



Práctica 2: TAD Pila y TAD Cola

Objetivos:

- Diseñar, implementar y usar un TAD.

ENUNCIADO

Se desea diseñar los TADs Pila y Cola siguiendo las distintas etapas de la “técnica de abstracción de datos” que hemos estudiado.

Se pide:

PARTE 1: TAD PILA

1.1.- Realizar la **especificación lógica del TAD Pila** teniendo en cuenta las operaciones básicas que hemos considerado.

1.2.- **Implementar el TAD Pila en Java.**

La implementación se llevará a cabo en memoria dinámica (mediante una **lista enlazada**).

- a) Diseñar la **Interfaz del TAD Pila** (Interfaz Pila<E>, paquete “modelos”)
- b) Implementar en Java el TAD Pila mediante una lista enlazada (Clase LEPila, paquete “lineales”). Esta clase debe implementar la Interfaz anterior.

PARTE 2: TAD COLA

2.1.- Realizar la **especificación lógica del TAD Cola** teniendo en cuenta las operaciones básicas que hemos considerado.

2.2.- **Implementar el TAD Cola en Java.**

La implementación se llevará a cabo en memoria estática (mediante un **array circular**, tal y como hemos estudiado en clase).

- a) Diseñar la **Interfaz del TAD Cola** (Interfaz Cola<E>, paquete “modelos”)
- b) Implementar en Java el TAD Cola mediante mediante un **array circular** (Clase ArrayCola, paquete “lineales”). Esta clase debe implementar la Interfaz anterior.

PARTE 3: USO DE LOS TADS ANTERIORES

En este apartado se hará uso de los TADs Pila y Cola que acaba de diseñar. Para ello, debe realizar un programa que presente, de forma repetitiva, el siguiente sistema de menús de opciones:



MENÚ PRINCIPAL

- 1.- Submenú Pila de Enteros
- 2.- Submenú Cola de Enteros
- 3.- Comprobar frase palíndroma
- 0.- Salir

SUBMENÚ PILA DE ENTEROS

- 1.- Crear Pila de enteros
- 2.- Introducir elementos en la Pila
- 3.- Duplicar Pila
- 4.- Invertir Pila
- 5.- Comprobar Base de la Pila
- 0.- Volver

SUBMENÚ COLA DE ENTEROS

- 1.- Crear Cola de enteros
- 2.- Introducir elementos en la Cola
- 3.- Copiar Cola
- 4.- Repetir n-ésimo
- 0.- Volver

A continuación se detalla la tarea que se debe llevar a cabo al ejecutar cada una de las opciones de los menús anteriores:

MENÚ PRINCIPAL:

Opción 1.- “Submenú Pila de Enteros”: Muestra en pantalla el submenú Pila.

Opción 2.- “Submenú Cola de Enteros”: Muestra en pantalla el submenú Cola.

Opción 3.- “Comprobar frase palíndroma”: Solicita al usuario la introducción de una frase por teclado y muestra en pantalla un mensaje “La frase es palíndroma”, o bien, “La frase no es palíndroma”. Para llevar a cabo la comprobación anterior debe implementar un método estático booleano que, utilizando pilas y colas, compruebe si la frase que recibe como parámetro es palíndroma. => ***Método comprobarFrasePalindroma()***

Nota: Una frase es palíndroma si se lee igual de izquierda a derecha que de derecha a izquierda, una vez que se han eliminado los espacios en blanco. Por simplicidad, suponemos que la frase no contiene signos de puntuación. Ejemplos de frases palíndromas: “Yo hago yoga hoy” - “Amor a Roma”

Opción 0.- “Salir”: Finaliza la ejecución del programa mostrando un mensaje de despedida. “Gracias por utilizar nuestros TADs Pila y Cola”.



SUBMENÚ PILA DE ENTEROS:

Opción 1.- “Crear una pila de enteros”: Crea una pila de enteros vacía. A continuación, muestra en pantalla el mensaje “Pila vacía creada”. Si la pila ya existía, la deja vacía. =>

Método vaciarPila() y método Método crearPilaEnteros()

Pila vacía creada.
Pulse <Intro> para continuar...

Opción 2.- “Introducir elementos en la Pila”: Almacena en la pila los elementos enteros introducidos por el usuario. Sólo se admitirán elementos positivos. La introducción de datos finaliza con 0. Después, mostrará por pantalla los elementos almacenados en la Pila (uno en cada línea). => **Método apilarElemPos()**. Ejemplo:

Entrada de datos:

7 56 32 10 0

Salida en pantalla:

PILA
10
32
56
7

Opción 3.- “Duplicar Pila”: Obtiene un duplicado de la pila original con la que estamos trabajando y devuelve la nueva pila creada. La pila original NO queda modificada. => **Método recursivo duplicarPila()**

Salida en pantalla:

Pila original
PILA
10
32
56
7

Pila Copiada
NUEVA_PILA
10
32
56
7

Opción 4.- “Invertir Pila”: Para resolver esta opción del menú, **enriquezca el modelo Pila** añadiendo un nuevo método que modifique la Pila de tal forma que en ella queden almacenados los elementos en orden inverso al actual. Después, mostrará la Pila en pantalla. La pila original



quedará modificada. **Debe utilizar pilas y colas en la implementación del método.** => ***Método invertirPilaPC()*** . Salida en pantalla:

```
Pila Original
PILA
10
32
56
7

La pila invertida queda...
PILA
7
56
32
10
```

Opción 5.- “Comprobar Base Pila”: Al seleccionar esta opción se ejecuta un método estático recursivo que recibe como parámetro una pila de enteros y devuelve 0 si el elemento que se encuentra en la base de la pila es igual que el número de elementos de la misma, 1 si es mayor y 2 si es menor. La pila original NO debe quedar modificada. La base de la pila es el extremo opuesto a la cima de la misma. => ***Método recursivo comprobarBasePila()***

Nota: En base al valor retornado por el método anterior se mostrará en pantalla el mensaje correspondiente, por ejemplo: “*El elemento de la base de la pila es MAYOR que el número de elementos de la misma*” .

Salida en pantalla:

```
PILA
10
32
56
7

El elemento de la base de la pila es MAYOR
que el número de elementos de la misma
```

Opción 0.- “Volver”: Al seleccionar esta opción se vuelve a mostrar el menú principal de la aplicación.

SUBMENÚ COLA DE ENTEROS:

Opción 1.- “Crear una Cola de enteros”: Crea una cola de enteros vacía. A continuación, muestra en pantalla el mensaje “*Cola vacía creada*”. Si la cola ya existía, la deja vacía. => ***Método vaciarCola() y método crearColaEnteros()***

```
Cola vacía creada.
Pulse <Intro> para continuar...
```



Opción 2.- “Introducir elementos en la Cola”: Almacena en la cola los elementos enteros introducidos por el usuario. Sólo se admitirán elementos positivos. La introducción de datos finaliza con 0. Después, mostrará por pantalla los elementos almacenados en la Cola (uno en cada línea). => **Método encolarElemPos()**. Ejemplo:

Entrada de datos:

7 56 32 10 0

Salida en pantalla:

COLA

10
32
56
7

Opción 3.- “Copiar Cola”: Al ejecutar esta opción se realiza una copia de la cola original con la que estamos trabajando y devuelve la nueva cola creada. La cola original NO queda modificada. => **Método recursivo copiarCola()**. A continuación se muestra por pantalla el siguiente resultado:

Cola original

COLA

10
32
56
7

Cola Copiada

NUEVA_COLA

10
32
56
7

Opción 4.- “Repetir n-ésimo” : Solicita al usuario un número entero (n) e invoca al **método recursivo repetirNesimo()** que localiza el elemento que ocupa la n-ésima posición en la cola y le replica tantas veces como indique dicho elemento. La cola quedará modificada. No se permite utilizar estructuras de datos auxiliares. (Ver ejemplo a continuación).

COLA ORIGINAL

1
56
87
3
25

****Ejemplo: Replicar el 4º elemento. La cola queda:**

COLA REPLICADA

1
56
87
3
3
3
3
25

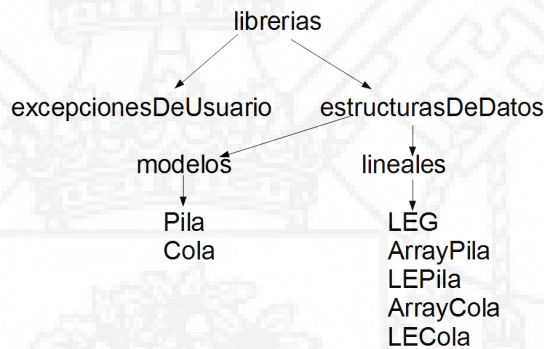


Opción 0.- “Volver”: Al seleccionar esta opción se vuelve a mostrar el menú principal de la aplicación.

Observaciones:

- Las opciones 2, 3, 4 y 5 del “submenú Pila” sólo podrán ser ejecutadas si con anterioridad se ha ejecutado la opción 1 de este submenú.
- Las opciones 2, 3 y 4 del “submenú Cola” sólo podrán ser ejecutadas si con anterioridad se ha ejecutado la opción 1 de este submenú.
- Puede diseñar las excepciones que considere oportunas.

Esquema de directorios que usaremos en la asignatura:



NORMAS DE REALIZACIÓN DE LA PRÁCTICA:

- Para cada práctica, además será preciso elaborar una memoria con el siguiente formato :
 - **Portada que incluya:**
 - * Título de la práctica
 - * Alumnos
 - * Nº de Grupo de prácticas
 - * Turno de Laboratorio
 - * Fecha de entrega
 - **Especificación lógica completa** de los TAD diseñados.
 - **Breve descripción de los procesos** llevados a cabo en los distintos métodos que implementan cada una de las opciones de los menús/submenús.
 - Listado de todo el **código fuente** desarrollado.
 - **Pruebas de ejecución** (pantallazos de ejecución de la aplicación).

FECHAS:

- Publicación de la práctica: Martes, 11 de abril de 2023
- Entrega de la práctica: Martes, 2 de Mayo de 2023 (**22 h - campus virtual**)



ENTREGA DE LA PRÁCTICA:

- La memoria de la práctica y el código fuente de la misma deberán ser entregados a través del campus virtual de acuerdo a las instrucciones que allí se detallan.
- No se admitirán entregas por correo electrónico ni por cualquier otro medio distinto al campus virtual.
- No se admitirá entregas fuera del plazo establecido.

AVISO:

- * Cualquier intento de plagio de la práctica supondrá la obtención de una nota 0.0 en la misma.
- * Además, los alumnos implicados deberán realizar de forma INDIVIDUAL una nueva práctica de complejidad muy superior a la actual para poder optar a superar la asignatura en la convocatoria extraordinaria.
- * La nota máxima que podrá obtener en esta nueva práctica será 6, siendo necesario obtener al menos un 5 para aprobar la parte práctica de la asignatura.
- * Se recuerda que para superar la asignatura es preciso aprobar ambas partes (Teoría y Práctica) por separado.