**PILA**

**package** Estructutrados;

**public** **class** Pila {

//con el criterio LIFO

//Última en Entrar, Primera en Salir

**public** Nodo raiz;

//constructor

**public** Pila() {

raiz = **null**;

}

//insertar(int x): inserta en la pila

**public** **void** insertar(**int** x) {

Nodo nuevo = **new** Nodo();

nuevo.info = x;

**if**(raiz == **null**) {

//nuevo.info = 4;

raiz = nuevo;

}

**else** {

//nuevo.info=7;

nuevo.sig = raiz;

raiz = nuevo;

}

}

//sacar (): saca el primer elemento de la pila

**public** **void** sacar(**int** x) {

Nodo aux = raiz;

Nodo anterior = raiz;

**while** (aux!=**null**) {

**if** (aux.info == x) {

anterior.sig= aux.sig;

}

anterior = aux;

aux = aux.sig;

raiz = raiz.sig;

}

}

//mostar lista

**public** **void** mostrarLista(**int** x) {

Nodo uno = raiz;

**while** (uno!=**null**) {

System.***out***.print(uno.info + "->");

uno = uno.sig;

}

System.***out***.println("");

}

//cima(): muestra el primer elemento de la pila

**public** **void** cima() {

}

//size(): muestra el tamaño de la pila

**public** **void** size(**int** x) {

}

//MAIN

**public** **static** **void** main(String[] args) {

Pila pila = **new** Pila();

pila.insertar(7);

pila.insertar(5);

pila.mostrarLista(0);

pila.sacar(7);

}

}

**COLA**

**package** Estructutrados;

**public** **class** Cola {

//con el criterio FIFO

//por el cual la última unidad de carga en entrar al almacén será la primera en salir del mismo

**public** Nodo raiz;

//constructor

**public** Cola() {

raiz = **null**;

}

//insertar(int x): inserta el elemento x al final de la cola

**public** **void** insertar(**int** x) {

Nodo nuevo = **new** Nodo();

nuevo.info = x;

**if**(raiz == **null**) {

//nuevo.info = 4;

raiz = nuevo;

}

**else** {

//nuevo.info=7;

nuevo.sig = raiz;

raiz = nuevo;

}

}

**public** **void** insertaPos(**int** x) {

Nodo nuevo = **new** Nodo();

nuevo.info = x;

Nodo busca = raiz;

**while** (busca.sig != **null**) {

busca = busca.sig;

}

busca.sig = nuevo;

}

// sacar (): saca el primer elemento de la cola

**public** **void** sacar(**int** x) {

Nodo aux = raiz;

Nodo anterior = raiz;

**while**(aux != **null**) {

**if** (aux.info == x) {

anterior.sig = aux.sig;

}

anterior = aux;

aux = aux.sig;

raiz = raiz.sig;

}

}

//mostar lista

**public** **void** mostrarLista(**int** x) {

Nodo uno = raiz;

**while** (uno!=**null**) {

System.***out***.print(uno.info + "->");

uno = uno.sig;

}

System.***out***.println("");

}

//primero(): muestra el primer elemento de la cola

**public** **void** primero(**int** x) {

Nodo num = **new** Nodo();

num.info = x;

Nodo busca = raiz;

**while**(busca.sig != **null**) {

busca = busca.sig;

}

busca.sig = num;

}

//ultimo(): muestra el ultimo elemento de la cola

**public** **void** ultimo(**int** x) {

}

//size(): muestra el tamaño de la cola

**public** **void** size() {

}

//MAIN

**public** **static** **void** main(String[] args) {

Cola cola = **new** Cola();

cola.insertar(5);

cola.insertar(7);

cola.insertaPos(8);

cola.sacar(5);

cola.primero(0);

cola.mostrarLista(0);

}

}

**LISTA\_ORDENADA**

**package** Estructutrados;

**public** **class** Lista\_ordenada {

//manteniendo todos los elementos ordenados de menor a mayor

**public** Nodo raiz;

//constructor

**public** Lista\_ordenada() {

raiz = **null**;

}

//boolean esta (int x) : devuelve true si existe un nodo en la lista con ese valor

**public** **boolean** numExiste(**int** x) {

**return** x;

}

//int estaPosicion (int x) : devuelve la posición del elemento x en la lista, -1 en caso contrario.

**public** **int** estaPosicion(**int** x) {

**return** x;

}

//insertar(int x): inserta el elemento en la posición que garantice que la lista se mantiene ordenada

**public** **int** insertar (**int** x) {

**return** x;

}

//borrar(int x): borrar el nodo que tenga el valor x

**public** **int** borra(**int** x) {

**return** x;

}

//primero(): muestra el valor del primer elemento de la lista (el nodo raíz)

**public** **void** primero() {

}

//ultimo(): muestra el valor del último elemento

**public** **void** ultimo() {

}

//size(): muestra el tamaño de la lista

**public** **void** size() {

}

//MAIN

**public** **static** **void** main(String[] args) {

Lista\_ordenada ordenada = **new** Lista\_ordenada();

}

}

**NODO**

**package** Estructutrados;

**public** **class** Nodo {

**public** **int** info;

**public** Nodo sig;

}