Задача. Розробити клас, який представляє динамічний масив (кількість елементів динамічно може збільшуватися і зменшуватися).

----------------------- Array.h

#pragma once

#include<string>

using namespace std;

class Array

{

private:

double\* data\_; //Покажчик на масив з даними

int length\_, maxLength\_; //реальні і максимальна кількість елементів

void set\_maxLength(int value);

public:

bool isIndexCorrect(int index);

double& getElement(int index);

int get\_length();

int get\_maxLength();

void addElement(double value);

void removeAt(int index);

void remove(double value);

int indexOf(double value);

double getSum();

string toString();

Array(int maxLength);

Array();

~Array();

};

------------------- Array.cpp

#include "stdafx.h"

#include "Array.h"

#include <math.h>

int Array::indexOf(double value)

{

for (int i = 0; i < length\_; i++)

{

if (abs(data\_[i] - value)<0.00001) //if (data\_[i]==value) //

{

return i;

}

}

return -1;

}

double Array::getSum()

{

double s = 0;

for (int i = 0; i < length\_; i++)

{

s += data\_[i];

}

return s;

}

string Array::toString()

{

if (length\_==0)

{

return "array is empty";

}

string s = to\_string(data\_[0]);

for (int i = 1; i < length\_; i++)

{

s += ", " +to\_string( data\_[i]);

}

return s;

}

Array::Array(int maxLength)

{

set\_maxLength(maxLength);

}

Array::Array():Array(15) //Якщо не задано то за замовчуванням кількість максимальна дорівнює 15 (нехай)

{

}

Array::~Array()

{

delete[] data\_;

}

void Array::set\_maxLength(int value)

{

if (value>0)

{

maxLength\_ = value; //Збергіємо початкову макксимальну кількість

data\_ = new double[maxLength\_]; //Виділяємо пам"ять для масиву

length\_ = 0; //Початкова справжня кількість спочатку дорівнює 0

}

else

{

throw "Error";

}

}

bool Array::isIndexCorrect(int index)

{

return index>=0 && index<length\_;

}

double & Array::getElement(int index)

{

if (isIndexCorrect(index))

{

return data\_[index];

}

else

{

throw "Index is incorrect";

}

}

int Array::get\_length()

{

return length\_;

}

int Array::get\_maxLength()

{

return maxLength\_;

}

void Array::addElement(double value)

{

if (length\_==maxLength\_) //Якщо вільне місце вичерпано

{

maxLength\_ \*= 2; //Максимальну кількість елементів збільшуємо удвічі (ми так вирішили)

double\* temp = new double[maxLength\_]; //створюємо новий масив (удвічі більший)

for (int i = 0; i < length\_; i++) //копіюємо у новий масив елементи з старого

{

temp[i] = data\_[i];

}

delete[]data\_; //Знищуємо старий масив

data\_ = temp; //Запам"ятовуємо адресу нового

}

//------------------------ Додаємо новий елемент

data\_[length\_] = value;

length\_++;

}

void Array::removeAt(int index)

{

if (isIndexCorrect(index))

{

for (int i = index+1; i < length\_; i++)

{

data\_[i - 1] = data\_[i];

}

length\_--;

}

else

{

throw "Index is incorrect";

}

}

void Array::remove(double value)

{

int index = indexOf(value);

if (index!=-1)

{

removeAt(index);

}

else

{

throw "no such element";

}

}

--------------------------- Main

// ConsoleApplication36.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "Array.h"

#include<iostream>

using namespace std;

int main()

{

//------------------ Ввяться числа поки не 0 (скільки наперед не знаємо). Вивисти ті, які більші за суму

Array\* a = new Array();

double el;

do

{

printf("el=");

cin >> el;

a->addElement(el);

} while (el!=0);

double s = a->getSum();

for (int i = 0; i < a->get\_length(); i++)

{

if (a->getElement(i)>s)

{

printf("%f, ", a->getElement(i));

}

}

delete a;

//--------------------------------------------

system("pause");

//------------------ Задача 2. Додавання елементів, вилучення через меню

Array\* array= new Array();

int answer, index;

double element;

do

{

system("cls");

printf("1.Add\n");

printf("2.Remove\n");

printf("3.Remove at index\n");

printf("4.Print\n");

printf("5.Exit\n");

printf("Your choise:");

cin >> answer;

switch (answer)

{

case 1:printf("Element=");

cin >> element;

array->addElement(element);

break;

case 2:printf("Element=");

cin >> element;

array->remove(element);

break;

case 3:printf("Index =");

cin >> index;

array->removeAt(index);

break;

case 4: printf("Array:%s\n", array->toString().data());

break;

}

system("pause");

} while (answer!=5);

delete array;

return 0;

}