package pila.numerico;

/\*\*

\* M�todos para manejar una pila.

\*

\* C�digo del libro:

\* Data Structures & Algorithms in Java 2nd Ed., Robert Lafore, 2003.

\*

\* @version 1.00 15/08/11

\*/

class Pila {

private int MAXIMO; // tama�o de la pila

private long[] arreglo;

private int top; // cima o parte superior de la pila

// --------------------------------------------------------------

public Pila(int t) { // constructor

MAXIMO = t; // asigna el tama�o del arreglo

arreglo = new long[MAXIMO]; // crea arreglo

top = -1; // todavia no tiene elementos

}

// --------------------------------------------------------------

public void **inserta**(long e) { **----1** // inserta un elemento en la cima de la pila

if (!estaLlena()) **------------2**

arreglo[++top] = e; // -----1**O(n) incrementa top, inserta elemento**

else **--------------------------1**

System.out.println("No se puede insertar, la pila est� llena.");

} total:4

// --------------------------------------------------------------

public long elimina() { **----1**// elimina un elemento de la cima de la pila

long valor = Long.MIN\_VALUE; **-----1**

if (!estaVacia())**----2**

valor = arreglo[top--]; // ----1 retorna el elemento, decrementa el top

else **------- 1**

System.out.println("No se puede eliminar, la pila est� vac�a.");

return valor;**---1**

} **total=6**

// --------------------------------------------------------------

public long peek() { // obtiene el elemento de la cima de la pila

return arreglo[top];

}

// --------------------------------------------------------------

public boolean estaVacia() { // **----1** verdad si la pila est� vac�a

return (top == -1); **----1**

} **total=2**

// --------------------------------------------------------------

public boolean estaLlena() { // verdad si la pila est� llena---1

return (top == MAXIMO - 1);-------------------------------1

}

// --------------------------------------------------------------

} // fin de la clase Pila