INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

Ingeniería en Sistemas Computacionales

7mo Semestre

Alumno:

Daniel Alejandro de la Rosa Castañeda

NC:16070126

Materia:

Programación lógica y funcional

Nombre del trabajo:

Mapa Conceptual

Docente:

ISC Salvador Acevedo

Jerez de García Salinas a 20 de marzo del 2020

1. Matemáticamente, que es el cálculo Lambda

Originalmente, Church había tratado de construir un sistema formal completo para modelizar la Matemática; pero cuando éste se volvió susceptible a la paradoja de Russell, separó del sistema al cálculo lambda y lo usó para estudiar la computabilidad, culminando en la respuesta negativa al problema de la parada.

El cálculo lambda puede considerarse como la base teórica de la programación funcional. Es un lenguaje completo de Turing; es decir, cualquier máquina que pueda calcular el cálculo lambda puede calcular todo lo que una máquina Turing puede (y viceversa).

2. Que son las “functional interfaces” en Java

Una interfaz funcional es una interfaz que contiene solo un método abstracto. Solo pueden tener una funcionalidad para exhibir. A partir de Java 8, las expresiones lambda se pueden utilizar para representar la instancia de una interfaz funcional. Una interfaz funcional puede tener cualquier número de métodos predeterminados. Runnable, ActionListener, Comparable son algunos de los ejemplos de interfaces funcionales.

3. ¿Cuáles son las 6 interfaces funcionales del paquete java.util.function?

- Function: Representa una función que acepta un argumento y genera un resultado.
- LongPredicate: Representa un predicado (función con valores booleanos) de un argumento con valores.long
- Predicate: Representa un predicado (función con valores booleanos) de un argumento.
- Supplier: Representa a un proveedor de resultados.
- IntConsumer: Representa una operación que acepta un argumento de un solo valor y no devuelve ningún resultado.int
- LongSupplier: Representa un proveedor de resultados valiosos.long

4. ¿Qué son las expresiones lambda?

Son funciones anónimas que se usan frecuentemente para crear delegates en LINQ. Son simples, sin ninguna declaracion, es decir, sin modificadores de acceso, la declaración de valor de retorno y nombre.

5. Sintaxis de las expresiones lambda

Están en el lado derecho del operador =>. Una expresión lambda devuelve el resultado de la evaluación de la condición.

(input-parameters) => expression

6. Que son los streams y para qué sirven

A manera de definición, un stream es una especie de canal a través del cual fluyen los datos. Técnicamente, un stream es el enlace lógico utilizado por el programador

Los streams más comunes son los archivos. También pueden ser caracteres de un string o bloque de memoria o bytes de un “socket”. La idea es que el programador los trate de la misma forma que archivos

7. Funciones más relevantes de la clase stream

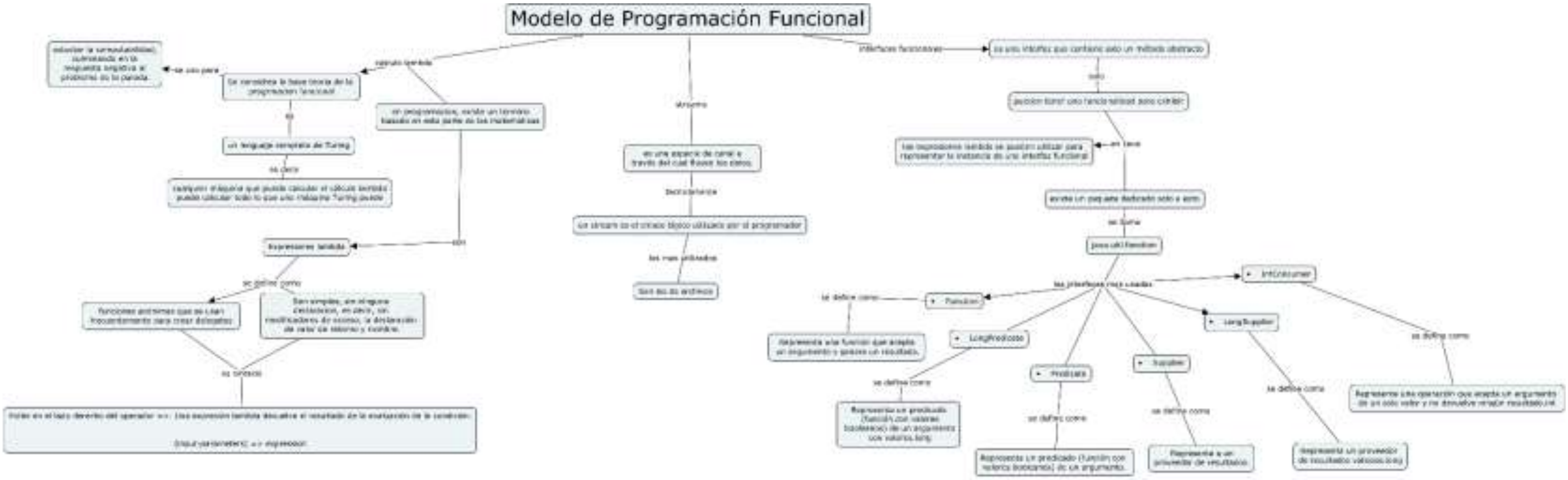
El siguiente ejemplo nos permite visualizar las tres (3) partes que componen un Stream:

Fuente de información → La lista de transacciones

Cero o más operaciones intermedias → Se puede apreciar la operación filter y la operación mapToInt, operaciones que serán explicadas más adelante en este mismo artículo.

Operación terminal que produce un resultado o un efecto sobre los datos → Se puede apreciar la operación sum, la cual se explicará más adelante en este mismo artículo.

```
List transacciones = ...  int  sum = transacciones.stream().
    filter(t ->  t.getProveedor().getCiudad().equals(“Cali”)).
    mapToInt(Transaccion::getPrecio).
    sum();
```



Modelo de Programación Funcional

cálculo lambda

Se considera la base teoría de la programación funcional

se uso para

resolver la computabilidad, contestando en la respuesta negativa al problema de la parada.

es

un lenguaje completo de Turing

es decir

cualquier máquina que pueda calcular el cálculo lambda puede calcular todo lo que una máquina Turing puede

en programación, existe un término basado en esta parte de las matemáticas

son

Expresiones lambda

se define como

Funciones anónimas que se usan frecuentemente para crear delegados

Son simples, sin ninguna declaración, es decir, sin modificadores de acceso, la declaración de valor de retorno y nombre.

su sintaxis

Está en el lado derecho del operador =>. Una expresión lambda devuelve el resultado de la evaluación de la condición.

(input-parameters) => expression

streams

es una especie de canal a través del cual fluyen los datos.

técnicamente

un stream es el único bloque utilizado por el programador

los mas utilizados

son los de archivos

interfaces funcionales

es una interfaz que contiene solo un método abstracto

solo

[illegible]

en Java

las expresiones lambda se pueden utilizar para representar la instancia de una interfaz funcional

existe un paquete dedicado solo a esto

se llama

java.util.function

las interfaces mas usadas

- Function

se define como

Representa una función que acepta un argumento y genera un resultado.

- LongPredicate

se define como

Representa un predicado (función con valores booleanos) de un argumento con valores long

- Predicate

se define como

Representa un predicado (función con valores booleanos) de un argumento.

- Supplier

se define como

Representa a un proveedor de resultados

- LongSupplier

se define como

Representa un proveedor de resultados valorados long

- IntConsumer

se define como

Representa una operación que acepta un argumento de un solo valor y no devuelve ningún resultado.int

Referencias:
--> https://www.geeksforgeeks.org/functional-interfaces-java/
-->https://sites.google.com/site/[illegible]/calculo-lambda
-->https://www.ecured.cu/C%C3%A1lculo_lambda
-->https://brilliant.org/wiki/lambda-calculus/
-->https://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html
-->https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/statements-expressions-operators/lambda-expressions
-->https://geekytheory.com/entendiendo-las-expresiones-lambda
-->https://www.[illegible].com/2015/09/23/uso-basico-de-java-8-stream-y-lambdas/
-->https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C%2B%2B/Streams
-->https://www.oracle.com/technetwork/es/articles/java/expresiones-lambda-api-stream-java-2633852-esa.html
-->https://www.oracle.com/technetwork/es/articles/java/expresiones-lambda-api-stream-java-2737544-esa.html