INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

Ingeniería en Sistemas Computacionales

8° Semestre

Alumno: José Félix Alvarado Barrios

NC:16070124

Materia: Prog. Lógica y Funcional

Nombre del trabajo: Mapa conceptual

Docente: ISC Salvador Acevedo Sandoval

Jerez de García Salinas a 20 de Marzo 2020

1. ¿Matemáticamente que es el cálculo lambda?

El lambda calculo (λC) es un lenguaje simple que permite la descripción de las funciones matemáticas y de sus propiedades; fue introducido por Church en los años 30 como fundamento de la matemática (funciones y lógica) y constituye un modelo formal; muchos lenguajes funcionales (como Haskell) son a menudo descritos como un super λC o λC extendido.

2. ¿Que son las interfaces funcionales en java?

Se le conoce como interface funcional a toda aquella interface que tenga solamente un método abstracto, es decir puede implementar uno o más métodos default, pero deberá tener forzosamente un único método abstracto.

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

4. Que son las expresiones lambda

En el ámbito de la programación, una expresión lambda, también denominada función lambda, función literal o función anónima; es una subrutina definida que no está enlazada a un identificador.

5. Sintaxis de las expresiones lambda

(Parametros)->{Cuerpo- lambda}

El operador lambda (->) separa la declaración de parámetros de la declaración del cuerpo de la función.

Parámetros:

Cuando se tiene un solo parámetro no es necesario utilizar los paréntesis.

Cuando no se tienen parámetros, o cuando se tienen dos o más, es necesario utilizar paréntesis.

Cuerpo de lambda:

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.

Ejemplo (int longitud, int altura) -> { return altura * longitud; }

6. Que son los streams y para qué sirven

Stream se define como una secuencia de elementos que provienen de una fuente que soporta operaciones para el procesamiento de sus datos:

- De forma declarativa usando expresiones lambda.
- Permitiendo el posible encadenamiento de varias operaciones, con lo que se logra tener un código fácil de leer y con un objetivo claro.
- De forma secuencial o paralela (Fork/Join).

Stream nos permite realizar operaciones de tipo filtro/mapeo/reducción sobre colecciones de datos de forma secuencial o paralela.

7. Funciones mas relevantes de la clase Stream

Stream.of(T...): Stream<T>

Retorna un Stream ordenado y secuencial de los elementos pasados por parámetro.

Stream.empty():Stream

Retorna un Stream secuencial y vacío.

Arrays.stream(T[]):Stream<T>

Retorna un Stream secuencial del arreglo pasado por parámetro. Si T[] es un arreglo de datos “primitivos” entonces retorna: DoubleStream, IntStream o LongStream según el caso.

Collection<E>.stream():Stream<E>

Retorna un Stream secuencial de los elementos de la colección, para obtener una versión en paralelo basta con usar: Collection<E>.parallelStream():Stream<E>

Stream.iterate(T, UnaryOperator<T>):Stream<T>

Retorna un Stream infinito, ordenado y secuencial a partir del valor inicial T y de aplicar la función pasada por parámetro UnaryOperator al valor inicial para obtener los demás elementos. Para limitar su tamaño, se puede usar el método +limit(long):Stream

Stream.generate(Supplier<T>):Stream<T>

Retorna un Stream infinito, secuencial pero no ordenado, a partir de una función de tipo Supplier que provee los elementos.

+filter(Predicate<T>):Stream<T>

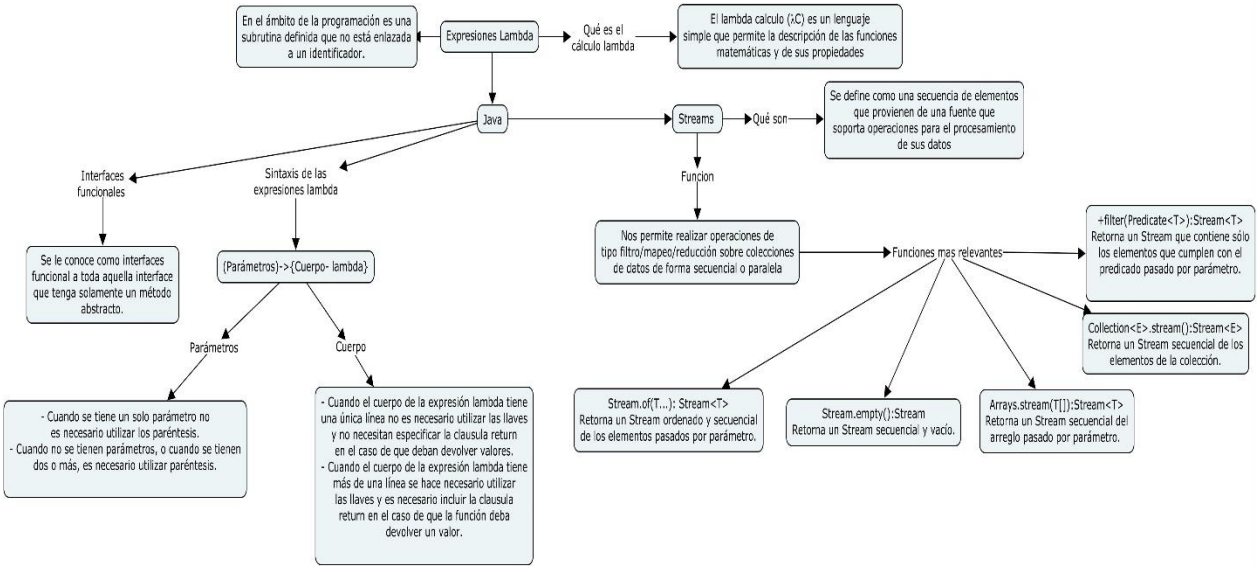
Retorna un Stream que contiene sólo los elementos que cumplen con el predicado pasado por parámetro.

+distinct():Stream<T>

Retorna un Stream sin elementos duplicados. Depende de la implementación de +equals(Object):boolean.

+limit(long):Stream<T>

Retorna un Stream cuyo tamaño no es mayor al número pasado por parámetro. Los elementos son cortados hasta ese tamaño.



Referencias

Docs Oracle. (s.f.). Obtenido de Oracle: https://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html#package.description

Java 8 – Interfaces funcionales. (8 de Diciembre de 2016). Obtenido de Java 8 – Interfaces funcionales: https://www.oscarblancarteblog.com/2016/12/08/java-8-interfaces-funcionales/

Jimíenez, B. C., & García, P. G. (6 de Marzo de 2002). El λ–cálculo (sin tipos y con tipos). Malaga, Malaga, España.

Lopez, A. (Octubre de 2015). *Introducción a Expresiones Lambda y API Stream en Java SE 8*. Obtenido de Oracle: https://www.oracle.com/technetwork/es/articles/java/expresiones-lambda-api-stream-java-2737544-esa.html