Codigo de lista con cursor en C#

public class Nodo

{

int dato;

int siguiente;

public Nodo()

{

siguiente = -2; //equivale a null

}

public void setDato(int x)

{

dato = x;

}

public int getDato()

{

return dato;

}

public void setSiguiente(int xp)

{

siguiente = xp;

}

public int getSiguiente()

{

return siguiente;

}

}

public class Lista

{

int max;

int cab;

int cantidad;

Nodo [] espacio;

int disponible;

public Lista (int xmax)

{

max = xmax;

cab = 0;

cantidad = 0;

espacio = new Nodo[max];

for (int i = 0; i < max; i++) {

espacio[i] = new Nodo();

}

disponible = 0;

}

public bool vacia()

{

return cantidad == 0;

}

public bool getDisponible(out int disp)

{

int i = 0;

while ((i < max) && (espacio[i].getSiguiente() != -2)) {

i++;

}

if (i < max) {

disp = i;

return true;

} else {

disp = -2;

return false;

}

}

public bool freeDisponible(int disp)

{

if ((disp >= 0) && (disp < max)) {

espacio[disp].setSiguiente(-2);

return true;

} else {

return false;

}

}

public bool insertar(int x, int xp) //inserta por posición

{

if ((cantidad < max) && (xp >= 0) && (xp <= cantidad) && (getDisponible(out disponible))) {

espacio[disponible].setDato(x);

int ant = cab, cabeza = cab, i = 0;

while (i < xp) {

i++;

ant = cabeza;

cabeza = espacio[cabeza].getSiguiente();

}

if (cabeza == cab) { //inserta al inicio de la lista

if (cantidad == 0) { //inserta en la lista vacía

espacio[cab].setSiguiente(-1);

} else { //inserta en la lista con elementos

espacio[disponible].setSiguiente(cab);

}

cab = disponible;

} else if (cabeza == -1) { //inserta al final de la lista

espacio[disponible].setSiguiente(-1);

espacio[ant].setSiguiente(disponible);

} else {

espacio[disponible].setSiguiente(cabeza);

espacio[ant].setSiguiente(disponible);

}

cantidad++;

return true;

} else {

Console.WriteLine("Espacio lleno o posición incorrecta.");

return false;

}

}

public bool insertar (int x) //inserta por contenido

{

if ((cantidad < max) && (getDisponible(out disponible))) {

int ant = cab, cabeza = cab, i = 0;

espacio[disponible].setDato(x);

while ((i < cantidad) && (cabeza != -1) && (espacio[cabeza].getDato() < x)) {

i++;

ant = cabeza;

cabeza = espacio[cabeza].getSiguiente();

}

if (cabeza == cab) { //inserta al inicio de la lista

if (cantidad == 0) { //inserta en la lista vacía

espacio[cab].setSiguiente(-1);

} else { //inserta en la lista con elementos

espacio[disponible].setSiguiente(cab);

}

cab = disponible;

} else if (cabeza == -1) { //inserta al final de la lista

espacio[disponible].setSiguiente(-1);

espacio[ant].setSiguiente(disponible);

} else {

espacio[disponible].setSiguiente(cabeza);

espacio[ant].setSiguiente(disponible);

}

cantidad++;

return true;

} else {

Console.WriteLine("Espacio lleno.");

return false;

};

}

public bool suprimir(out int x, int xp)

{

if ((cantidad != 0) && (xp >= 0) && (xp < cantidad)) {

int ant = cab, cabeza = cab, i = 0;

while ((i < xp) && (cabeza != -1)) {

i++;

ant = cabeza;

cabeza = espacio[cabeza].getSiguiente();

}

if (cabeza == cab) {

if (cantidad == 1) {

cab = 0;

} else {

cab = espacio[ant].getSiguiente();

}

} else {

espacio[ant].setSiguiente(espacio[cabeza].getSiguiente());

}

x = espacio[cabeza].getDato();

disponible = cabeza;

freeDisponible(disponible);

cantidad--;

return true;

} else {

Console.WriteLine("Lista vacía o posición incorrecta.");

x = 0;

return false;

}

}

public bool recuperar(out int x, int xp)

{

if ((cantidad != 0) && (xp >= 0) && (xp < cantidad)) {

int cabeza = cab, i = 0;

while ((cabeza != -1) && (i < xp)) {

i++;

cabeza = espacio[cabeza].getSiguiente();

}

x = espacio[cabeza].getDato();

return true;

} else {

Console.WriteLine("Lista vacía o posición incorrecta.");

x = 0;

return false;

}

}

public bool buscar(int x, out int xp)

{

if (cantidad != 0) {

int cabeza = cab, i = 0;

while ((i < cantidad) && (cabeza != -1) && (espacio[cabeza].getDato() != x)) {

i++;

cabeza = espacio[cabeza].getSiguiente();

}

if (i < cantidad) {

xp = i + 1;

return true;

} else {

xp = 0;

return false;

}

} else {

Console.WriteLine("Lista vacía.");

xp = 0;

return false;

}

}

public bool primer\_elemento(out int x)

{

if (cantidad != 0) {

x = espacio[cab].getDato();

return true;

} else {

Console.WriteLine("Lista vacía.");

x = 0;

return false;

}

}

public bool ultimo\_elemento(out int x)

{

if (cantidad != 0) {

int cabeza = cab, aux = 0;

while (cabeza != -1) {

aux = espacio[cabeza].getDato();

cabeza = espacio[cabeza].getSiguiente();

}

x = aux;

return true;

} else {

Console.WriteLine("Lista vacía.");

x = 0;

return false;

}

}

public bool siguiente\_posicion(int xp, out int xp1)

{

if ((cantidad != 0) && (xp >= 0) && (xp < cantidad - 1)) {

xp1 = xp + 2;

return true;

} else {

Console.WriteLine("Lista vacía o último elemento.");

xp1 = 0;

return false;

}

}

public bool anterior\_posicion(int xp, out int xp1)

{

if ((cantidad != 0) && (xp > 0) && (xp < cantidad)) {

xp1 = xp;

return true;

} else {

Console.WriteLine("Lista vacía o primer elemento.");

xp1 = 0;

return false;

}

}

public bool recorrer ()

{

if (cantidad != 0) {

int cabeza = cab;

Console.Write ("Lista:");

while (cabeza != -1) {

Console.Write (" " + espacio[cabeza].getDato ());

cabeza = espacio [cabeza].getSiguiente ();

}

Console.WriteLine ();

return true;

} else {

Console.WriteLine ("Lista vacía.");

return false;

}

}

public bool sucesor(int x, out int x1)

{

int xx, p1, p2;

if ((cantidad != 0) && (buscar(x, out p1)) && (siguiente\_posicion(p1, out p2)) && (recuperar(out xx, p2))) {

x1 = xx;

return true;

} else {

x1 = 0;

return false;

}

}

public bool predecesor(int x, out int x1)

{

int xx, p1, p2;

if ((cantidad != 0) && (buscar(x, out p1)) && (anterior\_posicion(p1, out p2)) && (recuperar(out xx, p2))) {

x1 = xx;

return true;

} else {

x1 = 0;

return false;

}

}