

Programa

111

Mil

Guia 5.1: Collections - Pilas y Colas



Coordinadora: Adriana Brugnoli

Instructores: Leandro Maro, Matias Parra

13/06/18

E.E.T. N° 24 Simon de Iriondo

EJERCICIOS

1. Implementa una cola y una pila utilizando una lista enlazada como estructura de datos. La cola debe permitir almacenar cualquier tipo de objeto, e implementar al menos métodos para:
 - Insertar un objeto en ambas listas.
 - Recuperar un objeto poner un comentario la diferencia entre las 2 listas.
 - Obtener el primer objeto (first) de listas, sin extraerlo.
 - Averiguar si hay algún objeto almacenado (isEmpty).
 - Devolver el número de objetos almacenados (size).
 - Diseña la clase, decidiendo el nombre de cada método, sus parámetros y su valor de retorno.
2. Haz un metodo `balancedBrackets(String expr)` y se pide lo siguiente:
Devuelve true si la cadena `expr` que recibe como parámetro contiene una expresión en donde todos sus símbolos de agrupación (paréntesis `()`, corchetes `[]` y llaves `{}`) están correctamente anidados y balanceados. Devuelve false en caso contrario. Se debe ignorar cualquier carácter de `expr` que no sea paréntesis, corchetes o llaves.

Descripción del algoritmo: Empieza con una pila vacía. Recorre cada uno de los caracteres de `expr` de izquierda a derecha:

Si encuentras un símbolo de apertura (`(`, `[` o `{`) debes insertarlo en la pila.

Si encuentras un carácter de cierre (`)`, `]` o `}`) debes remover el carácter del tope de la pila y verificar que ambos caracteres hagan pareja. Debes terminar el algoritmo con false si los respectivos caracteres no hacen pareja, o si la pila estaba vacía antes de intentar remover el elemento del tope.

Si al final la pila está vacía debes devolver true, de otra forma debes devolver false.

3. Haz un método llamado merge(Queue<Integer> q1, Queue<Integer> q2) se pide:

Devuelve una nueva fila con el resultado de efectuar el algoritmo de mezcla a partir del contenido de las filas q1 y q2. Las filas q1 y q2 deben llegar ya ordenadas de forma ascendente.

Descripción del algoritmo: Empieza con una fila resultante vacía. En cada iteración determina quien tiene al inicio el elemento x más pequeño de entre q1 y q2 (recuerda que tanto q1 y q2 están ordenadas de forma ascendente, por lo que los elementos más pequeños siempre estarán justo al inicio de cada una de estas filas). Remueve a x de la fila que lo contiene y añádelo al final de la fila resultante. Las iteraciones terminan cuando alguna de q1 o q2 queda vacía. Posteriormente hay que copiar a la fila resultante todos los elementos que quedaron en la fila q1 o q2 que aún no está vacía. Finalmente debes regresar la fila resultante.

4. El Nuevo Banco del Chaco está solicitando que se realice un aplicación por consola donde se pueda gestionar los turnos de las personas que ingresan al banco para su atención, al momento del turno se tiene que emitir por pantalla el nombre de la persona, dni y el motivo porque está asistiendo al banco, cabe destacar que el motivo solo se tiene que elegir una opción.
5. Siguiendo el ejercicio anterior se desea modificar el código para que asigne el dicho cliente a un empleado que se encuentra en una lista particular, el empleado al ser asignado se lo tiene que sacar de esa lista y agregar a una lista de empleado ocupado al liberarse el empleado tiene que agregarse al final de la lista para esperar un nuevo cliente.
6. Escribir una clase TorreDeControl que modele el trabajo de una torre de control de un aeropuerto, con una pista de aterrizaje. La torre trabaja en dos etapas: reconocimiento y acción.

Escribir un metodo reconocimiento, que verifique si hay algún nuevo avión esperando para aterrizar y/o despegar, y de ser así los encole en la cola correspondiente. Para ello, utilizar random.randrange(2).

- a. Escribir un metodo acción, que haga aterrizar o bien despegar, al primero de los aviones que este esperando (los que esperan para aterrizar tienen prioridad). Debe desencolar el avión de su cola y devolver la información correspondiente.

- b. Escribir un metodo `__str__` que imprima el estado actual de ambas colas.
 - c. Escribir un programa que inicialice la torre de control, y luego llame continuamente a los metodos reconocimiento y acción, imprimiendo la acción tomada y el estado de la torre de control cada vez.
- 7. Atención a los pacientes de un consultorio medico, con varios doctores.
 - a. Escribir una clase ColaDePacientes, con los metodos nuevo_paciente, que reciba el nombre del paciente y lo encole, y un metodo proximo_paciente que devuelva el primer paciente en la cola y lo desencole.
 - b. Escribir una clase Recepcion, que contenga un diccionario con las colas correspondientes a cada doctor o doctora, y los métodos nuevo_paciente que reciba el nombre del paciente y del especialista, y proximo_paciente que reciba el nombre de la persona liberada y devuelva el próximo paciente en espera.
 - c. Escribir un programa que permita al usuario ingresar nuevos pacientes o indicar que un consultorio se ha liberado y en ese caso imprima el próximo paciente en espera.
- 8. Juego de Cartas
 - a. Crear una clase Carta que contenga un palo y un valor.
 - b. Crear una clase PilaDeCartas que vaya apilando las cartas una debajo de otra, pero sólo permita apilarlas si son de un número inmediatamente inferior y de distinto palo. Si se intenta apilar una carta incorrecta, debe lanzar una excepción.
 - c. Agregar el metodo `__str__` a la clase PilaDeCartas, para imprimir las cartas que se hayan apilado hasta el momento.
- 9. Pizzeria

La pizzeria **Guerrin** fundada hacia el año 1932 por el Sr. Arturo Malvezzi cuenta con un solo horno por tal motivo los pedidos que se toman ingresan a una cola de pedidos. Estos pedidos tienen una duración de entre 30 y 120 segundos. La persona que atiende la casa central al tomar un pedido desea saber el tiempo en el cual éste estará listo. Además se requiere la funcionalidad de retirar los pedidos,

la cual debe revisar el horario en que fue tomado y el tiempo que tarda ese pedido en cocinarse.

El escenario además deberá contar con todos los métodos get y set. Además se considera que la pizzería no tiene sucursales (no se pueden generar instancias de la misma)