***Харківський національний університет радіоелектроніки***

**Лабораторна робота №1**

**З дисципліни «Алгоритми та структури даних»**

*Виконав студент групи ПЗПІ-20-2 Столбецький Ю.О.*

*Перевірив ассистент кафедри ПІ Терещенко Г.Ю.*

**Тема: “**Sorted linked list**”**

**Мета: навчитися створювати структуру даних Sorted linked list.**

**Код програми:**

using System;

namespace Lab

{

class Program

{

static void Main()

{

var linkedList = new Lab1.LinkedList<int>();

Console.Write("Элементы списка: ");

linkedList.PrintLinkedList();

Console.WriteLine("Пустой ли список ? " + linkedList.IsEmpty());

Console.WriteLine("Кол-во элементов в списке: " + linkedList.Count);

Console.WriteLine("Есть ли число 11 в списке ? " + linkedList.Search(11));

Console.WriteLine("Добавление элементов");

Console.WriteLine();

linkedList.Read(linkedList);

Console.Write("Элементы списка: ");

linkedList.PrintLinkedList();

Console.WriteLine("Пустой ли список ? " + linkedList.IsEmpty());

Console.WriteLine("Кол-во элементов в списке: " + linkedList.Count);

Console.WriteLine("Есть ли число 11 в списке ? " + linkedList.Search(11));

Console.WriteLine("Есть ли число 12 в списке ? " + linkedList.Search(12));

Console.WriteLine("Удаление элементов");

Console.WriteLine();

linkedList.Delete(10);

linkedList.Delete(2);

linkedList.Delete(22);

linkedList.Delete(11);

Console.Write("Элементы списка: ");

linkedList.PrintLinkedList();

Console.WriteLine("Пустой ли список ? " + linkedList.IsEmpty());

Console.WriteLine("Кол-во элементов в списке: " + linkedList.Count);

Console.WriteLine("Есть ли число 11 в списке ? " + linkedList.Search(11));

Console.ReadLine();

}

}

}

public class Item<T>

{

private T data = default(T);

public T Data

{

get

{

return data;

}

set

{

if (value != null)

data = value;

}

}

public Item<T> Next { get; set; }

public Item(T data)

{

Data = data;

}

public override string ToString()

{

return $"{Data}";

}

}

public class LinkedList<T> : IEnumerable

{

public Item<T> First { get; set; }

public Item<T> Last { get; set; }

public int Count { get; set; }

public LinkedList()

{

Clear();

}

public void Delete(T data)

{

if (First == null)

return;

if(First.Data.Equals(data))

{

First = First.Next;

Count--;

return;

}

var current = First.Next;

var previous = First;

while(current != null)

{

if(current.Data.Equals(data))

{

previous.Next = current.Next;

Count--;

if (First.Next == null)

Last = null;

return;

}

previous = current;

current = current.Next;

}

}

public void Add(T data)

{

var item = new Item<T>(data);

if (Last != null)

{

var previous = First;

var current = previous.Next;

if (Compare(previous, item))

{

First = item;

First.Next = previous;

Count++;

return;

}

while(current != null)

{

if (Compare(current, item))

{

item.Next = current;

previous.Next = item;

Count++;

return;

}

previous = current;

current = previous.Next;

}

previous.Next = item;

Last = item;

Count++;

}

else

{

First = item;

Last = item;

Count = 1;

}

}

private bool Compare(Item<T> item1, Item<T> item2)

{

if(item1.Data is int || item1.Data is double)

{

return Convert.ToDouble(item1.Data) >= Convert.ToDouble(item2.Data);

}

else

{

List<string> mass = new List<string>();

mass.Add(item1.Data.ToString());

mass.Add(item2.Data.ToString());

mass.OrderBy(x => x);

if (mass[0] == item1.Data.ToString())

return true;

else

return false;

}

}

public bool IsEmpty()

{

if (First == null && Last == null)

return true;

return false;

}

public bool Search(T data)

{

foreach(var item in this)

if (item.Equals(data))

return true;

return false;

}

public void Clear()

{

First = null;

Last = null;

Count = 0;

}

public void Read(LinkedList<int> list)

{

List<int> data = Console.ReadLine().Split(" ").Select(x => Convert.ToInt32(x)).ToList();

foreach (var item in data)

list.Add(item);

}

public void PrintLinkedList()

{

foreach(var item in this)

{

Console.Write(item.ToString() + " ");

}

Console.WriteLine();

}

public IEnumerator GetEnumerator()

{

var current = First;

while(current != null)

{

yield return current.Data;

current = current.Next;

}

}

}