|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODULO** | MF0966\_3: Consulta y manipulación de informacióncontenida en gestores de datos | | | |
| **UNIDAD FORMATIVA** | UF2216 Herramientas de los gestores de Bases de Datos. Pasarelas y medios de conexión | | | |
| **NOMBRE DE LA ACTIVIDAD** | Repaso Listas | | NÚMERO DE ACTIVIDAD | 14 |
| **Nombre y apellidos del alumno:** | |  | | |

**DESCRIPCIÓN DE LA ACTIVIDAD:**

1**.-** Crea una clase llamada “Ciudad” con 2 atributos, nombre y temperatura. Crea su construtor con todos los atributos y sus getters y setters.

2.- Crea una clase llamada “Ejecutor”. Crea una lista de Ciudades:

List<Ciudad> ciudades = new ArrayList<>();

ciudades =List.of(new Ciudad("Londres",30),new Ciudad("Paris",40)

,new Ciudad("Madrid",35),new Ciudad("Las Palmas",25));

3.- Crea los siguientes métodos en la clase “Ejecutor”.

a. Recibe una lista de ciudades y devuelve la ciudad más calurosa.

b. Recibe una lista de ciudades y una temperatura. Devuelve un conjunto con las ciudades cuya temperatura sea inferior a dicho parámetro.

c. Recibe una lista de ciudades y devuelve la temperatura media.

d. Recibe una lista de ciudades y muestra su nombre y su temperatura.

package model;

public class Ciudad {

private String nombre;

private double temperatura;

public Ciudad(String nombre, double temperatura) {

super();

this.nombre = nombre;

this.temperatura = temperatura;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public double getTemperatura() {

return temperatura;

}

public void setTemperatura(double temperatura) {

this.temperatura = temperatura;

}

}

package view;

import java.util.ArrayList;

import java.util.Comparator;

import java.util.List;

import model.Ciudad;

public class Ejecutor {

public static void main(String[] args) {

List<Ciudad> ciudades = new ArrayList<>();

ciudades = List.of(new Ciudad("Londres",30),new Ciudad("Paris",40)

,new Ciudad("Madrid",35),new Ciudad("Las Palmas",25)

,new Ciudad("Valsequillo",20)

,new Ciudad("Valsendero",15));

//Recibe una lista de ciudades y devuelve la ciudad más calurosa.

//Programación funcional

System.out.println("La ciudad mas calurosa es: " +

ciudades.stream()

.max(Comparator.comparingDouble(Ciudad::getTemperatura))

.orElse(null).getNombre());

System.out.println("La ciudad mas calurosa es: " +

MasCalurosa(ciudades).getNombre());

//Recibe una lista de ciudades y una temperatura.

//Devuelve un conjunto con las ciudades cuya temperatura

//sea inferior a dicho parámetro.

List<Ciudad> ciudadesFrias = temperaturaInferior(ciudades,25.0);

System.out.println("Las ciudades más frías son: ");

temperaturaInferior(ciudades,15.0) //Devuelve una lista

.forEach(n->System.out.println(n.getNombre()));

// ciudadesFrias.forEach(n->System.out.println(n.getNombre()));

System.out.println("Las ciudades inferiores a 35ºC son:");

ciudades.stream().filter(c -> c.getTemperatura() < 35)

.forEach(c -> System.out.println("\*" + c.getNombre()));

System.out.println("La temperatura media es: "+temperaturaMedia(ciudades));

ciudades.forEach(ciudad->{

System.out.println("La ciudad es: "+ciudad.getNombre());

System.out.println("La temperatura es: "+ciudad.getTemperatura());

});

ciudadesFrias.forEach(ciudad->{

System.out.println("La ciudad es: "+ciudad.getNombre());

System.out.println("La temperatura es: "+ciudad.getTemperatura());

});

}

//Recibe una lista de ciudades y devuelve la ciudad más calurosa.

//Programación tradicional

public static Ciudad MasCalurosa(List<Ciudad> ciudades) {

Ciudad ciudadMasCalurosa = null;

double maxTemperatura = -100;

for (Ciudad ciudad : ciudades) {

if (ciudad.getTemperatura() > maxTemperatura) {

maxTemperatura = ciudad.getTemperatura();

ciudadMasCalurosa = ciudad;

}

}

return ciudadMasCalurosa;

}

//Recibe una lista de ciudades y una temperatura.

//Devuelve un conjunto con las ciudades cuya temperatura

//sea inferior a dicho parámetro.

public static List<Ciudad> temperaturaInferior(List<Ciudad> ciudades, double temperaturaLimite) {

List<Ciudad> ciudadesFrias = new ArrayList<>();

for (Ciudad ciudad : ciudades) {

if (ciudad.getTemperatura() < temperaturaLimite) {

ciudadesFrias.add(ciudad);

}

}

return ciudadesFrias;

}

// Recibe una lista de ciudades y devuelve la temperatura media.

public static double temperaturaMedia(List<Ciudad> ciudades) {

double suma=0;

for (Ciudad ciudad : ciudades) {

suma+=ciudad.getTemperatura();

}

try {

suma/=ciudades.size();

}

catch (Exception e) {

System.out.println(e.getMessage());

return 0.0;

}

return suma;

}

}