

## HIPOTECA FLEXIBLE

DECIDES EL DÍA DE PAGO Y MODIFICAS  
CÓMO Y CUÁNTO PAGAS



Noticias Base de conocimientos Ofertas Acerca de

g+ f t in Sea



**Java Code Geeks**  
JAVA 2 JAVA DEVELOPERS RESOURCE CENTER

## HIPOTECA FLEXIBLE

DECIDES EL DÍA DE PAGO Y MODIFICAS  
CÓMO Y CUÁNTO PAGAS



ANDROIDE JAVA CENTRAL JAVA DE ESCRITORIO JAVA EMPRESARIAL FUNDAMENTOS DE JAVA IDIOMAS JVM DESARROLLO DE SOFTWARE

Inicio » Core Java » Java 8 Expresiones Lambda Ejemplo de introducción

### SOBRE YATIN



El autor se graduó en Electrónica y Telecomunicaciones. Durante sus estudios, ha estado involucrado en un número significativo de proyectos que van desde programación e ingeniería de software hasta análisis de telecomunicaciones. Trabaja como líder técnico en el sector de la tecnología de la información, donde se dedica principalmente a proyectos basados en la plataforma de tecnologías Java / J2EE y nuevas tecnologías de IU.



## Ejemplo de introducción de expresiones Lambda de Java 8

Publicado por: Yatin en Core Java 9 de julio de 2019 0 963 reproducciones

Hola lectores En este tutorial, presentamos un artículo completo sobre Java 8 Expresiones Lambda.

### 1. Introducción

Para lograr los beneficios de la programación funcional en Java, los desarrolladores de JDK presentaron Lambda Expressions en la programación de Java 8.

- Una expresión lambda es una función sin nombre que no tiene el nombre, el tipo de retorno y los modificadores de acceso.
- Se puede asignar una instancia de expresión lambda a cualquier interfaz que tenga un solo método abstracto (también conocido como Interfaces funcionales)
- Una expresión lambda también se conoce como funciones anónimas o cierres

- Una expresión lambda en la memoria permanente de JVM (es decir, área de método)
- Sintaxis:

```
1 | (Parameter-list) -> { Body }
```



### ¿Quieres ser un Java 8 Ninja?

Suscríbete a nuestro boletín y descarga la Guía definitiva de características de Java 8 [ahora mismo](#).

Con el fin de ponerlo al día con la versión principal de Java 8, hemos compilado una guía increíble con todas las nuevas características y beneficios. ¡Además de estudiarlos en línea, puede descargar el libro electrónico en formato PDF!

[¡Descargar ahora!](#)

TAN A GUSTITO,  
SIN PERMANEN



### HOJA INFORMATIVA

**162,963** personas con **información** privilegiada de actualizaciones semanales **blancos** gratuitos! **Únase a ellos ahora** **acceso exclusivo** a las en el mundo de Java, así como sobre Android, Scala, Groovy tecnologías relacionadas.

**Dirección de correo electrónico:**

Your email address

☒ Reciba alertas de trabajo desarrollador en su área

[Regístrate](#)

[ÚNETE A NOSOTROS](#)



Parameters list

Lambda Expression body

## 1.1 Comprensión de la expresión de Lambda

- Una expresión lambda nunca puede ser instanciada y extender clases abstractas o concretas
- Una expresión lambda nunca puede tener variables de instancia. Cualquier variable declarada dentro de una expresión lambda es una variable local

this

La palabra clave dentro de una expresión lambda representa la referencia de objeto de clase externa (es decir, la clase en la que se declara la expresión lambda)

- Se requieren *interfaces funcionales* para llamar a una expresión lambda
- Una expresión lambda puede tener cero o más parámetros.
- Los programadores pueden o no especificar el tipo de datos de parámetro (s) en una expresión lambda. Si el compilador detecta el tipo de parámetro (s) según el contexto, los programadores pueden eliminar el tipo de datos. Por ejemplo,

```
1 // Lambda expression.
2 (int a, int b) -> System.out.println("a= " + a + ", b= " + b);
3
4 // Re-written lambda expression.
5 (a, b) -> System.out.println("a= " + a + ", b= " + b);
```

- Múltiples parámetros en una expresión lambda deben separarse con una coma
- Si el número de parámetros en una expresión lambda es cero, entonces los programadores deben especificar parámetros vacíos como

()

. Por ejemplo,

```
1 // Lambda expression.
2 () -> System.out.println("Hello World!");
```

- Si un único parámetro está disponible en una expresión lambda y el compilador puede detectar el tipo de datos, los programadores pueden eliminar el tipo y el paréntesis. Por ejemplo,

```
1 // Lambda expression.
2 (int x) -> System.out.println(x);
3
4 // Re-written lambda expression.
5 x -> System.out.println(x);
```

- Similar al cuerpo del método, el cuerpo de la expresión lambda puede tener múltiples declaraciones. Si hay más de una declaración presente, entonces las llaves son obligatorias, de lo contrario, son opcionales

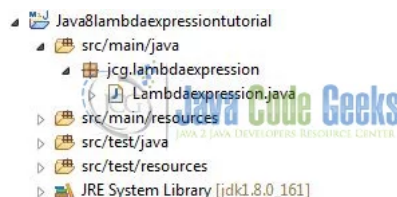
## 1.2 Ventajas de la expresión Lambda

- Reduce la longitud del código para aumentar la legibilidad del código
- Resuelve las complejidades de las clases internas anónimas.
- Se puede pasar como argumentos de método

Para comenzar con este tutorial, esperamos que los usuarios tengan actualmente su IDE y JDK 1.8 preferidos instalados en sus máquinas. Para un uso fácil, estoy usando Eclipse IDE.

## 2. Ejemplo de Introducción a las Expresiones Lambda de Java 8

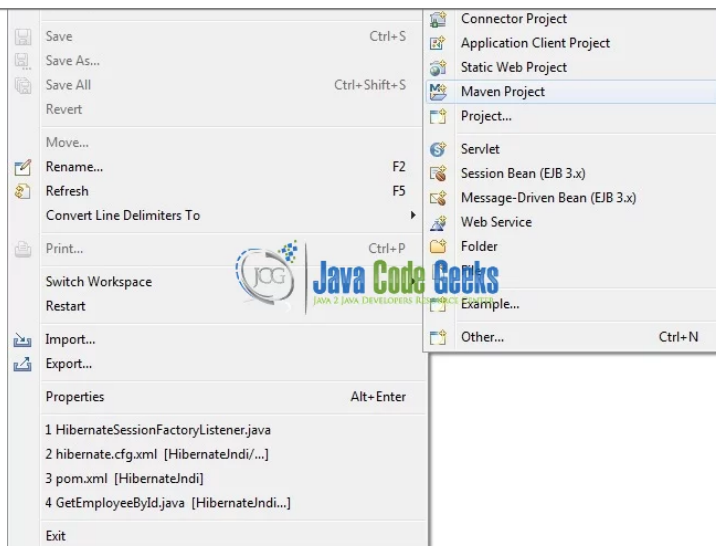
En primer lugar, revisemos la estructura del proyecto que tiene una sola clase para demostrar una expresión lambda en Java 8.



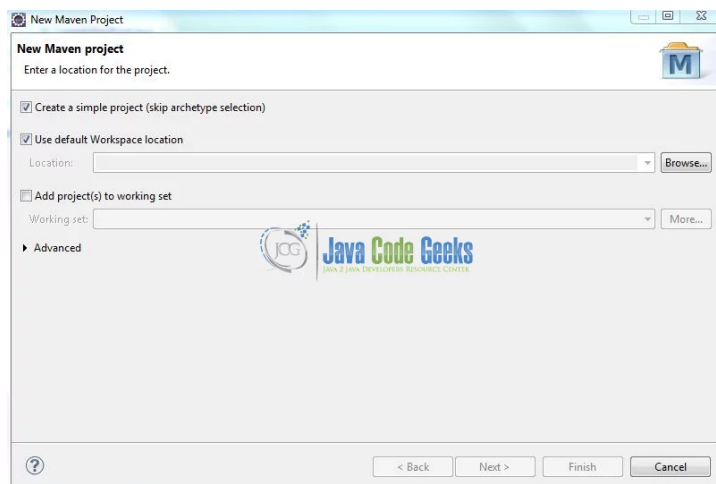
### 2.1 Creación de proyectos

Esta sección mostrará cómo crear un proyecto Maven basado en Java con Eclipse. En Eclipse IDE, vaya a





En la ventana Nuevo proyecto Maven, le pedirá que seleccione una ubicación de proyecto. De forma predeterminada, se seleccionará 'Usar ubicación de espacio de trabajo predeterminada'. Seleccione la casilla de verificación 'Crear un proyecto simple (omitir la selección de arquetipos)' y simplemente haga clic en Siguiente para continuar.



Le pedirá que 'Ingrese el grupo y la identificación del artefacto para el proyecto'. Ingresaremos los detalles como se muestra en la imagen a continuación. El número de versión será por defecto:

0.0.1-SNAPSHOT

Haga clic en Finalizar y se completará la creación de un proyecto maven. Si observa, ha descargado las dependencias de Maven y se

pom.xml

creará un archivo.

### 3. Construcción de aplicaciones

Para crear una nueva clase, haga clic derecho en la

src/main/java

carpeta,

New -> Class

. Complete los detalles sobre el paquete y el nombre de la clase como se muestra en la Fig. 6 y haga clic en finalizar.

Lambdaexpression.java

```

01 package jcg.lambdaexpression;
02
03 @FunctionalInterface
04 interface Interface1 {
05     public void greetings();
06 }
07
08 @FunctionalInterface
09 interface Interface2 {
10     public void name(String name);
11 }
12
13 @FunctionalInterface
14 interface Interface3 {
15     public void add(int a, int b);
16 }
17
18 años @FunctionalInterface
19 interface Interface4 {
20     public int multiple(int x);
21 }
22
23 public class Lambdaexpression {
24
25     public static void main(String[] args) {
26
27         // Lambda expression with no parameter.
28         Interface1 interf1 = () -> {
29             System.out.println("Hello World!");
30         };
31         interf1.greetings();
32
33         // Lambda expression with single parameter.
34         // Here java compiler can detect the parameter type based on the context (i.e. Type coherence
35         // Thus the lambda expression can be rewritten as :: Interface2 interf2 = (name) -> { Syst
36         Interface2 interf2 = (String name) -> {
37             System.out.println("My name is= " + name);
38         };
39         interf2.name("Java");
40
41         // Lambda expression with multiple parameters.
42         Interface3 interf3 = (int a, int b) -> {
43             System.out.println("Total sum is= " + (a+b));
44         };
45         interf3.add(5, 5);
46
47         // Lambda expression with return keyword.
48         // Here as curly parentheses consists of a single statement, we can omit them and the return
49         // Thus the lambda expression can be rewritten as :: Interface4 interf4 = (int x) -> x*x;
50         Interface4 interf4 = (int x) -> {
51             return x*x;
52         };
53         System.out.println("Total result is= " + interf4.multiple(10));
54     }
55 }

```

## 4. Ejecute la aplicación

Haga clic derecho en la

Lambdaexpression.java

clase,

Run As -> Java Application

. La clase se ejecutará y la salida se imprimirá en la consola ide.

Salida

```

1 | Hello World!
2 | My name is= Java
3 | Total sum is= 10
4 | Total result is= 100

```

Eso es todo para este tutorial y espero que el artículo le haya servido lo que estaba buscando. ¡Feliz aprendizaje y no te olvides de compartir!

## 5. Conclusión

En este tutorial, analizamos en profundidad las expresiones Lambda en Java 8. Los desarrolladores pueden descargar la aplicación de muestra como un proyecto Eclipse en la sección Descargas .

## 6. Descargue el proyecto Eclipse

## ¿Quieres saber cómo desarrollar tus habilidades para convertirte en un Rockstar de Java?

Suscríbete a nuestro boletín para comenzar a rockear [ahora mismo](#).

Para comenzar, ile ofrecemos nuestros eBooks más vendidos **GRATIS!**

1. JPA Mini libro
2. Guía de resolución de problemas de JVM
3. Tutorial JUnit para pruebas unitarias
4. Tutorial de anotaciones de Java
5. Preguntas de la entrevista Java
6. Preguntas de la entrevista de primavera
7. Diseño de la interfaz de usuario de Android

y muchos más ....

### Dirección de correo electrónico:

☒ Reciba alertas de trabajo de Java y desarrollador en su área

[Regístrate](#)



¿TE GUSTA ESTE ARTÍCULO? LEER MÁS DE JAVA CODE GEEKS



Fibra de Vodafone

Java Stack Example

Create Automated Test Suites

What are Microservices?

Ad Vodafone

javacodegeeks.com

Ad Postman

javacodegeeks.com

Simula la 2ª Guerra Mundial

Spring Boot Microservices , Docker and Kubernetes workshop – part2

Docker Command Line Example

Consuming J! based REST V Service in AD

Ad Call of War 1942

javacodegeeks.com

javacodegeeks.com

javacodegeeks.com

Deja una respuesta

✉ Suscribir ▼

## KNOWLEDGE BASE

Courses

Minibooks

News

Resources

Tutorials

## THE CODE GEEKS NETWORK

.NET Code Geeks

Java Code Geeks

System Code Geeks

Web Code Geeks

## HALL OF FAME

Android Alert Dialog Example

Android OnClickListener Example

How to convert Character to String and a String to Character Array in Java

Java Inheritance example

Java write to File Example

java.io.FileNotFoundException – How to solve File Not Found Exception

java.lang.arrayindexoutofboundsexception – How to handle Array Index Out Of Bounds Exception

java.lang.NoClassDefFoundError – How to solve No Class Def Found Error

JSON Example With Jersey + Jackson

Spring JdbcTemplate Example

## ABOUT JAVA CODE GEEKS

JCGs (Java Code Geeks) is an independent online community focused on the ultimate Java to Java developers resource center; targeted at the technical team lead (senior developer), project manager and junior dev. JCGs serve the Java, SOA, Agile and Telecom communities with daily news, domain experts, articles, tutorials, reviews, announcements, code snippets and source projects.

## DISCLAIMER

Todas las marcas comerciales y marcas comerciales registradas que aparecen en este sitio web son propiedad de sus respectivos dueños. Java es una marca comercial registrada de Oracle Corporation en los Estados Unidos y en otros países. Ejemplos Java Code Geeks no está conectado a Oracle Corporation patrocinado por Oracle Corporation.