code>ofVoid</code> ( <code>MemoryLayout</code> ... <code>argLayouts</code> )	Create a function descriptor with given argument layouts and no return layout.
<code>Optional&lt;MemoryLayout&gt;</code>		<code>returnLayout()</code>	Returns the return layout (if any) associated with this function descriptor.
<code>String</code>		<code>toString()</code>	Returns the string representation of this function descriptor.

Methods declared in class java.lang.Object

clone, finalize, getClass, notify, notifyAll, wait, wait, wait

Method Details

returnLayout

```
public Optional<MemoryLayout> returnLayout()
```

Returns the return layout (if any) associated with this function descriptor.

**Returns:**  
the return layout (if any) associated with this function descriptor

argumentLayouts

```
public List<MemoryLayout> argumentLayouts()
```

Returns the argument layouts associated with this function descriptor..

**Returns:**  
the argument layouts associated with this function descriptor

of

```
public static FunctionDescriptor of(MemoryLayout resLayout,
                                   MemoryLayout... argLayouts)
```

Create a function descriptor with given return and argument layouts.

**Parameters:**  
resLayout - the return layout.  
argLayouts - the argument layouts.

**Returns:**  
the new function descriptor.

ofVoid

```
public static FunctionDescriptor ofVoid(MemoryLayout... argLayouts)
```

Create a function descriptor with given argument layouts and no return layout.

**Parameters:**  
argLayouts - the argument layouts.

**Returns:**  
the new function descriptor.

asVariadic

```
public FunctionDescriptor asVariadic(MemoryLayout... variadicLayouts)
```

Obtain a specialized variadic function descriptor, by appending given variadic layouts to this function descriptor argument layouts. The resulting function descriptor can report the position of the [first variadic argument](#), and cannot be altered in any way: for instance, calling `changeReturnLayout(MemoryLayout)` on the resulting descriptor will throw an `UnsupportedOperationException`.

**Parameters:**  
variadicLayouts - the variadic argument layouts to be appended to this descriptor argument layouts.

**Returns:**  
a new variadic function descriptor, or this descriptor if `variadicLayouts.length == 0`.

firstVariadicArgumentIndex

```
public int firstVariadicArgumentIndex()
```

The index of the first variadic argument layout (where defined).

**Returns:**

The index of the first variadic argument layout, or -1 if this is not a [variadic](#) layout.

## appendArgumentLayouts

```
public FunctionDescriptor appendArgumentLayouts(MemoryLayout... addedLayouts)
```

Create a new function descriptor with the given argument layouts appended to the argument layout array of this function descriptor.

**Parameters:**

addedLayouts - the argument layouts to append.

**Returns:**

the new function descriptor.

## insertArgumentLayouts

```
public FunctionDescriptor insertArgumentLayouts(int index,
                                                MemoryLayout... addedLayouts)
```

Create a new function descriptor with the given argument layouts inserted at the given index, into the argument layout array of this function descriptor.

**Parameters:**

index - the index at which to insert the arguments

addedLayouts - the argument layouts to insert at given index.

**Returns:**

the new function descriptor.

**Throws:**

[IllegalArgumentException](#) - if `index < 0 || index > argumentLayouts().size()`.

## changeReturnLayout

```
public FunctionDescriptor changeReturnLayout(MemoryLayout newReturn)
```

Create a new function descriptor with the given memory layout as the new return layout.

**Parameters:**

newReturn - the new return layout.

**Returns:**

the new function descriptor.

## dropReturnLayout

```
public FunctionDescriptor dropReturnLayout()
```

Create a new function descriptor with the return layout dropped. This is useful to model functions which return no values.

**Returns:**

the new function descriptor.

## toString

```
public String toString()
```

Returns the string representation of this function descriptor.

**Overrides:**

[toString](#) in class [Object](#)

**Returns:**

the string representation of this function descriptor

## equals

```
public boolean equals(Object other)
```

Compares the specified object with this function descriptor for equality. Returns `true` if and only if the specified object is also a function descriptor, and all the following conditions are met:

- the two function descriptors have equals return layouts (see [MemoryLayout.equals\(Object\)](#)), or both have no return layout

- the two function descriptors have argument layouts that are pair-wise equal (see `MemoryLayout.equals(Object)`)

**Overrides:**

`equals` in class `Object`

**Parameters:**

`other` - the object to be compared for equality with this function descriptor.

**Returns:**

`true` if the specified object is equal to this function descriptor.

**See Also:**

`Object.hashCode()`, `HashMap`

**hashCode**

```
public int hashCode()
```

Returns the hash code value for this function descriptor.

**Overrides:**

`hashCode` in class `Object`

**Returns:**

the hash code value for this function descriptor

**See Also:**

`Object.equals(java.lang.Object)`,  
`System.identityHashCode(java.lang.Object)`

**describeConstable**

```
public Optional<DynamicConstantDesc<FunctionDescriptor>> describeConstable()
```

Returns an `Optional` containing the nominal descriptor for this function descriptor, if one can be constructed, or an empty `Optional` if one cannot be constructed.

**Specified by:**

`describeConstable` in interface `Constable`

**Returns:**

An `Optional` containing the resulting nominal descriptor, or an empty `Optional` if one cannot be constructed.

[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, 2022, 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](#).