Module java.base
Package java.util.function

Interface BinaryOperator<T>

Type Parameters:

T - the type of the operands and result of the operator

All Superinterfaces:

`BiFunction<T,T,T>`

Functional Interface:

This is a functional interface and can therefore be used as the assignment target for a lambda expression or method reference.

```
@FunctionalInterface
public interface BinaryOperator<T>
extends BiFunction<T,T,T>
```

Represents an operation upon two operands of the same type, producing a result of the same type as the operands. This is a specialization of `BiFunction` for the case where the operands and the result are all of the same type.

This is a functional interface whose functional method is `BiFunction.apply(Object, Object)`.

Since:

1.8

See Also:

`BiFunction`, `UnaryOperator`

Method Summary

Static Methods

Modifier and Type	Method	Description
`static <T> BinaryOperator<T>`	`maxBy(Comparator<? super T> comparator)`	Returns a `BinaryOperator` which returns the greater of two elements according to the specified `Comparator`.
`static <T> BinaryOperator<T>`	`minBy(Comparator<? super T> comparator)`	Returns a `BinaryOperator` which returns the lesser of two elements according to the specified `Comparator`.

Methods declared in interface java.util.function.BiFunction

`andThen`, `apply`

Method Details

minBy

```
static <T> BinaryOperator<T> minBy(Comparator<? super T> comparator)
```

Returns a `BinaryOperator` which returns the lesser of two elements according to the specified `Comparator`.

Type Parameters:

T - the type of the input arguments of the comparator

Parameters:

`comparator` - a `Comparator` for comparing the two values

Returns:

a `BinaryOperator` which returns the lesser of its operands, according to the supplied `Comparator`

Throws:

`NullPointerException` - if the argument is null

maxBy

```
static <T> BinaryOperator<T> maxBy(Comparator<? super T> comparator)
```

Returns a `BinaryOperator` which returns the greater of two elements according to the specified `Comparator`.

Type Parameters:

T - the type of the input arguments of the comparator

Parameters:

`comparator` - a `Comparator` for comparing the two values

Returns:

a `BinaryOperator` which returns the greater of its operands, according to the supplied `Comparator`

Throws:

`NullPointerException` - if the argument is null
