Instituto Politécnico Nacional

Escuela Superior de Cómputo

Departamento de Ciencias e Ingenieria de la Computación

Paradigmas de Programación

Profesor: Miguel Ángel Rodriguez Castillo

Grupo: 3CV2

Práctica : 2 Técnicas de Programación Funcional

Equipo:

- Ramírez Jiménez Itzel Guadalupe
- Colín Ramiro Joel

Fecha de entrega: 29 / 03 / 2021

Planteamiento del problema

Sin duda, la programación funcional es una técnica bastante implementada hoy en día no solo por su efectividad si no por su fama de ser más sencilla que otros tipos de programación. En esta práctica se realizarán dos ejercicios, en el primero se analizarán las técnicas eager y lazy sobre una expresión dada la cual tiene un parámetro el cual es $\frac{1}{0}$, la cual se indetermina, debido a que no existe ningún número divido en 0. En matemáticas, muchas veces en la parte de límites se dice que cuando se tiene una expresión similar, es decir cuando se tiene que un número se divida en 0, normalmente se tiene que esa expresión se va a infinito. En el segundo ejercicio se definirá una lista de 100 elementos y respecto al elemento que proporcione el usuario, se buscará y se indicará la posición en donde se encuentre dicho elemento.

Implementación de la solución

Ejercicio 1

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package eagerylazy;

/**
 *
 * @author itzel
 */
public class Eagerylazy {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
      //eager
      System.out.println("eager");
      int x=3*3;
      /* y=1/0;
       prim(x,y) = x;
      prim(cuad 3, 1/0)
      prim(3*3,indeterminado)
      prim(9,indeterminado)
      9*/
      System.out.println("" + x);
```

```
        System.out.println("Lazy");
        /*prim(cuad 3,1/0)
        (cuad 3)
        3*3
        9
        */
        System.out.println("" + x);

    }
```

prim (cuad 3, 1/0)

Eager

=prim (3*3, 1/0)

=prim (3*3, indeterminado)

=prim (9, indeterminado)

=9

Lazy

=cuad 3

=3*3

=9

Ejercicio 2

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package practica2;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Random;

/**
 *
 * @author itzel
 */
public class Practica2 {
```

```
   /**
    * @param args the command line arguments
    */

   public static int existeEnArreglo(int[] arreglo, int busqueda) {
  for (int x = 0; x < arreglo.length; x++) {
   if (arreglo[x] == busqueda) {
    return x;
   }
  }
  return -1;
}

   public static void main(String[] args) {

       int numeros[] = new int[101];

       for (int i=0;i<=100;i++){

       numeros[i] = (int)(Math.random()*100);

       System.out.println("" + i + "" +numeros[i]);

       }

       int numeroBuscado = 1289;
        int posicionDeElementoBuscado = existeEnArreglo(numeros, numeroBuscado);
       if (posicionDeElementoBuscado == -1) {
        System.out.println("El elemento NO existe en el arreglo");
        } else {
        System.out.println("El elemento existe en la posición: " + posicionDeElementoBuscado);
        }

   }

  }
```

Funcionamiento

Ejercicio 1

prim (cuad 3, 1/0)

Eager

=prim (3*3, 1/0)

=prim (3*3, indeterminado)

=prim (9, indeterminado)

=9

Lazy

=cuad 3

=3*3

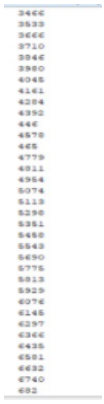
=9

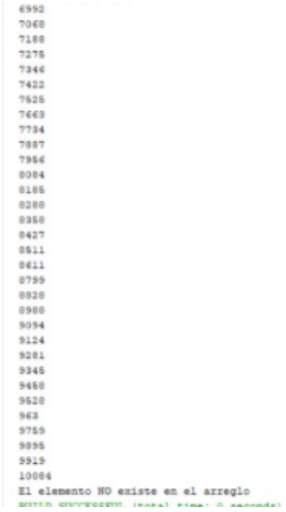
```
run:
eager
9
Lazy
9
BUILD SUCCESSFUL (total time: 0 seconds)
```

Prim(cuad x, 1/0)	cuad x	1/0	Valor final
Prim(cuad 3, 1/0)	3*3	indeterminación	**9**
Prim(cuad 4, 1/0)	4*4	indeterminación	**16**
Prim(cuad 5, 1/0)	5*5	indeterminación	**25**
Prim(cuad 6, 1/0)	6*6	indeterminación	**36**

Ejercicio 2

Numero	Numero dado	Coincide
1	3	**False**
10	4	**False**
20	20	**True**
30	5	**False**
40	10	**False**
50	70	**False**
60	60	**true**





Conclusiones

Estas técnicas implementadas en la práctica consideramos que son sencillas de entender y fáciles de implementar, se nos complicó un poco en el segundo ejercicio, ya que no sabíamos que librerías correspondían, finalmente lo pudimos resolver. No obstante, si analizamos a detalle estas y otras técnicas de la programación funcional para apoyarnos en la realización de esta práctica. Concluimos que tal vez no es la técnica más sencilla ni la más adecuada en muchos casos, pero si una de las más sencillas de entender y con la que alguien que no este familiarizado con la programación pueda iniciarse en este ámbito.

Bibliografía

- IONOS. (2020). Programación funcional. marzo 29 de 2021, de Ionos Sitio web: https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/programacion-funcional/#:~:text=En%20un%20programa%20funcional%2C%20todos,variante%20especial%20de%20un%20subprograma.
- Javier Velez Reyes. (2017). Las 3 Evaluaciones De La Programación Funcional. marzo 29 de 2021, de Javier Velez Reyes Sitio web: https://javiervelezreyes.com/las-3-evaluaciones-de-la-programacion-funcional/
- Francisco J. Gil Gala. (2015). Tres técnicas con las que programar de forma más eficiente. marzo 29 de 2020, de rootear Sitio web: https://rootear.com/desarrollo/tres-tecnicas-funcionales