

TAD Pila

Práctica 8 Aplicación del TAD Pila

Objetivo

El objetivo de esta práctica es que el alumno continúe reforzando sus conocimientos acerca del TAD Pila mediante el desarrollo de un ejercicio que usará dicha estructura de datos además de aplicar sus conocimientos sobre herencia de clases.

Descripción general

La práctica consiste en implementar una especialización de la clase Pila que trabaje con objetos. Esta nueva pila, además de soportar las operaciones definidas en la interfaz `InterfazPila`, también debe soportar una función que devuelva el objeto mínimo contenido en la pila. El tiempo de ejecución de esta nueva función debe ser constante. Para determinar el objeto mínimo de la pila se requiere que sus elementos implementen la interfaz `Comparator` proporcionada por Java. **Ayuda:** Hacer uso de una pila auxiliar para almacenar los objetos mínimos.

Material

El material de esta práctica consta de los siguientes archivos:

- **InterfazPila.class** interfaz para el tipo abstracto de datos Pila.
- **Nodo.class** clase que implementa los nodos que sirven para crear una pila.
- **Pila.class** clase con implementación de la interfaz `InterfazPila` haciendo uso de nodos.
- **Pila\$Milterador.class** clase interna de la clase Pila que implementa un iterador para la pila.
- **PilaConMinimo.java** clase que implementará una especialización de la clase `Pila`.
- **PruebaPilaConMinimo.class** programa para probar la clase `PilaConMinimo`.
- **Contenedor.class** clase para trabajar con contenedores. Esta clase se utilizará para probar la clase `PilaConMinimo`. Se usará para crear una pila de contenedores.
- **ComparaContenedores.java** clase que implementará a la interfaz `Comparator` de Java para determinar la relación de orden entre objetos de la clase `Contenedor`.
- Documentación:
 - `InterfazPila.html` documentación de la interfaz `InterfazPila`.
 - `Nodo.html` documentación de la clase `Nodo`.

- Pila.html documentación de la clase `Pila`.
- Contenedor.html documentación de la clase `Contenedor`.

Desarrollo

1. Descargar los archivos `InterfazPila.class`, `Nodo.class`, `Pila.class`, `Pila$MiIterador.class`, `PilaConMinimo.java`, `PruebaPilaConMinimo.class`, `Contenedor.class` y `ComparaContenedores.java` en el directorio donde se va a desarrollar la práctica.
2. Leer la documentación y revisar el material de la práctica. Recordar que la firma de los métodos proporcionados y de los requeridos en el desarrollo de la práctica no podrán ser modificados.
3. Completar el código en la clase `PilaConMinimo`. Su **constructor** recibirá como parámetro un `Comparador`. Dicha clase extenderá a la clase `Pila` y deberá soportar todas las funciones de una pila. Incluir el método **minimo()** que devolverá el objeto mínimo contenido en la pila sin alterar a esta. Si la pila está vacía entonces devolverá `null`.
4. Completar el código en la clase `ComparaContenedores` que implementará la interfaz `Comparator` de Java. Definirá la relación de orden entre dos contenedores de acuerdo al peso de estos.
5. Probar la implementación de las clases ejecutando el programa `PruebaPilaConMin`. Si el programa funciona adecuadamente se verán los siguientes mensajes:

```
Creando 5 contenedores...
```

```
Push a la pila. Insertando 5 contenedores a la pila...
```

```
Iterando la pila:
```

```
Identificador del contenedor: 105 Peso: 20.3 Kg.
```

```
Identificador del contenedor: 104 Peso: 125.0 Kg.
```

```
Identificador del contenedor: 103 Peso: 15.7 Kg.
```

```
Identificador del contenedor: 102 Peso: 35.8 Kg.
```

```
Identificador del contenedor: 101 Peso: 68.5 Kg.
```

```
Minimo de la pila:
```

```
Identificador del contenedor: 103 Peso: 15.7 Kg.
```

Pop a la pila.

Pop a la pila.

Pop a la pila.

Iterando la pila:

Identificador del contenedor: 102 Peso: 35.8 Kg.

Identificador del contenedor: 101 Peso: 68.5 Kg.

Minimo de la pila:

Identificador del contenedor: 102 Peso: 35.8 Kg.

Vaciando la pila...

Minimo de la pila:

Null

6. Escribir una tabla con la complejidad de los métodos programados en la clase PilaConMinimo.

Operación	Tiempo de ejecución
vaciar	
pop	
push	
minimo	