

Programación Avanzada I. Práctica 5.3

Tema 5. Comparadores, Colecciones y Correspondencias Ordenadas

Ejercicio 1. (Proyecto pa1p53)

Se va a crear una aplicación para manipular las alturas medias de la población de los distintos países del mundo. Para ello, se debe definir las clases **Pais**, **Mundo** y **CompAltura** en el paquete **alturas**. Todos los métodos que se piden se considerarán públicos y todas las variables de instancia privadas.

Clase Pais

La clase **Pais** mantiene información de un país, el continente al que pertenece y la altura media de sus habitantes. Así un país tendrá un nombre de país (**String nombre**), un nombre de continente (**String continente**), y la altura media de sus habitantes (**double altura**).

- Define un constructor que crea un país con la información del nombre del país, el continente y la altura media.
- Define métodos *getter* para las tres variables de instancia.
- Un país es igual a otro si sus nombres son iguales, considerando que las letras mayúsculas y minúsculas son diferentes.
- Un país se considera mayor que otro si su nombre lo es alfabéticamente.
- Un país se representará así: **Pais(nombre, continente, altura)**

Clase Mundo

La clase **Mundo** mantiene información de los países del mundo mediante una lista de países (**List<Pais> paises**), y define los siguientes constructores y métodos:

- Un constructor que inicializa correctamente la estructura.
- public List<Pais> getPaises()**
Devuelve la lista de países almacenados.
- public void cargar(String fich) throws FileNotFoundException;**
Lee y almacena los países del fichero cuyo nombre se recibe como argumento. El fichero tiene el siguiente formato por cada línea (consultar el fichero de ejemplo **alturas.txt**):

nombrePaís, continente, altura

Si una línea del fichero no es correcta, se ignora y se sigue con la siguiente.

d) `public static <K,V> void presentaEnConsola(Map<K,V> map);`

Recibe una correspondencia y muestra su clave y valor, una por línea.

e) `public Map<String, Integer> numeroDePaisesPorContinente();`

Devuelve una correspondencia que asocia cada continente con el número de países que contiene.

f) `public Map<Double, List<Pais>> paisesPorAltura();`

Devuelve una correspondencia que asocia cada altura con una lista de países que tienen esa altura media. Las alturas se deben **truncar** a un solo decimal.

g) `public ... paisesPorContinente();`

Devuelve una correspondencia ordenada por el orden natural, en la que cada continente se asocia con un conjunto de países que contiene, ordenados por el orden natural.

Clase CompAltura

Definir la clase `CompAltura` que actúe como un comparador de países por altura, de menor a mayor.

Clase Mundo

Añada los siguientes métodos a la clase `Mundo`:

h) `public ... paisesOrdenadosPorAltura();`

Devuelve un conjunto de países ordenados por altura.

i) `public ... paisesPorContinenteAltura();`

Devuelve una correspondencia ordenada por el orden natural, en la que cada continente se asocia con un conjunto de países que contiene, ordenados por altura.

j) `public ... paisesPorContinenteAlturaDec();`

Devuelve una correspondencia ordenada por el orden natural, en la que cada continente se asocia con un conjunto de países que contiene, ordenados decrecientemente por altura.

k) `public ... paisesPorInicial();`

Devuelve una correspondencia ordenada por el orden natural, en la que cada letra se asocia con el conjunto ordenado, según el orden natural, de paises que comienzan por dicha letra inicial.

l) `public ... mediaPorContinente();`

Devuelve una correspondencia ordenada por el orden natural, en la que cada continente se asocia con la altura media de entre los paises que pertenecen a cada continente. Se debe usar el método `paisesPorContinente()`.

m) `public ... continentesConMasPaises();`

Devuelve una lista con los nombres de los continentes que tienen el mayor número de paises. Se debe usar el método `numeroDePaisesPorContinente()`.

Clase Main

La clase Main se definirá en el paquete por defecto (anónimo) según el siguiente código:

```
import alturas.Mundo;
import java.io.IOException;
public class Main {
    public static void main(String[] args) throws IOException {
        Mundo ps = new Mundo();
        ps.cargar("data/alturas.txt");
        System.out.println("\nPaises por altura");
        Mundo.presentaEnConsola(ps.paisesPorAltura());
        System.out.println("\nNúmero de Paises por continente");
        Mundo.presentaEnConsola(ps.numeroDePaisesPorContinente());
        System.out.println("\nPaises por continente");
        Mundo.presentaEnConsola(ps.paisesPorContinente());
        System.out.println("\nConjunto de Paises ordenados por altura");
        System.out.println(ps.paisesOrdenadosPorAltura());
        System.out.println("\nPaises por continente ordenado por altura");
        Mundo.presentaEnConsola(ps.paisesPorContinenteAltura());
        System.out.println("\nPaises por continente ordenados decrecientes en altura");
        Mundo.presentaEnConsola(ps.paisesPorContinenteAlturaDec());
        System.out.println("\nPaises por inciales");
        Mundo.presentaEnConsola(ps.paisesPorInicial());
        System.out.println("\nMedias de alturas por continente");
        Mundo.presentaEnConsola(ps.mediaPorContinente());
        System.out.println("\nContinentes con mas paises");
        System.out.println(ps.continentesConMasPaises());
    }
}
```

cuya ejecución producirá la siguiente salida (partes eliminadas y saltos de línea incluidos por legibilidad):

```
Países por altura
1.5 [Pais(Indonesia, Asia, 1.58)]
1.6 [Pais(Bahrain, Asia, 1.651), Pais(Bolivia, South America, 1.6), ..., Pais(Uruguay, South America, 1.7), Pais(Uzbekistan, Asia, 1.754)]
1.8 [Pais(Bosnia & Herzegovina, Europe, 1.839), Pais(Croatia, Europe, 1.805), ..., Pais(Serbia, Europe, 1.82), Pais(Slovenia, Europe, 1.803)]

Número de Países por continente
Africa 15
Asia 32
Euro/Asia 1
Europe 34
North America 6
Oceania 3
South America 10

Países por continente
Africa [Pais(Algeria, Africa, 1.722), Pais(Cameroon, Africa, 1.706), ..., Pais(South Africa, Africa, 1.68), Pais(Tunisia, Africa, 1.723)]
Asia [Pais(Azerbaijan, Asia, 1.718), Pais(Bahrain, Asia, 1.651), ..., Pais(Uzbekistan, Asia, 1.754), Pais(Vietnam, Asia, 1.621)]
Euro/Asia [Pais(Turkey, Euro/Asia, 1.74)]
Europe [Pais(Albania, Europe, 1.74), Pais(Austria, Europe, 1.792), ..., Pais(Ukraine, Europe, 1.765), Pais(United Kingdom, Europe, 1.77)]
North America [Pais(Canada, North America, 1.751), Pais(Cuba, North America, 1.68), ..., Pais(Panama, North America, 1.65), Pais(United States, North America, 1.763)]
Oceania [Pais(Australia, Oceania, 1.756), Pais(New Zealand, Oceania, 1.77), Pais(Tonga, Oceania, 1.761)]
South America [Pais(Argentina, South America, 1.745), Pais(Bolivia, South America, 1.6), ..., Pais(Uruguay, South America, 1.7), Pais(Venezuela, South America, 1.69)]

Conjunto de Países ordenados por altura
[Pais(Indonesia, Asia, 1.58), Pais(Bolivia, South America, 1.6), ..., Pais(Netherlands, Europe, 1.838), Pais(Bosnia & Herzegovina, Europe, 1.839)]

Países por continente ordenado por altura
Africa [Pais(Nigeria, Africa, 1.638), Pais(Malawi, Africa, 1.66), ..., Pais(Morocco, Africa, 1.727), Pais(Senegal, Africa, 1.75)]
Asia [Pais(Indonesia, Asia, 1.58), Pais(Philippines, Asia, 1.619), ..., Pais(Israel, Asia, 1.77), Pais(Russia, Asia, 1.772)]
Euro/Asia [Pais(Turkey, Euro/Asia, 1.74)]
Europe [Pais(Malta, Europe, 1.699), Pais(Romania, Europe, 1.72), ..., Pais(Netherlands, Europe, 1.838), Pais(Bosnia & Herzegovina, Europe, 1.839)]
North America [Pais(Panama, North America, 1.65), Pais(Mexico, North America, 1.67), ..., Pais(Canada, North America, 1.751), Pais(United States, North America, 1.763)]
Oceania [Pais(Australia, Oceania, 1.756), Pais(Tonga, Oceania, 1.761), Pais(New Zealand, Oceania, 1.77)]
South America [Pais(Bolivia, South America, 1.6), Pais(Ecuador, South America, 1.635), ..., Pais(Brazil, South America, 1.731), Pais(Argentina, South America, 1.745)]

Países por continente ordenados decrecientes en altura
Africa [Pais(Senegal, Africa, 1.75), Pais(Morocco, Africa, 1.727), ..., Pais(Malawi, Africa, 1.66), Pais(Nigeria, Africa, 1.638)]
Asia [Pais(Russia, Asia, 1.772), Pais(Israel, Asia, 1.77), ..., Pais(Philippines, Asia, 1.619), Pais(Indonesia, Asia, 1.58)]
Euro/Asia [Pais(Turkey, Euro/Asia, 1.74)]
Europe [Pais(Bosnia & Herzegovina, Europe, 1.839), Pais(Netherlands, Europe, 1.838), ..., Pais(Romania, Europe, 1.72), Pais(Malta, Europe, 1.699)]
North America [Pais(United States, North America, 1.763), Pais(Canada, North America, 1.751), ..., Pais(Mexico, North America, 1.67), Pais(Panama, North America, 1.65)]
Oceania [Pais(New Zealand, Oceania, 1.77), Pais(Tonga, Oceania, 1.761), Pais(Australia, Oceania, 1.756)]
South America [Pais(Argentina, South America, 1.745), Pais(Brazil, South America, 1.731), ..., Pais(Ecuador, South America, 1.635), Pais(Bolivia, South America, 1.6)]

Países por inciales
A [Pais(Albania, Europe, 1.74), Pais(Algeria, Africa, 1.722), ..., Pais(Austria, Europe, 1.792), Pais(Azerbaijan, Asia, 1.718)]
B [Pais(Bahrain, Asia, 1.651), Pais(Belgium, Europe, 1.786), ..., Pais(Brazil, South America, 1.731), Pais(Bulgaria, Europe, 1.752)]
C [Pais(Cambodia, Asia, 1.625), Pais(Cameroon, Africa, 1.706), ..., Pais(Cuba, North America, 1.68), Pais(Czech Republic, Europe, 1.803)]
D [Pais(Denmark, Europe, 1.826)]
E [Pais(Ecuador, South America, 1.635), Pais(Egypt, Africa, 1.703), Pais(Estonia, Europe, 1.791)]
F [Pais(Finland, Europe, 1.789), Pais(France, Europe, 1.756)]
G [Pais(Gambia, Africa, 1.68), Pais(Germany, Europe, 1.81), Pais(Ghana, Africa, 1.695), Pais(Greece, Europe, 1.783)]
H [Pais(Hong Kong, Asia, 1.734), Pais(Hungary, Europe, 1.765)]
I [Pais(Iceland, Europe, 1.81), Pais(India, Asia, 1.647), ..., Pais(Italy, Europe, 1.765), Pais(Ivory Coast, Africa, 1.701)]
J [Pais(Jamaica, North America, 1.718), Pais(Japan, Asia, 1.707)]
K [Pais(Kazakhstan, Asia, 1.69), Pais(Korea North, Asia, 1.656), Pais(Korea South, Asia, 1.735)]
L [Pais(Lebanon, Asia, 1.76), Pais(Libya, Africa, 1.713), Pais(Lithuania, Europe, 1.772), Pais(Luxembourg, Europe, 1.799)]
M [Pais(Malawi, Africa, 1.66), Pais(Malaysia, Asia, 1.663), ..., Pais(Montenegro, Europe, 1.832), Pais(Morocco, Africa, 1.727)]
N [Pais(Nepal, Asia, 1.63), Pais(Netherlands, Europe, 1.838), ..., Pais(Nigeria, Africa, 1.638), Pais(Norway, Europe, 1.824)]
P [Pais(Pakistan, Asia, 1.693), Pais(Panama, North America, 1.65), ..., Pais(Poland, Europe, 1.787), Pais(Portugal, Europe, 1.737)]
Q [Pais(Qatar, Asia, 1.708)]
R [Pais(Romania, Europe, 1.72), Pais(Russia, Asia, 1.772)]
S [Pais(Saudi Arabia, Asia, 1.689), Pais(Senegal, Africa, 1.75), ..., Pais(Switzerland, Europe, 1.754), Pais(Syria, Asia, 1.73)]
T [Pais(Taiwan, Asia, 1.714), Pais(Thailand, Asia, 1.703), ..., Pais(Turkey, Euro/Asia, 1.74), Pais(Turkministan, Asia, 1.72)]
U [Pais(UAE, Asia, 1.734), Pais(Ukraine, Europe, 1.765), ..., Pais(Uruguay, South America, 1.7), Pais(Uzbekistan, Asia, 1.754)]
V [Pais(Venezuela, South America, 1.69), Pais(Vietnam, Asia, 1.621)]

Medias de alturas por continente
Africa 1.7020666666666666
Asia 1.6898750000000002
Euro/Asia 1.74
Europe 1.7841176470588234
North America 1.7053333333333331
Oceania 1.7623333333333333
South America 1.6847

Continentes con mas paises
[Europe]
```