

TEST PILAS Y COLAS (17/11/2010)
EDI - CURSO 2010/2011 (Grupo Sistemas)

Alumno/a: _____

D.N.I.: _____

TEST. 5 puntos.

Nota: cada pregunta correcta suma 1,25 puntos (las respuestas incorrectas no penalizan).

1.- Marque la respuesta/s correcta/s.

[X] En una pila los elementos se añaden y eliminan solo por un extremo: la cima.

[] En una cola los elementos se añaden y eliminan solo por un extremo: el final.

[X] En una cola los elementos se añaden solo por un extremo: el final.

[] En una cola los elementos se eliminan solo por un extremo: el final.

2.- ¿Qué operación diferencia la interfaz de pilas con gestión estática de memoria y gestión dinámica de memoria?

[] *Es_Vacía.*

[] Todas, ya que son implementaciones distintas.

[X] Ninguna, ya que la interfaz es común.

[] *Apilar.*

3.- ¿Qué estructura de datos (se permite marcar más de una respuesta) utilizaría para simular el comportamiento de clientes en un sistema de cajas de un supermercado? Justifique en una línea el porqué.

[] Cola con prioridad.

[] Cola sin prioridad.

[] Pila con ordenación.

[X] Cola incluyendo la operación borrar.

Justificación:

4.- Señale la principal desventaja de una interfaz respecto a una clase abstracta.

[] Las interfaces están ligadas a una jerarquía rígida de tipos de datos.

[] Las interfaces no están ligadas a una jerarquía rígida de tipos de datos.

[X] Es necesario hacer una inversión importante de esfuerzo si la interfaz cambia después de su especificación inicial.

[] No permiten el uso de herencia múltiple.

CUESTIONES. 5 puntos.

A (2,5 puntos) Realice la especificación algebraica de las operaciones *Último* y *Borrar* sobre el TAD Cola. La operación *Borrar* elimina todas las instancias del elemento a borrar en la cola.

* **Nota**: suponga $\forall c \in cola, \forall e, f \in elemento$

Último: Cola \rightarrow Elemento

Borrar: Cola \times Elemento \rightarrow Cola

Último (ColaVacía) = Error

Último (Añadir(c, e)) = e

Borrar (ColaVacía, e) = Error

Borrar (Añadir(c, e), f) =

Si $e == f$, entonces Borrar (c, f)

Si no, entonces Añadir (Borrar(c, f), e)

B (2,5 puntos) Suponiendo la siguiente definición de la clase *Cola*, la interfaz *Pila* y la clase *PilaDinamica*,

```
public abstract class Cola<E> implements Cloneable {
    public abstract boolean esVacia ();
    public abstract void anyadir (E e) throws ColaLlenaException;
    public abstract E primero () throws ColaVaciaException;
    public abstract void quitar () throws ColaVaciaException;
}

public interface Pila<E> {
    public boolean esVacia ();
    public void apilar (E v) throws DesbordamientoPilaException;
    public void desapilar () throws PilaVaciaException;
    public E cima () throws PilaVaciaException;
}

public class PilaDinamica<E> implements Pila<E> {
}
```

implemente en Java un método *public static <E> void invertir (Cola<E> cola)* que invierta el contenido de *cola* y el resultado lo almacene en la propia *cola*.

```
public static <E> void invertir (Cola<E> cola) {

    PilaDinamica<E> pila = new PilaDinamica<E>();

    try {
        while (!cola.esVacia()) {
            pila.apilar(cola.primer());
            cola.quitar();
        }

        while (!pila.esVacia()) {
            cola.anyadir(pila.cima());
            pila.desapilar();
        }
    } // Fin try.

    catch (ColaVaciaException exc) {}
    catch (ColaLlenaException exc) {}
    catch (DesbordamientoPilaException exc) {}
    catch (PilaVaciaException exc) {}

}
```