INSTITUTO TECNOLÓGICO DE COSTA RICA
Escuela de Computación - Bachillerato en Ingeniería en Computación
IC-2001 Estructuras de Datos - Prof. Mauricio Avilés

Tarea Corta 4 – Pilas y Colas

Esta tarea consiste en la programación, modificación y utilización de las siguientes estructuras de datos:

1. Pilas (ArrayStack y LinkedStack)
2. Colas (ArrayQueue y LinkedQueue)

Parte 1

Implementar las clases **ArrayStack** y **LinkedStack**. Estas clases deben derivar de la clase abstracta **Stack** vista en clase. A continuación, se detallan los métodos que deben implementarse en cada una de ellas. Se subrayan los métodos que son diferentes a lo implementado en clase.

1. **Constructor y destructor**
2. **void push(E element)**
   En el caso del ArrayStack, si la pila ya alcanzó su tamaño máximo, debe crearse un nuevo arreglo del doble de tamaño y traspasarse todos los elementos al nuevo arreglo.
3. **E pop()**
4. **E topValue()**
5. **void clear()**
6. **bool isEmpty()**
7. **int getSize()**
8. **void print()**
   Este método imprime en una sola línea los contenidos de la pila, empezando por el elemento que se encuentra en el tope de la pila.

Escriba un programa que lea un *string* y determine si contiene una secuencia de caracteres de agrupamiento **()**, **[]**, **{}**, válida. Cualquier otro carácter que no sea de agrupamiento debe ignorarse al procesar el *string*. Se lee la secuencia de caracteres y se procesan uno por uno. Si se encuentra un carácter de apertura **(**, **[**, **{**, entonces se agrega (**push**) a la pila. Si se encuentra un carácter de cierre **)**, **]**, **}**, entonces se verifica que el elemento en el tope de la pila sea el carácter de apertura correspondiente. Si coincide, se elimina de la pila (**pop**) y se continua con el siguiente carácter en la secuencia. Si en algún momento no corresponden, la secuencia no es válida. Si la pila está vacía y se encuentra con un carácter de cierre, tampoco es válida. Si se llega al final de la secuencia y quedan elementos en la pila, tampoco es válida. La secuencia es válida si se llega al final y la pila está vacía. Imprima los contenidos de la pila cada vez que procesa un carácter. El usuario debe escoger cuál tipo de pila utilizar, si ArrayStack o LinkedStack.
Algunos ejemplos:

- {([hola])} → válido
- {(} → no válido
- {([(1)(2)][{3}])} → válido

Parte 2

Implementar las clases **ArrayQueue** y **LinkedQueue**. Estas clases deben derivar de la clase abstracta **Queue** vista en clase. Se agregan métodos para que las clases puedan ser utilizadas como una estructura **Deque** (double-ended-queue). Este tipo de estructura es una cola que permite operaciones de inserción y de borrado en ambos extremos de la cola. A continuación, se detallan los métodos que deben implementarse.

1. **Constructor y destructor**
2. **void enqueue(E element)**
   Inserta un nuevo elemento en el final de la cola (**rear**).
3. **E dequeue()**
   Elimina y retorna el elemento ubicado en el frente de la cola (**front**).
4. **E frontValue()**
5. **void clear()**
6. **bool isEmpty()**
7. **int getSize()**
8. **void enqueueFront(E element)**
   Inserta un nuevo elemento en el frente de la cola (**front**).
9. **E dequeueRear()**
   Elimina un elemento del final de la cola (**rear**).
10. **E rearValue()**
    Retorna el elemento situado en el final de la cola (**rear**).
11. **void print()**
    Imprime en una sola línea los contenidos de la cola, empezando por el frente.

Escriba un programa que pregunte al usuario un número entero positivo. Este número indicará la cantidad de operaciones que se realizarán sobre la estructura. Las operaciones por realizar se escogerán de forma aleatoria, es decir, **enqueue**, **dequeue**, **enqueueFront**, **dequeueRear**. Las operaciones de **dequeue** y **dequeueRear** sólo se pueden escoger si la cola tiene elementos. En las operaciones de agregar elementos, debe generarse un número aleatorio para insertar. Al efectuar cada una de las operaciones, debe imprimirse cuál operación se realiza, el tamaño de la cola, el elemento que se inserta o se elimina y los contenidos de la cola. En el caso de la cola implementada con arreglos, si esta llega a llenarse, debe mostrarse un mensaje indicando que esto sucedió. Al finalizar las operaciones debe limpiarse la cola (**clear**) y preguntar de nuevo al usuario si desea repetir la prueba con una cantidad diferente de operaciones. El usuario también puede escoger cuál tipo de cola utilizar, si ArrayQueue o LinkedQueue.

Entrega: Sección de evaluaciones del TEC-Digital.
Formato: Archivo comprimido ZIP con los proyectos de Code::Blocks, uno por cada parte de la tarea. Corrobore que se incluyen todos los archivos necesarios para que los proyectos se ejecuten correctamente.