30/08/2019

Tema: Estructuras de datos lineales:

TDA:

* Tipos de datos abstractos
* Fundamentos matemáticos
* Todas las listas de java utiliza la interface de List
* Listas enlazadas:

Add

addFirst

addList

remove

remove first

* Árboles
* Grafos
* Set/Conjuntos
* Maps/ Diccionary
* Colas
* Pilas

Arrays:

* Colección finita y bien definida de elementos de x tipo.
* Cada elemento esta contiguo una al otro en memoria.

Int[] A = new int[8];

A[0] = 10;

A[7] = 20;

Pros del Array:

* Rápida lectura/ escritura

Contras:

* Eliminación/ Inserción en posiciones aleatorias si no hay espacio.
* No puede crecer o decrecer por demanda

Matrices:

* Array multidimensional.
* Filas y Columnas
* No todos los elementos están contiguos

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Listas Enlazadas:

* Estructura de datos dinámica.
* Crece o decrece por demanda.
* No está contigua en memoria.
* Lista secuencial de elementos.
* Cada elemento es un nodo.
* Un nodo es un elemento compuesto por dos elementos:
  + El valor que guarda el dato.
  + Referencia al siguiente elemento de la lista.
* La lista tiene una referencia al primer elemento.
* La referencia al primero debe protegerse.
* El último nodo apunta a null.

Pros:

* Crece y decrece.
* Inserción en posiciones aleatoria es rápida.

Contras:

* Lectura/ búsqueda es lenta.

Package list;

Package nodo;

Pubic list;

Class nodo{

Int dato;

Nodo next;

Public nodo(int dato){

This.dato = dato;

This.next = null;

}

}

Public class List{

Private nodo first;

Public List(){

This.first = null;

}

Public void addList(int e){

If(this.first == null){

This.first = new nodo(e);

} else {

Nodo current = this.first;

While(current.next != null){

Current = current.next;

}

Current.next = new Nodo(e);

}

}

}

Buscar términos de generics.