Clase 4 - “Desarrollador Java inicial”
Programas y Archivos

1. Tomando los Ejercicios de la clase anterior
    a. haga un main, donde por parámetro ponga 3 números y una letra que represente ascendente o descendente y los muestre ordenados por tal criterio
    b. haga lo mismo, pero solicitando los parámetros de a uno por consola
    c. lo mismo, pero usando los parámetros si hay alguno (como en a) y haciendo (b) si no detecta ninguno. Vea si con una función puede evitar repetir código.
    d.
2. Haga una main donde por parámetro envíe la ruta de un archivo. Ese archivo debe contener números. El programa debe escribir por consola la suma de esos números.
    a. Al programa anterior agreguele un parámetro para que la operación pueda ser suma o multiplicación.
3. 3. Tome el ejercicio B de la clase 3 y que por parámetro se pueda elegir si es una codificación o decodificación, el valor del desplazo, y 2 archivos, uno para la entrada y otro para la salida. Que por pantalla solo indique si terminó o no correctamente, los resultados deben estar en el archivo de salida
4. 4. Suba el proyecto / ejercicios a GIT

Ejercicio 1.a)

```
/*
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
*/

package com.mycompany.clase4ejercicios;

import java.util.Arrays;
import javax.swing.JOptionPane;

/**
** @author luciana*/
```

```java
public class Clase4Ejercicios {
   public static void main(String[] args) {
       JOptionPane.showMessageDialog(null, sort(2,4,0, 'A'));
   }

   public static String sort(int num1, int num2, int num3, char sortBy){

       int[] numeros = {num1, num2, num3};

       for (int i = 0; i < numeros.length; i++){

           if(sortBy == 'D'){
               for (int j = 0; j <numeros.length; j++){
                   if (numeros[i] > numeros[j]){
                       int aux = numeros[i];
                       numeros[i] = numeros[j];
                       numeros[j] = aux;
                   }
               }
           } else if(sortBy == 'A'){
               for (int j = 0; j < numeros.length; j++){
                   if (numeros[i] < numeros[j]){
                       int aux = numeros[i];
                       numeros[i] = numeros[j];
                       numeros[j] = aux;
                   }
               }
           }
       }
       return Arrays.toString(numeros);
   }
}
```

Ejercicio 1.b)

```java
import java.util.Arrays;
import javax.swing.JOptionPane;

public class Clase4Ejercicios {

   public static void main(String[] args) {

       String input_num1 = JOptionPane.showInputDialog("Ingrese número 1");
       int num1 = Integer.parseInt(input_num1);

       String input_num2 = JOptionPane.showInputDialog("Ingrese número 2");
       int num2 = Integer.parseInt(input_num2);

       String input_num3 = JOptionPane.showInputDialog("Ingrese número 3");
       int num3 = Integer.parseInt(input_num3);

       String input_sort = JOptionPane.showInputDialog("Ingrese orden; A(ascendente) / D (descendente)");
       char sort = input_sort.charAt(0);

       JOptionPane.showMessageDialog(null, sort(num1,num2,num3, sort));
   }


   public static String sort(int num1, int num2, int num3, char sortBy){

       int[] numeros = {num1, num2, num3};

       for (int i = 0; i < numeros.length; i++){

           if(sortBy == 'D'){
               for (int j = 0; j <numeros.length; j++){
                   if (numeros[i] > numeros[j]){
```

```
                    int aux = numeros[i];
                    numeros[i] = numeros[j];
                    numeros[j] = aux;
                }
            }
        } else if(sortBy == 'A'){
            for (int j = 0; j < numeros.length; j++){
                if (numeros[i] < numeros[j]){
                    int aux = numeros[i];
                    numeros[i] = numeros[j];
                    numeros[j] = aux;
                }
            }
        }
    }

    return Arrays.toString(numeros);
  }
}
```

Ejercicio 1.c)

```
import java.util.Arrays;
import javax.swing.JOptionPane;
/**
*
* @author luciana
*/
public class Clase4Ejercicios {

   public static void main(String[] args) {
       JOptionPane.showMessageDialog(null, sort(5,'D'));
   }
```

```java
public static String sort(int ...optionalParams){

    int num1;
    int num2;
    int num3;
    char sortBy;

    boolean sort = false;

    if(CheckCharParam(optionalParams)){
        sortBy = (char) optionalParams[optionalParams.length - 1];
        sort = true;
    }else{
        sortBy = InsertChar();
    }

    if(sort){
        switch(optionalParams.length){
            case 1:
                num1 = InsertNum();
                num2 = InsertNum();
                num3 = InsertNum();
                break;
            case 2:
                num1 = optionalParams[0];
                num2 = InsertNum();
                num3 = InsertNum();
                break;
            case 3:
                num1 = optionalParams[0];
                num2 = optionalParams[1];
                num3 = InsertNum();
                break;
            case 4:
```

```
            num1 = optionalParams[0];
            num2 = optionalParams[1];
            num3 = optionalParams[2];
            break;

        default:
            return "Ocurrió un error, reintente";
    }
}else{
    int [] completeParams = CompleteParams(optionalParams);
    num1 = completeParams[0];
    num2 = completeParams[1];
    num3 = completeParams[2];
}

int[] numeros = {num1, num2, num3};
for (int i = 0; i < numeros.length; i++){

    if(sortBy == 'D'){
        for (int j = 0; j <numeros.length; j++){
            if (numeros[i] > numeros[j]){
                int aux = numeros[i];
                numeros[i] = numeros[j];
                numeros[j] = aux;
            }
        }
    } else if(sortBy == 'A'){
        for (int j = 0; j < numeros.length; j++){
            if (numeros[i] < numeros[j]){
                int aux = numeros[i];
                numeros[i] = numeros[j];
                numeros[j] = aux;
            }
        }
```

```
        }
    }

    return Arrays.toString(numeros);
}

public static int InsertNum(){
    String input_num = JOptionPane.showInputDialog("Ingrese número");
    return Integer.parseInt(input_num);
}

public static char InsertChar(){
    String input_char = JOptionPane.showInputDialog("Ingrese orden; A(ascendente) / D (descendente)");
    return input_char.charAt(0);
}

public static boolean CheckCharParam(int[] array){
    boolean check = false;
    if (array.length > 0){
        if (array[array.length - 1] == 'A' || array[array.length - 1] == 'D'){
            check = true;
        }
    }
    return check;
}

public static int[] CompleteParams(int[] params){
    int[] newParams = new int[3];

    for (int i = 0; i < params.length; i++){

        if (params[i] != 'A' || params[i] != 'D'){
            newParams[i] = params[i];
        }
```

```
        }

        for (int i = params.length ; i < newParams.length; i++){
            if (newParams[i] == 0){
                newParams[i] = InsertNum();
            }
        }
        return newParams;
    }
};
```

Ejercicio 2.

```
package com.mycompany.clase4ejercicios;

import java.io.FileInputStream;
import java.io.InputStream;
import java.util.Scanner;

public class Ejercicio2 {
 public static void main(String[] args)  throws Exception {
   try {
     System.out.println(sumarNumeros("number.txt"));
   } catch (Exception e) {
     System.out.println("No se pudo leer el archivo");
   }
 }
 public static int sumarNumeros(String ruta ) throws Exception {
   // Creo un objeto de tipo InputStream para leer el archivo y poder lanzar una excepción si no se puede leer
   InputStream file = new FileInputStream(ruta);
   //Scanner permite leer el archivo linea por linea
   try (Scanner obj = new Scanner(file)) {
     int suma = 0;
```

```
    while(obj.hasNextLine()){

      String linea = obj.nextLine();

      if(linea.matches("[0-9]+")){

        suma += Integer.parseInt(linea);
      }
    }
    return  suma;
  }
 }
}
```

Ejercicio 2a)

```
import java.io.FileInputStream;
import java.io.InputStream;
import java.util.Scanner;
import javax.swing.JOptionPane;

/**
*
* @author luciana
*/
public class Ejercicio2 {
 public static void main(String[] args)  throws Exception {
   String ruta = JOptionPane.showInputDialog("Ingrese la ruta del archivo");
   char operador = JOptionPane.showInputDialog("Ingrese el operador").charAt(0);

   try {
     System.out.println(sumarNumeros(ruta, operador));
```

```java
  } catch (Exception e) {
    // si no existe el archivo o no se puede leer
    System.out.println("No se pudo leer el archivo");
  }
}

public static int sumarNumeros(String ruta, char operador ) throws Exception {
  // Creo un objeto de tipo InputStream para leer el archivo y lanzo una excepción si no se puede leer

  InputStream file = new FileInputStream(ruta);

  //Scanner permite leer el archivo linea por linea
  try (Scanner obj = new Scanner(file)) {

    int resultado = 1;

    while(obj.hasNextLine()){

      String linea = obj.nextLine();

      if(linea.matches("[0-9]+")){
          if(operador == '+'){
              resultado += Integer.parseInt(linea);
          }else if(operador == '*'){
              resultado *= Integer.parseInt(linea);
          }
      }
    }
    if(operador == '+'){
        return resultado -1;
          }
    return resultado;
  }
}
```

```
}
```

Ejercicio 3.

```
package com.mycompany.clase4ejercicios;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.util.Scanner;
import javax.swing.JOptionPane;

/**
*
* @author luciana
*/
public class Ejercicio3 {
   public static void main(String[] args) throws Exception {

       String option = JOptionPane.showInputDialog("Ingrese una opción: \n 1. Codificar \n 2. Decodificar");

       switch (option){
           case "1":
               Codificar();
               break;
           case "2":
               Decodificar();
               break;
           default:
               System.out.println("Opción incorrecta");
               break;
       }
```

```
}

public static String getTexto(String ruta) throws Exception {

    InputStream file = new FileInputStream(ruta);

    try(Scanner obj = new Scanner(file)){
            String texto = "";

            while(obj.hasNextLine()){
                texto += obj.nextLine();
            }

            return texto;
    }

}

public static  String saveTextToFile(String texto, String ruta) throws Exception {

    // Creo un objeto de tipo OutputStream para escribir el archivo y lanzo una excepción si no se puede escribir

    OutputStream file = new FileOutputStream(ruta);

    //Scanner permite leer el archivo linea por linea

    try (PrintWriter obj = new PrintWriter(file)) {
        obj.println(texto);
        return "El archivo se guardo correctamente";
    }

}
```

```java
public static void Codificar() throws Exception {
    String fileToCode = JOptionPane.showInputDialog("Ingrese archivo a codificar. ");
    String inputText ="";

    String fileCoded = JOptionPane.showInputDialog("Ingrese archivo donde guardar el texto codificado. ");

    String inputNum = JOptionPane.showInputDialog("Ingrese paso: ");
    int key = Integer.parseInt(inputNum);

    try {
        inputText = getTexto(fileToCode).toLowerCase();
    } catch (Exception e) {
        System.out.println("No se pudo leer el archivo");

    }

    String coded = "";
    String alpha = "abcdefghijklmnopqrstuvwxyz ";

    for (int i = 0; i < inputText.length(); i++){
        int position = alpha.indexOf(inputText.charAt(i));
        int keyValue= (key + position ) % 27;
        coded += alpha.charAt(keyValue);
    }

    System.out.println(saveTextToFile(coded, fileCoded));
}

public static void Decodificar() throws Exception {
    String fileToDecode = JOptionPane.showInputDialog("Ingrese archivo a codificar. ");
    String inputText ="";

    String fileDecoded = JOptionPane.showInputDialog("Ingrese archivo donde guardar el texto codificado. ");
```

```
        String inputNum = JOptionPane.showInputDialog("Ingrese paso: ");
        int key = Integer.parseInt(inputNum);

        String coded = "";
        String alpha = "abcdefghijklmnopqrstuvwxyz ";

        try {
            inputText = getTexto(fileToDecode).toLowerCase();
        } catch (Exception e) {
            System.out.println("No se pudo leer el archivo");
        }

        for (int i = 0; i < inputText.length(); i++){
            int position = alpha.indexOf(inputText.charAt(i));
            int keyValue= (position - key ) % 27;

            if(keyValue < 0){
                keyValue = alpha.length() + keyValue;
            }
            coded += alpha.charAt(keyValue);

        }

        System.out.println(saveTextToFile(coded, fileDecoded));
    }
}
```

Ejercicio 4.

https://github.com/LucianaCHA/argProg-java