

Ejercicios de estructuras de datos (parte 3)

ÍNDICE

8. Maps	1
1. Países y capitales	1
2. Teléfonos	1
3. Teléfonos ampliado	1
4. Terminaciones	1
5. Palabras	1
6. Apellidos	2
9. HashSet	2
1. Incremento	2
2. Eliminar duplicados	2
3. Imprimir duplicados	2
4. Unión	2
5. Intersección	2
6. Elementos únicos	3
10. Stream	3
1. Año nacimiento	3
2. Apellido	3
3. Ordenar	3
4. Imprimir	3
5. Negativo	3
6. Personas	3

8. Maps

1. Países y capitales

Declara un HashMap que almacene el país y la capital de varios países Europeos. Luego realiza un programa que pida un País al usuario y muestre su capital.

2. Teléfonos

Queremos tener una guía de teléfonos que asocie un número de teléfono a un contacto. El programa debe pedir un contacto y mostrar su número asociado.

3. Teléfonos ampliado

Amplía el ejercicio anterior para que un contacto pueda tener más de un teléfono.

4. Terminaciones

Haz un programa para calcular cuántas veces ha salido una terminación (último número) en un sorteo de lotería a partir de un array con varios números de décimos. La salida debe ser un Map. La clave será la terminación y el valor será el número de veces que se repite esa terminación. Ejemplos:

- Entrada: "00004", "03847", "39804" salida: {4=2, 7=1}
- Entrada: "58975", "25894", "52985", "98598" salida: {4=1, 5=2, 8=1}

5. Palabras

Pide por pantalla las palabras que quieras (palabra vacía para terminar) y calcula las veces que se repite de cada una.

Ejemplos:

- Entrada: uno dos dos tres tres tres
- Salida:
 - uno - 1
 - dos - 2
 - tres - 3

6. Apellidos

A partir de un fichero csv que contiene apellidos y la frecuencia con la que se repiten en España, realiza un programa que pida por pantalla un apellido y muestre la frecuencia de éste. El programa finalizará cuando el usuario introduzca una línea vacía.

9. HashSet

1. Incremento

Haciendo uso de un Iterator crea un método estático que acepte un HashSet de enteros como entrada y lo retorne modificado incrementando en uno cada valor del HashSet original.

2. Eliminar duplicados

Crea un método estático que reciba un array de tipo String y lo imprima eliminando los elementos duplicados.

3. Imprimir duplicados

Crea un método estático que recibe un array de tipo String e imprime únicamente, y una sola vez, aquellos elementos que aparecen duplicados.

4. Unión

Crea un método estático que recibe dos conjuntos (tipo Set) como parámetros y devuelve el conjunto resultante de la unión de estos.

5. Intersección

Crea un método estático que recibe dos conjuntos como parámetros y devuelve el conjunto resultante de la intersección de estos.

6. Elementos únicos

Crea un método estático que recibe dos conjuntos como parámetros y devuelve el conjunto que contiene todos los elementos que sólo están presentes en uno de los dos conjuntos originales.

10. Stream

1. Año nacimiento

A partir de una lista de Personas. Imprime cuántas nacieron antes de 1970.

2. Apellido

A partir de una lista de Personas. Imprime cuántas tienen su primer apellido que empiece por A.

3. Ordenar

A partir de una lista de Personas. Imprime los nombres que son únicos en orden alfabético.

4. Imprimir

Escribe un programa que lea la entrada del usuario como cadenas. Cuando el usuario ingresa una cadena vacía (solo presiona enter), la lectura de entrada se detendrá y el programa imprimirá todas las entradas del usuario mediante el uso de streams.

5. Negativo

Escribe un programa que lea la entrada del usuario. Cuando el usuario introduzca un número negativo, la lectura de entrada se detendrá. Después de esto, imprime todos los números que el usuario ha dado como entrada que están entre 1 y 5 con el uso de streams.

6. Personas

Crea un programa que vaya leyendo la entrada desde el teclado y cree objetos de la clase Person (tienes el código más abajo). Para finalizar se introducirá una cadena vacía. Ahora imprime los lastNames únicos en orden alfabético.

```
public class Person {  
    private String firstName;  
    private String lastName;  
    private int birthYear;  
  
    public Person(String firstName, String lastName, int birthYear){  
        this.firstName = firstName;  
        this.lastName = lastName;  
        this.birthYear = birthYear;  
    }  
  
    public String getFirstName() {  
        return this.firstName;  
    }  
  
    public String getLastName() {  
        return this.lastName;  
    }  
  
    public int getBirthYear() {  
        return this.birthYear;  
    }  
}
```