



Lambda y Stream

Que es?

El cálculo lambda es el lenguaje de programación más pequeño conocido en transformación (operadores variables) y en definir funciones. El cálculo lambda se puede decir que es equivalente a las máquinas Turing porque es capaz de evaluar y expresar cualquier función computable.

Functional Interfaces

Que es?

Una interfaz funcional es una interfaz que contiene solo un método abstracto. Solo pueden tener una funcionalidad para exhibir. Desde Java 8 en adelante, las expresiones lambda se pueden usar para representar la instancia de una interfaz funcional. Una interfaz funcional puede tener cualquier cantidad de métodos predefinidos. Runnable, ActionListener, Comparable son algunos de los ejemplos de interfaces funcionales.

Cuales son?

- consumers
- functions
- suppliers
- predicates
- operators

Que son?

Son simples, sin ninguna declaración, es decir, sin modificadores de acceso, la declaración de valor de retorno y nombre. Están en el lado derecho del operador \rightarrow . Una expresión lambda devuelve el resultado de la evaluación de la expresión.

Expresiones lambda

Que son?

1. El operador lambda (\rightarrow) separa la declaración de parámetros de la declaración del cuerpo de la función.
2. Parámetros
3. Cuerpo de la función
Cuando no se tienen parámetros, o cuando se tienen dos o más, es necesario utilizar paréntesis.
Cuando el cuerpo de la expresión lambda tiene una única línea no es necesario utilizar las llaves y no necesitan especificar la cláusula return en el caso de que deban devolver valores.
Cuando el cuerpo de la expresión lambda tiene más de una línea se hace necesario utilizar las llaves y es necesario incluir la cláusula return en el caso de que la función deba devolver un valor.

Sintaxis

(parámetros) -> (cuerpo lambda)

Stream

Que es?

Un Stream es una abstracción que se enfoca en conjuntos y no en piezas individuales de código permitiendo realizar programación funcional en lugar de hacer programación imperativa.

Referencias

- cecilia urbina. (2010). Cálculo lambda. 17/marzo/2020, de blogspot Sitio web: <http://ceciliaurbina.blogspot.com/2010/11/calculo-lambda.html>
- Mariana. (2017). Cálculo Lambda (No Tipado). 17/marzo/2020, de github Sitio web: [https://github.com/iguar-tallos-de-linguajes-taller-2017/wiki/C%3A%3ALambda-\(No-Tipado\)](https://github.com/iguar-tallos-de-linguajes-taller-2017/wiki/C%3A%3ALambda-(No-Tipado))
- tutorialspoint. (2019). Java 8 - Functional Interfaces. 17/marzo/2020, de tutorialspoint Sitio web: https://www.tutorialspoint.com/java/java8_functional_interfaces.htm
- mosheret. (2017). Functional Interfaces In Java. 17/marzo/2020, de geeksforgeeks Sitio web: <https://www.geeksforgeeks.org/functional-interfaces-java/>
- Juan Mellado. (2018). Interfaces Funcionales en Java. 17/marzo/2020, de inventa Sitio web: <https://inventas.com/blog/2018/05/09/interfaces-funcionales-en-java/>
- Cecilio Alvarez. (2018). Java 8 Functional Interfaces y sus tipos. 17/marzo/2020, de arquitecturajava Sitio web: <https://www.arquitecturajava.com/java-8-functional-interfaces-y-sus-tipos/>
- Jose Luis Rodriguez Villapacellin. (2015). Expresiones Lambda con Java 8. 17/marzo/2020, de adictosaltrabajo Sitio web: <https://www.adictosaltrabajo.com/2015/12/04/expresiones-lambda-con-java-8/>
- Elkar Larga. (2016). Tutorial de expresiones Lambda en Java 8 paso a paso. 17/marzo/2020, de ecodeup Sitio web: <https://www.ecodeup.com/entendiendo-paso-a-paso-las-expresiones-lambda-en-java/>
- Alfonso Garcia. (2016). ¿Que es y para que sirve un stream? 17/marzo/2020, de quora Sitio web: <https://es.quora.com/Qu%C3%A9-es-y-para-qu%C3%A9-sirve-un-stream>

1.- El cálculo lambda es el lenguaje de programación más pequeño conocido en transformación (operadores variables) y en definir funciones. El cálculo lambda se puede decir que es equivalente a las máquinas Turing porque es capaz de evaluar y expresar cualquier función computable. Church había querido hacer un sistema formal completo para modelar la matemática pero después se separó el cálculo lambda y lo dejó para que estudiara la computabilidad.

El Cálculo Lambda (o, mejor dicho, LOS Cálculos Lambda) son una herramienta interesante porque modelan una de las piezas centrales (y más flexibles) de la computación. La Abstracción. Variante de estos cálculos se utilizan muchísimo para probar conceptos y analizar propiedades y son una de las construcciones más estudiadas en la bibliografía.

2.- A functional interface is an interface that contains only one abstract method. They can have only one functionality to exhibit. From Java 8 onwards, lambda expressions can be used to represent the instance of a functional interface. A functional interface can have any number of default methods. Runnable, ActionListener, Comparable are some of the examples of functional interfaces.

Functional interfaces have a single functionality to exhibit. For example, a Comparable interface with a single method compareTo() is used for comparison purpose. Java 8 has defined a lot of functional interfaces to be used extensively in lambda expressions.

- consumers
- functions
- suppliers
- predicates
- operators

4.- Son funciones anónimas que se usan frecuentemente para crear delegados en LINQ. Son simples, sin ninguna declaración, es decir, sin modificadores de acceso, la declaración de valor de retorno y nombre. Están en el lado derecho del operador \rightarrow . Una expresión lambda devuelve el resultado de la evaluación de la expresión. Las expresiones lambda son funciones anónimas, es decir, funciones que no necesitan una clase.

5.- su sintaxis básica se detalla a continuación:

1. El operador lambda (\rightarrow) separa la declaración de parámetros de la declaración del cuerpo de la función.
2. Parámetros
3. Cuerpo de la función
Cuando no se tienen parámetros, o cuando se tienen dos o más, es necesario utilizar paréntesis.
Cuando el cuerpo de la expresión lambda tiene una única línea no es necesario utilizar las llaves y no necesitan especificar la cláusula return en el caso de que deban devolver valores.
Cuando el cuerpo de la expresión lambda tiene más de una línea se hace necesario utilizar las llaves y es necesario incluir la cláusula return en el caso de que la función deba devolver un valor.

6.- En general se utiliza el término "Stream" para describir los flujos de datos donde una vez que se envía un dato este no se va a volver a enviar nunca.

El caso más común de "streams" son las señales de video o audio en directo. Los micrófonos o sensores de luz no pueden volver a enviar los valores que están recogiendo y enviando a otros dispositivos ya que no disponen de un sitio para guardarlos. Cuando se habla de "Streams" en protocolos multimedia el significado es ligeramente distinto. En ese caso los datos que se envían pueden que se puedan volver a enviar varias veces, pero el protocolo que los envía solo sabe enviar tramos delimitados por marcas de tiempo. Esta marca tienen una resolución arbitraria, que puede llegar a ser tan poco fina como solo poder enviar todos los datos desde el principio.

The streams se usan en situaciones donde es más importante que haya un flujo continuo de datos que el que los datos estén completos el 100 por 100, bien porque son tolerantes las inconexiones, bien porque su volumen es tan grande que solo nos cabe una cantidad limitada de ellos en el dispositivo y vamos a descartar enseguida los más antiguos, o bien porque simplemente no nos interesa más que el valor que tiene ahora mismo. El caso más clásico de nuevo es la reproducción de video.

Un Stream es una abstracción que se enfoca en conjuntos y no en piezas individuales de código permitiendo realizar programación funcional en lugar de hacer programación imperativa.