Comenzado el Tuesday, 17 de January de 2023, 12:31
Estado Finalizado
Finalizado en Tuesday, 17 de January de 2023, 12:54
Tiempo empleado 22 minutos 54 segundos
Calificación **100,00** de 100,00
Comentario - Respuesta correcta del examen

```
public clsLista AddS(int i, int x)
{
    if (!LLena(i))                          // Se adicionan elementos si no esta llena
        if (Vacia(i))                       // Si esta vacia la lista i se adiciona el primer elemento
        {
            listaS[i].AddPrimero(x);
            C[i] = C[i] + 1;                // incrementa en uno el contador de elemento de la lista i
        }
        else                                // Sino esta vacia se adiciona el elemento al final
        {
            listaS[i].AddUltimo(x);
            C[i] = C[i] + 1;                // incrementa en uno el contador de elemento de la lista i
        }
    return listaS[i];                       // retorna la lista i
}
//Funciona retorna TRUE si la lista i esta llena
public bool LLena(int i)
{
    return (C[i]==B[i]);
}

//Funcion retorna TRUE si la lista i esta vacia
public bool Vacia(int i)
{
    return (C[i]==0);
}
// funcion que retorna la cantidad de espacio disponible en la lista i
public int DispoS(int i)
{
    return (B[i]-C[i]);
}
// funcion que retorna la cantidad de elementos existentes en las N listas
public int Long()
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        s = s + (B[i]-DispoS(i));
    }
    return s;
}
```

Pregunta **1**

Correcta

Se puntúa 100,00 sobre 100,00

Se requiere gestionar **n** en listas enlazadas simples, considerar que cada lista tiene un tamaño máximo de elementos. La especificación formal del TAD Listas es la siguiente:

```
NOMBRE: TAD ListaS
CONJUNTO:  N número natural (0..n), Li es la identificación de la i-ésima lista enlazada
(0<=i<n), n cantidad de listas enlazadas a gestionar
FUNCIONES
 1. Create ListaS()                    // crea las n listas enlazadas

 1. AddS(Li,x)   --> Li                // Adiciona el elemento x a la lista Li

 1. LlenaS(Li) -->Boolean              // Retorna true si Li está llena

 1. DispoS(Li)-- > N                       // Retorna la cantidad de espacio disponible de la
    lista Li

 1. Long()  --> N                           // Retorna la cantidad de elementos de las n listas

AXIOMAS
AddS(Li,x)::= if LlenaS(Li) then “ERROR”
```

Rellene el espacio en blanco con la lógica correcta para que funcionen correctamente las funciones de la clase clsListaS.

NOTA: Todas las instrucciones debe ser rellenada sin espacios en blanco, Si alguna función no funciona después de rellenar las instrucciones, se descuenta el 40% de la nota final

```
// Clase del TAD ListaS
 public class clsListaS
    {
       //ATRIBUTOS DE LA CLASE
        private static int N = 5;             // Numero de Listas enlazadas simple a gestionar
        public clsLista[] listaS;            // Arreglo donde se almacena las N listas enlazadas
        public int[] B;                      // Arreglo donde se define el tamaño de cada lista i,  para 0<=i<N
        public int[] C;                      // Arreglo para el contador de elementos de cada lista i, para 0<=i<N
        //CONSTRUCTOR DE LA CLASE
        public clsListaS()
        {
            listaS = new clsLista[N];
            B = new int[N];
            C = new int[N];
            for (int i = 0; i<N; i++)
            {
                listaS[i] = new clsLista();         // Se inicializa cada Lista i
                B[i] = ((i+1)*2);                   // Se define el tamaño de cada Lista i
                C[i] = 0;                           // El contador de elementos de cada Lista i se coloca en cero
            }
        }
        // MODIFICA EL CONSTRUCTOR DE LA CLASE -- con el fin de inicializar los ATRIBUTOS

        //PROPIEDADES DE DESCRIPCION DE ACCESOS -- (get/set) que permite leer o cambiar los campos desde otras clases
        public clsLista[] Listas
        {
            get { return listaS; }
            set { listaS = value; }
        }
        //FUNCIONES DE LA CLASE
        //Funcion para adicionar un elemento x a la Lista i
        public clsLista AddS(int i, int x)
        {
            if (!LLena(i))                          // Se adicionan elementos si no esta llena
                if (Vacia(i))                       // Si esta vacia la lista i se adiciona el primer elemento
                {
                    listaS[i].AddPrimero(x);
                    C[i] = C[i] + 1;                // incrementa en uno el contador de elemento de la lista i
                }
                else                                // Sino esta vacia se adiciona el elemento al final
                {
                    listaS[i].AddUltimo(x);
                    C[i] = C[i] + 1;                // incrementa en uno el contador de elemento de la lista i
                }
            return
```

listaS[i];

✔

```
        // retorna la lista i
        }
        //Funciona retorna TRUE si la lista i esta llena
        public bool LLena(int i)
        {
            return
```

(C[i]==B[i]);

✔

```
        }

        //Funcion retorna TRUE si la lista i esta vacia
        public bool Vacia(int i)
        {
            return
```

(C[i]==0);

✔

```
        }
        // funcion que retorna la cantidad de espacio disponible en la lista i
        public int DispoS(int i)
        {
            return
```

(B[i]-C[i]);

```
        }
        // funcion que retorna la cantidad de elementos existentes en las N listas
        public int Long()
        {
            int s = 0;
            for (int i = 0; i < n; i++)
            {
                s = s +
```

(B[i]-DispoS(i));

```
            }
            return s;
        }
    }
}
```

◀ Cuestionario 1 - T7

Ir a...

Resumen de retención de datos
Descargar la app para dispositivos móviles