|  |  |  |
| --- | --- | --- |
| **Nombre:** Sergio Aramis Heredia Graham | | **Matrícula:** 2835840 |
| **Nombre del curso:**  Programación orientada a objetos | **Nombre del profesor:**  Wilberth Eduardo Lara Peraza | |
| **Módulo:**  1er. Módulo | **Actividad:**  Reto 3 | |
| **Fecha:** 01/Feb/2024 | | |
| **Bibliografía:** | | |

package com.mycompany.reto3;

import java.util.Scanner;

public class Reto3 {

public static void main(String[] args) {

Scanner scanner = new Scanner (System.in);

System.out.print("""

Elige un programa:

1) Diccionario

2) Generador de arreglo bidimensional

""");

int programa;

do{

System.out.print("Programa (Valores validos 1 o 2): ");

programa = MetodosParte2.solicitarInt();

}while(programa != 1 && programa != 2);

if(programa == 1){

//Programa 1

String[] diccionario = {

"Altruismo",

"Benevolente",

"Conciso",

"Diligencia",

"Efímero",

"Ferviente",

"Gregario",

"Heterogéneo",

"Inexorable",

"Jovial"

};

String[] significado = {

"Desinterés por el bienestar de los demás.",

"Que muestra deseo de ayudar a los demás.",

"Que expresa mucho en pocas palabras.",

"Cuidado y rapidez en realizar algo.",

"Que dura poco tiempo.",

"Que muestra pasión intensa.",

"Que disfruta estando en grupos.",

"Compuesto de elementos diferentes.",

"Que no se puede detener o cambiar.",

"Caracterizado por un espíritu alegre."

};

System.out.print("""

\*\*Diccionario\*\*

A. Enlistar palabras del diccionario

B. Buscar palabras en el diccionario

C. Detalle del diccionario

S. Finalizar programa

""");

char opcion;

do {

opcion = ' ';

while (opcion != 'A' && opcion != 'B' && opcion != 'C' && opcion != 'S'){

System.out.print("\nOpción(Valores validos A, B, C o S): ");

opcion = scanner.next().toUpperCase().charAt(0);

}

switch(opcion){

case 'A':

MetodosParte1.mostrarArreglo(diccionario);

break;

case 'B':

MetodosParte1.mostrarArreglo(diccionario);

MetodosParte1.buscarYMostrarInformacion(diccionario, significado);

break;

case 'C':

System.out.println("\nCantidad de palabras en el diccionario: " + diccionario.length);

MetodosParte1.totalCaracteresEnElementoArreglo(diccionario, significado);

break;

}

}while(opcion != 'S');

System.out.print("Saliendo del programa...");

scanner.close();

}else{

//Programa 2

System.out.print("""

\*\*Programa generador de arreglos bidimensionales con numeros random\*\*

""");

int filas, columnas;

//Se piden los valores de filas y columnas hasta que ambos sean iguales

do{

System.out.println("\n(Las filas y columnas deben de ser iguales)");

System.out.print("\nIndica la cantidad de filas: ");

filas = MetodosParte2.solicitarInt();

System.out.print("\nIndica la cantidad de columnas: ");

columnas = MetodosParte2.solicitarInt();

}while(filas != columnas);

//Se genera un arreglo con los valores dados por el usuario y se llena de numero random del 1 al 100

int[][] arreglo = MetodosParte2.generarArregloBidimensionalRandom(filas, columnas);

//Se muestra el creado arreglo al usuario

MetodosParte2.mostrarArregloBidimensional(arreglo);

//Se cuentan los pares e impares de cada fila y de cada columna

MetodosParte2.contarParesImparesArregloBidimensional(arreglo);

}

}

}

package com.mycompany.reto3;

import java.util.Scanner;

public class MetodosParte1 {

static Scanner scanner = new Scanner(System.in);

public static void mostrarArreglo(String arreglo[]){

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

System.out.println((i + 1) + ". " + arreglo[i]);

}

}

// Método que cumple con la funcionalidad de mostrar una lista de elementos y luego buscar y mostrar la información correspondiente a un elemento seleccionado por el usuario

public static void buscarYMostrarInformacion(String elemento[], String detalle[]){

boolean encontrado; // Si el usuario introduce un valor que este dentro de elementos la variable cambia a true

String elementoBuscado; //Variable de entrada usada para buscar su informacion en detalle[]

do{

encontrado = false;

System.out.print("\nIngresa palabra a buscar: ");

elementoBuscado = scanner.nextLine();

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

if(elemento[i].equalsIgnoreCase(elementoBuscado)){

System.out.println("Significado: " + detalle[i]);

encontrado = true;

break;

}

}

if (!encontrado){

System.out.println("La palabra " + elementoBuscado + " no existe. Intenta nuevamente.");

}

}while(!encontrado);

}

// Metodo que imprime la longitud de cada cadena en dos arreglos elemento y detalle, junto con sus índices.

public static void totalCaracteresEnElementoArreglo(String elemento[], String detalle[]) {

int longitudElemento;

int longitudDetalle;

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

longitudElemento = elemento[i].length();

longitudDetalle = detalle[i].length();

System.out.println((i + 1) + ". " + elemento[i] + " tiene " + longitudElemento + " caracteres, su significado tiene " + longitudDetalle + " caracteres");

}

}

}

package com.mycompany.reto3;

import java.util.Scanner;

import java.util.Random;

public class MetodosParte2 {

static Scanner scanner = new Scanner(System.in);

// Metodo para solicitar int sin que de error

public static int solicitarInt(){

while (true) {

String intValue = scanner.next();

try {

return Integer.parseInt(intValue);

}catch (NumberFormatException e){

System.out.println("\nError: No has introducido un valor valido\n");

// No es necesario hacer nada aquí, el bucle continuará automáticamente

}

}

}

// Metodo para generar arreglos bidimensionales con dimensiones dadas por el usuario y genera valores random del 1 al 10

public static int[][] generarArregloBidimensionalRandom(int filas, int columnas){

Random random = new Random();

int[][] arreglo = new int[filas][columnas];

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

for(int j=0;j < columnas; j++){

arreglo[i][j] = random.nextInt(100)+1;

}

}

return arreglo;

}

//Metodo para mostrar el arreglo proporsionado

public static void mostrarArregloBidimensional(int[][] arreglo){

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

for(int j=0;j < arreglo[i].length;j++){

System.out.print(String.format("%4d", arreglo[i][j])); // Formatea cada número para que ocupe un mínimo de 4 espacios

}

System.out.println();

}

}

// Metodo que cuenta e imprime la cantidad de pares e impares de cada fila y de cada columna

public static void contarParesImparesArregloBidimensional(int[][] arreglo){

int pares;

int impares;

System.out.println("-------------------------------------------------");

//Pares e impares de filas

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

pares = 0;

impares = 0;

for(int j=0;j < arreglo[i].length;j++){

if(arreglo[i][j] % 2 == 0){

pares++;

}else{

impares++;

}

}

System.out.println("Cantidad de pares: " + pares + ", impares: " + impares + " en la fila " + (i+1));

}

System.out.println("-------------------------------------------------");

//Pares e impares de columnas

for(int i=0;i < arreglo[0].length;i++){

pares = 0;

impares = 0;

for(int j=0;j < arreglo.length;j++){

if(arreglo[j][i] % 2 == 0){

pares++;

}else{

impares++;

}

}

System.out.println("Cantidad de pares: " + pares + ", impares: " + impares + " en la columna " + (i+1));

}

System.out.println("-------------------------------------------------");

}

}