**Харківський національний університет радіоелектроніки**

**Лабораторна робота №2**

**З дисципліни**

**«Алгоритми та структури даних»**

Виконав студент групи ПЗПІ-20-4

Прудіус Владислав

Перевірив асистент кафедри ПІ

Терещенко Г.Ю.

**2021**

**Тема:** Вивчення методів створення модифікації та використання структури даних sorted linked list (сортований зв’язний список).

**Завдання:**

1. Написати програму в якій реалізувати структуру даних sorted linked list

2. Додати до списку такі методи: ListSize, IsEmpty, AddItem, Print, Search, Retrieve, DdeleteItem, MakeEmpty

**Хід роботи**

Код програми на C#:

Файл Program.cs

class Program

{

static void Main(string[] args)

{

var a = new List<int>();

a.AddItem(14);

a.AddItem(2);

a.AddItem(5);

a.AddItem(45);

a.AddItem(-5);

a.AddItem(-4);

a.AddItem(-45);

a.AddItem(4);

a.Print();

Console.Write("List empty: ");

Console.WriteLine(a.IsEmpty());

Console.Write("List size: ");

Console.WriteLine(a.ListSize());

Console.Write("Retrieve elemet -5 (if 0 then no such element): ");

Console.WriteLine(a.Retrieve(-5));

Console.Write("Retrieve elemet 15 (if 0 then no such element): ");

Console.WriteLine(a.Retrieve(15));

Console.Write("Search elemet -5: ");

Console.WriteLine(a.Search(-5));

Console.Write("Search elemet 15: ");

Console.WriteLine(a.Search(15));

Console.WriteLine("Deleting elements -4 and 2");

a.DeleteItem(-4);

a.DeleteItem(2);

a.Print();

}

}

Файл Node.cs

class Node<T>

{

public T data;

public Node<T> next;

public Node(T d, Node<T> link = null)

{

data = d;

next = link;

}

}

Файл List.cs

class List<T> where T: IComparable<T>

{

public Node<T> first;

private int \_count;

public List()

{

first = null;

\_count = 0;

} // constructor

public int ListSize() { return \_count; }

public bool IsEmpty()

{

return (first == null) ? true : false;

} // IsEmpty

public void AddItem(T item) // keeps list sorted

{

if (IsEmpty()) // insert very first item

{

first = new Node<T>(item);

\_count++;

}

else

{

Node<T> current = first;

Node<T> previous = null;

while (current != null && item.CompareTo(current.data) > 0)

{

previous = current;

current = current.next;

}

if (current == first && item.CompareTo(current.data) <= 0) // insert to the first position

first = new Node<T>(item, current);

else // insert to the last and middle position

previous.next = new Node<T>(item, current);

\_count++;

}

} // AddItem

public void Print()

{

Node<T> n = first;

Console.WriteLine("List content: ");

if (IsEmpty())

Console.WriteLine("List is empty.");

else

{

while (n != null)

{

Console.Write($"| {n.data} ");

n = n.next;

}

Console.WriteLine("|");

}

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

} // Print

public bool Search(T item)

{

Node<T> n = first;

while (n != null)

{

if (n.data.Equals(item))

return true;

n = n.next;

}

return false;

} // Search

public T Retrieve(T item) // finds and returns item

{

Node<T> n = first;

while (n != null)

{

if (n.data.Equals(item))

return n.data;

n = n.next;

}

return default(T);

} // Retrieve

public T DeleteItem(T item)

{

if (!Search(item))

{

Console.WriteLine("There is no such element in the list");

return default(T);

}

else if (IsEmpty())

{

Console.WriteLine("There is an empty list");

return default(T);

}

Node<T> current = first;

Node<T> previous = null;

while (!item.Equals(current.data) && current.next != null) // find right element

{

previous = current;

current = current.next;

}

if (current == first && item.Equals(current.data)) // delete from first position

{

first = current.next;

\_count--;

return current.data;

}

else // delete from other positions

{

previous.next = current.next;

\_count--;

return current.data;

}

} // DeleteItem

public void MakeEmpty()

{

first = null;

\_count = 0;

} // MakeEmpty

} // List

Висновок: на даній лабортаторній роботі я навчився проектувати, розробляти та модифікувати прості структури даних зокрема Sorted Linked List мовою програмування С#.