



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Tipos de Datos Abstractos (TDA) - COLA.

Precondición para ver este documento es haber visto, trabajado y no queden dudas con los documentos anteriores, los correspondientes al TDA PILA

Entendiendo el problema:

El modelo computacional para representar una cola, por ejemplo, una cola de personas en la puerta de un comercio o en la ventanilla de un banco, de autos en una estación de servicio u otras contempla una variedad acotada de operaciones.

Antes de abrir el banco, se prepara el espacio en que se formará la fila de personas en espera de ser atendidas. Se entiende esta operación como preparar la cola (antes de poner nada en ella), inicializarla como cola vacía o simplemente . . .

- crear cola¹, esta será nuestra primera operación previa a cualquier otra operación sobre la cola. Una vez que esté cumplida se podrá comenzar a operar con la cola.

- poner en cola², es una operación que puede fallar (usted está haciendo la cola para entrar al banco, pero como ya están por cerrar y salió un empleado entregando un número a cada persona que está afuera, cuando usted se va a agregar a la cola se lo avisan. Se dio el caso de cola llena, que se podría determinar previamente con:

- cola llena³. Si no está llena, se puede poner en cola. Con esta operación, no hay mucha seguridad de que al poner en cola después de la determinación de cola llena (usted ve que están entrando al banco, pero mientras cruza la calle, cierran las puertas).

Otra operación que se puede realizar en cualquier momento (no sólo si hay elementos en la cola), es la de:

- ver primero⁴ esta operación permite ver el primero de la cola sin eliminarlo. Esta operación puede fallar si la cola está vacía, o ser exitosa. Se asemeja a que el cajero vea que no hay gente esperando para dejar momentáneamente su trabajo e ir a tomar un café. Apenas se da vuelta, llega una persona. También pudo ver que había una



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

persona, le dice que puede acercarse, pero entre una y otra cosa lo llamó el cajero de al lado. A esta operación se la podría hacer depender de que no se de *cola vacía*, o simplemente si ver primero es o no exitosa.

. . . Y como todo lo que se *pone en cola* en algún momento hay que sacarlo, ¡que a nadie se le ocurra sacar del final o del medio!, así será necesaria la operación de:

- sacar de cola⁵, esta operación podrá ser exitosa o no. Cuando no es exitosa es porque se dio la condición de . . .

- cola vacía⁶, anteriormente mencionada, si se da la condición de que está vacía, no se podrá sacar de cola y uno deja de sacar y se retira, lo que no siempre es cierto porque entre que usted determinó que la cola está vacía y quiere dar por concluida su tarea, llega una nueva persona a la cola.

Hasta acá estuvimos entendiendo el problema y hasta hemos considerado el caso de que haya más de uno operando con la cola.

Además de estas seis operaciones hay otra (su mayor importancia está en el modelo computacional):

- vaciar cola⁷, deja la cola vacía y lista para ser utilizada nuevamente. O tan solo para liberar la cola (sale un empleado avisando que ¡¡¡se cayó el sistema!!!).

OBSERVACIONES: Cuando se trabaja con colas, no se puede hacer sobre las mismas ninguna otra operación que no sea una de las siete que hemos dado. Una cola no se puede recorrer para inspeccionar, contar cuántos hay u otras acciones. No se puede sacar cualquiera de la cola, sólo el primero. Y también sólo se puede ver el primero.

NOTA: la bibliografía dice que poner en cola tiene como precondition la determinación previa de cola llena, y que ver primero, y sacar de cola tienen como precondition la determinación previa de cola vacía, pero en un modelo computacional con procesos con memoria compartida esto deja de ser cierto. Pasan a ser operaciones convenientes



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

acorde a la situación que se esté modelando. Estas cinco operaciones son booleanas, devuelven un indicador de falso o verdad (en el caso de cola llena y cola vacía), y de éxito o fracaso (en los casos de las operaciones de poner en cola, ver primero y sacar de cola). En tanto que las operaciones de crear cola y vaciar cola no devuelven ningún indicador.

Estrategia de solución:

Para resolver este TDA tenemos dos posibilidades: con asignación estática o dinámica de memoria.

TDA Cola con asignación dinámica de memoria, y**TDA Cola con asignación estática de memoria.**

Para el primero, se genera memoria en tiempo de ejecución para almacenar cada elemento que queremos poner en cola. Para el segundo, se declara el espacio máximo dedicado a la cola.

Una última observación: trataremos con tipos de datos abstractos, por lo que los elementos a tratar tendrán dos atributos: el elemento en sí mismo (haciendo abstracción de su tipo de dato), además de su tamaño (la cantidad de bytes que ocupa). Así, se podrá tratar con objetos diversos (personas, productos, cadenas de caracteres, etc.), del mismo modo que se hizo con el TDA Pila.