

Ejercicios Java

Colecciones II

INSTRUCCIONES PARA RESOLVER LOS EJERCICIOS	2
EJERCICIO 1.....	2
EJERCICIO 2.....	2
EJERCICIO 3.....	2
EJERCICIO 4.....	2
EJERCICIO 5.....	2
EJERCICIO 6.....	2
EJERCICIO 7.....	2
EJERCICIO 8.....	3
EJERCICIO 9.....	3
EJERCICIO 10.....	3
EJERCICIO 11.....	3
EJERCICIO 12.....	3
EJERCICIO 13.....	4

Instrucciones para resolver los ejercicios

Crea una clase llamada `Colecciones` y define en ella los métodos **de clase** que se piden en cada ejercicio.

Ejercicio 1

Método llamado `eliminarLasDeLongitudPar` que acepte un conjunto de cadenas y elimine del mismo todas aquellas cadenas de una longitud par. No se permite el uso de ninguna otra estructura de datos auxiliar.

Ejercicio 2

Método llamado `contieneImpares` que acepte un conjunto de números enteros y retorne `true` si contiene al menos un número impar o `false` en caso contrario.

Ejercicio 3

Método llamado `valoresUnicos` que acepte un mapa de cadenas a cadenas y retorne `false` si dos o más claves se asocian a un mismo valor o `true` en caso contrario.

Ejercicio 4

Método llamado `algunaSeRepiteAlMenos3Veces` que acepte una lista de cadenas como parámetro y retorne verdadero si alguna cadena se repite al menos 3 veces en la lista o falso en caso contrario.

Resolver el problema utilizando un mapa como almacenamiento auxiliar.

Ejercicio 5

Método llamado `negativosAbajoPositivosArriba` que acepte una pila de números enteros como parámetro y la reorganice para que se queden debajo los negativos y encima los positivos. Usa una cola como estructura de datos auxiliar para realizar el proceso.

Ejemplo:

Entrada: `[23, -5, 37, 29, -19, -55, 89]` ← Tope de la pila

Salida: `[-55, -19, -5, 23, 89, 29, 37]` ← Tope de la pila

Ejercicio 6

Método llamado `moda` que acepte una lista de números enteros como parámetro y retorne el número que se repite con mayor frecuencia (la moda) o 0 si la lista está vacía.

Ejercicio 7

Método llamado `eliminarSiMayoresEncima` que acepte una pila de números enteros como parámetro y elimine de ella los elementos que tienen por encima alguno cuyo valor sea mayor.

Por ejemplo, suponiendo que una pila almacena los siguientes números:

`[6, 22, 17, 17, 10, 7, 9, 14, 5, 12, 7, 2]` ← Tope

Los números 5, 9, 7, 10 y 6 deben eliminarse porque todos ellos tienen algún número mayor por encima.

Sólo se permite el uso de otras pilas o colas para solucionar este problema.

Ejercicio 8

Método llamado `interseccion` que acepte dos mapas, ambos de `String` a `Integer`, y retorne un nuevo mapa que contenga la intersección de los dos primeros. Por ejemplo, si recibe los mapas:

```
{Fernando=53, Manuela=29, Ana=41, Luis=65, Mario=33, Adrián=21, Carmen=39, Elena=19}
{Valentina=37, Ana=41, Mario=33, Benito=67, Carmen=39, Ramón=44, Elena=19, Hugo=32}
```

retornará el mapa:

```
{Ana=41, Mario=33, Carmen=39, Elena=19}
```

Ejercicio 9

Método llamado `valorMenosRepetido` que acepte un mapa cuyas claves de tipo `String` representan nombres de personas, y cuyos valores de tipo `Integer` representan sus edades. El método retornará la edad que menos se repite dentro del mapa. En caso de empate, retornará el valor menor.

Por ejemplo, si el mapa contiene:

```
{Juan=33, Hugo=29, Ana=45, Luis=47, Mario=33, Rosa=29, Carmen=33, Elena=59, Benito=33}
```

El método retornará 45, que es el valor más pequeño de los dos que menos se repiten (45 y 59 se repiten una sola vez).

Ejercicio 10

Método llamado `cuentaPares` que acepte una lista de cadenas que representan palabras individuales y retorne una estructura de datos en la que se almacene el resultado de contar el número de ocurrencias de cada secuencia de dos caracteres. Por ejemplo, dada la lista:

```
[banana, pera, melón, o, sandía]
```

- La palabra banana contiene los pares `ba`, `an`, `na`, `an`, `na`.
- La palabra pera contiene los pares `pe`, `er`, `ra`.
- La palabra melón contiene los pares `me`, `el`, `lón`.
- La palabra o no contiene secuencias de dos caracteres por ser de longitud uno.
- La palabra sandía contiene los pares `sa`, `an`, `nd`, `dí`, `ía`.

Por tanto, el par `ba` se repite una vez, el par `an` tres veces, el par `na` dos veces, etc.

Ejercicio 11

Método llamado `contarComunes` que acepte dos listas como parámetros y retorne la cantidad de números que tienen en común sin contar los repetidos.

Ejercicio 12

Método llamado `duplicar` que acepte una pila de números enteros como parámetro y sustituya cada valor almacenado por el mismo duplicado, manteniendo el orden original.

Por ejemplo, dada la pila:

```
[7, 6, 2, 9, 5] ← Tope
```

Almacenará en ella los valores:

```
[7, 7, 6, 6, 2, 2, 9, 9, 5, 5] ← Tope
```

Ejercicio 13

Método llamado `colapsar` que acepte un parámetro de tipo `Deque` que represente una pila de números enteros y modifique su contenido reemplazando cada par de números consecutivos por la suma de ambos.

Por ejemplo:

Contenido inicial: $[5, 11, -3, 14, 9, 4, 21, -42, 13, 2] \leftarrow \text{Tope}$

Contenido final: $[16, 11, 13, -21, 15] \leftarrow \text{Tope}$

Si la pila original almacena un número impar de números, el número que está en el fondo de la pila se quedará como está. Por ejemplo:

Contenido inicial: $[3, -5, 7, -9, 11] \leftarrow \text{Tope}$

Contenido final: $[3, 2, 2] \leftarrow \text{Tope}$

Se han de respetar las restricciones siguientes para resolver el problema:

- No está permitido usar ningún método de iteración.
- No está permitido el uso del método `stream()` de ninguna colección.
- Si fuese necesario usar estructuras de datos auxiliares para resolver el problema, sólo se permitirán pilas y/o colas.
- Sólo se permite el uso de las operaciones propias de pilas y colas simples.