



## TAREA 2 DEL BLOQUE DE CONTENIDOS 8 y ejemplo de uso de una pila y un ArrayList

Ejemplo de uso de una pila y de un ArrayList en el ejercicio “143 Tortitas” de Acepta el reto.

### Enunciado:

La especialidad de Michelón es hacer tortitas. En la cocina de su restaurante hace a diario muchísimas y las va colocando una encima de otra según las saca de la plancha. Le gusta ver variedad en su pila de tortitas, por lo que va haciéndolas de tamaños distintos.

Como la tortita que está más arriba se enfría mucho más rápido que el resto, de vez en cuando mete la espátula entre dos tortitas de la torre y las da la vuelta, de forma que la tortita que estaba encima queda ahora en medio, y la tortita que estaba en medio, queda la primera.

Tras una serie de volteos, ¿serías capaz de decir el tamaño de la tortita que queda más arriba?

## Entrada

Cada caso de prueba contiene dos líneas. La primera describe la situación inicial de la pila de tortitas. Contiene una sucesión de números terminada con  $-1$  que indica los tamaños de cada una de las tortitas, empezando por el tamaño de la situada encima de la mesa y terminando por la de más arriba. Todos los tamaños son positivos, por lo que el  $-1$  final no debe considerarse.

Una segunda línea describe los volteos que hace Michelón. El primer número de la línea indica el número de volteos que hace. Le sigue un número por cada movimiento, indicando el número de tortitas totales que coge de golpe. Se garantiza que el número siempre es válido (es decir, está entre 0 y el número total de tortitas).

La entrada termina con un caso de prueba sin tortitas.

## Salida

Por cada caso de prueba escribe el tamaño de la tortita que queda arriba tras los movimientos de Michelón.

## Entrada de ejemplo

```
5 4 3 2 1 -1
0
5 4 3 2 1 -1
2 3 2
5 4 3 2 1 -1
1 5
-1
0
```

## Salida de ejemplo

```
1
2
5
```



### Solución:

```
import java.util.ArrayList;
import java.util.Scanner;
import java.util.Stack;

public class Tortitas {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int tam;
        int numVolteos;
        int numTortitas;
        int[] v;
        ArrayList<Integer> lista = new ArrayList();
        Stack<Integer> pila = new Stack();
        tam = teclado.nextInt();
        while (tam >= 0) {
            pila.push(tam);
            tam = teclado.nextInt();
            while (tam >= 0) {
                pila.push(tam);
                tam = teclado.nextInt();
            }

            numVolteos = teclado.nextInt();
            for (int i = 0; i < numVolteos; i++) {
                numTortitas = teclado.nextInt();
                lista.clear();
                for (int j = 0; j < numTortitas; j++) {
                    lista.add(pila.pop());
                }
                for (int j = 0; j <= lista.size() - 1; j++) {
                    pila.push(lista.get(j));
                }
            }
            System.out.println(pila.pop());
            tam = teclado.nextInt();
            pila.clear();
        }
    }
}
```



## TAREA 2

### 1) Programar una clase de nombre Coche.

La clase tendrá dos propiedades privadas:

- velocidad: de tipo entero, que almacenará la velocidad a la que va el coche.
- marcha: de tipo entero. Será un valor comprendido entre 0 y 6, reflejando el 0 el punto muerto, y el resto de los valores la marcha que se tiene seleccionada.

La clase tendrá los siguientes métodos:

- Constructor por defecto (no se le pasan argumentos), que dé el valor 0 a las dos propiedades.
- Constructor al que se le pasen como argumentos valores para las dos propiedades.
- Métodos set y get para todas las propiedades (setVelocidad, getVelocidad, setMarcha y GetMarcha).
- Método acelera, al que se le pasa como argumento un valor entero, e incrementa la velocidad en el valor indicado.
- Método frena, que decrementa la velocidad en el valor indicado.
- Metodo subeMarcha, que incrementa en 1 la marcha que llevamos. Si vale 6 no se modifica.
- Método reduceMarcha. Que reduce en 1 la marcha. Si vale 0 no se modifica.

Esta clase tiene que estar en un paquete de nombre vehiculo.

### 2) Hacer un programa en el que se declare una tabla de dimensiones 5 x 5 en la que se guardaran objetos de la clase Coche. Darle valores a los objetos de la tabla utilizando el siguiente criterio:

Asignar a la velocidad valores aleatorios comprendidos entre 10 y 120, ambos incluidos (serán valores enteros).  
Asignar un valor aleatorio para la marcha comprendido entre 0 y 6.

Guardar en un ArrayList los objetos que tengan una velocidad superior a 60, y en otro ArrayList los objetos con una velocidad menor o igual a 40. Visualizar los valores de la tabla y de los ArrayList (ordenados de menor a mayor velocidad). Mirar en la página 5 del documento adjunto “consejos competición programame”, cómo se ordena un ArrayList de objetos.

Esta clase tiene que estar en un paquete de nombre tabla.

### 3) Hacer un programa en el que se declare un vector de 50 valores de tipo entero. Asignarle valores aleatorios comprendidos entre 1 y 20, ambos incluidos. Escribir a continuación:

- a) El contenido del vector, visualizando 10 números por línea.
- b) El número más grande y el más pequeño.

Guardar en una colección los valores pares del vector sin repeticiones, es decir, si el valor 16 aparece 4 veces solo almacenaremos un 16. Podéis utilizar un ArrayList o un TreeSet, aunque este último os facilita el trabajo. Escribir a continuación estos valores ordenados de menor a mayor.

### 4) Hacer el ejercicio “146 Números afortunados” de Acepta el reto, donde resulta muy útil el uso de un ArrayList.

**Puntuación de cada ejercicio:**



Ejercicio 1: 1,5 puntos

Ejercicio 2: 3 puntos

Ejercicio 3: 3 puntos

Ejercicio 4: 2,5 puntos

### **Cómo enviar la tarea:**

Enviar un único fichero, en cualquier formato excepto pdf. El documento tiene que incluir el código fuente de todos los ejercicios y una captura de pantalla del ejercicio 4, donde se vea el veredicto de "Acepta el reto". El nombre de fichero tiene que estar formado por vuestro nombre y apellidos.