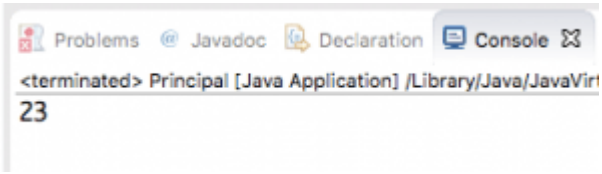
Java Lambda reduce es una de las operaciones más utilizadas cuanto trabajamos con colecciones de objetos y expresiones lambda. Reduce sirve para convertir un Array de elementos en un único elemento y se usa por ejemplo para calcular la suma de n términos. Veamos un ejemplo sin expresiones lambda:

```
package com.arquitecturajava;

import java.util.ArrayList;
import java.util.List;

public class Principal {

public static void main(String[] args) {
List<Integer> lista= new ArrayList<Integer>();
lista.add(5);
lista.add(2);
lista.add(7);
lista.add(9);
int total=0;
for (int numero:lista) {

total+=numero;
}

System.out.println(total);


}
```

```
}
```

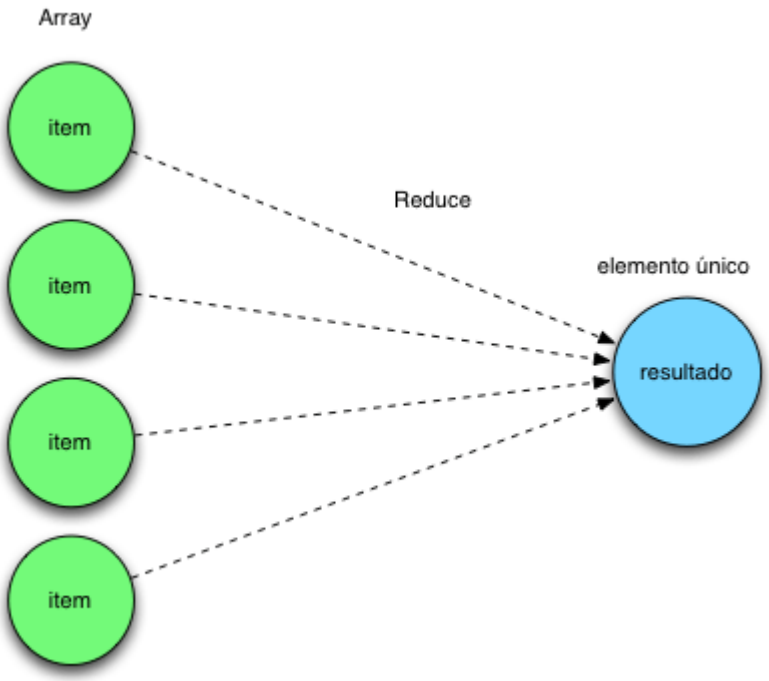
```
 
```

El resultado por pantalla será :



Java Lambda reduce

Es fácil convertir esto al mundo de los Streams y las expresiones lambda utilizando Java Lambda reduce para que sea esa función la que sume.



```
package com.arquitecturajava;

import java.util.stream.IntStream;

public class Principal2 {

public static void main (String args[]) {


int resultado=IntStream.of(5,2,7,9).reduce(0,(x,y)->x+y);
System.out.println(resultado);
}
}
```

El código se reduce de forma clara , la realidad es que este tipo de operaciones las tenemos que hacer continuamente. La nueva forma de operar con Streams y Lambda lo simplifica mucho .¿Podemos mejorar un poco el código?. Si en vez de crear nosotros nuestra propia expresión lambda podemos usar un reference method y apuntar a la clase wrapper(Integer) que será la encargada de realizar la suma apoyándonos en el método sum.

```
package com.arquitecturajava;

import java.util.stream.IntStream;

public class Principal3 {

public static void main (String args[]) {
```

```
int resultado=IntStream.of(5,2,7,9).reduce(0,Integer::sum);
System.out.println(resultado);
}
}
```

De esta forma evitaremos errores. La clase Integer acaba de añadir nuevos métodos en Java 8 para que puedan ser utilizados por las expresiones lambda , sum es uno de ellos.

otros artículos relacionados: Expresiones Lambda , JavaScript Reduce , Java Lambda Workflow