Опис елемента однозв’язного списку

#pragma once

class Element //Елемент однозв"язного списку

{

public:

double Data; //Інформаційне поле

Element\* Link; //Поле зв"язку

Element(double data, Element\* link);

Element();

};

---------------

#include "stdafx.h"

#include "Element.h"

Element::Element(double data, Element \* link)

{

Data = data;

Link = link;

}

Element::Element():Element(0,0)

{

}

===================Стек=======================

#pragma once

//Динамічна структура даних

//Додавання лише в кінець

//Вилучення лише з кінця

//Видалення з середини заборонено

//LIFO (Last In First Out)

#include "Element.h"

class Stack

{

public:

int Length;

Element\* Top;

Stack();

void push(double newData);

bool pop(double& removedData);

};

-----------------------------------

#include "stdafx.h"

#include "Stack.h"

Stack::Stack()

{

Top = 0;

Length = 0;

}

void Stack::push(double newData)

{

Element\* current = new Element(newData, Top);

Top = current;

Length++;

}

bool Stack::pop(double & removedData)

{

if (Top==0)

return false;

removedData = Top->Data;

Element\* current = Top;

Top = Top->Link;

delete current;

Length--;

return true;

}

============================== Черга ================

#pragma once

//Динамічна структура даних

//Додавання лише в кінець

//Вилучення лише з початку

//Видалення з середини заборонено

//FIFO (First In First Out)

#include "Element.h"

class Queue

{

public:

int Length;

Element\* First;

Element\* Last;

Queue();

void add(double newData);

bool remove(double& removedData);

};

-----------------------------------

#include "stdafx.h"

#include "Queue.h"

Queue::Queue()

{

First = 0;

Last = 0;

Length = 0;

}

void Queue::add(double newData)

{

Element\* current = new Element(newData, 0);

if (First==0)

{

First = current;

}

else

{

Last->Link = current;

}

Last = current;

Length++;

}

bool Queue::remove(double & removedData)

{

if (First==0)

return false;

removedData = First->Data;

Element\* current = First;

First = First->Link;

delete current;

if (First==0)

{

Last = 0;

}

Length--;

return true;

}

============================Main ===============

// ConsoleApplication117.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include"Stack.h"

#include"Queue.h"

using namespace std;

int main()

{

Stack\* stack = new Stack();

stack->push(33);

stack->push(25);

stack->push(20);

stack->push(11);

double number;

while (stack->pop(number))

{

printf("%f\n", number);

}

printf("-------------------------\n");

Queue\* queue = new Queue();

queue->add(2);

queue->add(7);

queue->add(8);

queue->add(9);

while (queue->remove(number))

{

printf("%f\n", number);

}

system("pause");

return 0;

}