НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.2 »

*Звiт з лабораторної роботи №10*

*Тема:* Шаблонні функції

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Асистент Віктор ЧЕЛАК

Асистент Владислав ЯЛОВЕГА

Харків – 2020

***Мета:*** Отримати базові знання про шаблонізацію (узагальнення) на  
основі шаблонних функцій

Варіант 6.

*Загальне завдання.* (Задача не пов’язана з попередніми роботами).  
Створити клас, який не має полів, а усі необхідні дані передаються  
безпосередньо у функції.  
Клас має виконувати наступні дії:  
− виводити вміст масиву на екран;  
− визначати індекс переданого елементу в заданому масиві;  
− сортувати елементи масиву;  
− визначати значення мінімального елементу масиву.  
При цьому необхідно продемонструвати роботу програми як з

використанням стандартних типів даних так і типів, створених користувачем.

Тексt програми

prototaype.h

……….

without\_fields.h

#pragma once

#include "prototaype.h"

#define ARRAY\_SIZE 255

class massiv

{

protected:

int sizes = 5;

double mas[5] = { 12.5,42.2,1,-4,0 };

public:

massiv() { ; }

~massiv() { ; }

int GET\_sizes() { return sizes; }

double& operator[] (int index) { return mas[index]; }

};

class without\_fields

{

public:

template < typename T>

void printEl(int size,T EL[])

{

cout << "Ваш масив: ";

for(int i=0;i<size;i++)

cout << EL[i] << " ";

cout << endl << endl;;

}

template < typename T>

void poisk(int size, T\*EL )

{

T poshuk;

cout << "Що хочете знайти у масиві :" << endl;

cin >> poshuk;

cout << endl;

int a = 0;

for (int i = 0; i < size; i++)

if (EL[i] == poshuk)

{

cout <<++a<< "Індекс вашого елементу: " << i << endl;

}

if(a==0)

cout<< "Такого елементу немає в масиві" << endl;

}

template < typename T>

void print\_min\_Element(int size, T\* EL)

{

cout << "Мінімальний елемент масиву: ";

T a= EL[0];

for(int i=0;i<size;i++)

a=GetMin(a, EL[i]);

cout << a << endl<<endl;

}

template <typename T>

T\* sortEL(int sizes, T\*el)

{

T temp;

bool pr; // для проверки отсортирован ли массив

do

{

pr = false;

for (int i = 0; i < sizes - 1; i++) // проходим слева направо

if ( el[i] > el[i + 1])

{

temp = el[i];

el[i] = el[i +1];

el[i +1] = temp;

pr = true;

}

} while (pr);

cout << " \*сортування завершено " << endl;

return el;

}

template <typename T>

T GetMin(T aLeft, T aRight)

{

return aLeft < aRight ?aLeft :aRight ;

}

};

Main.cpp

#include "prototaype.h"

#include "without\_fields.h"

typedef double vegas;

typedef int real;

int main()

{

setlocale(LC\_CTYPE, "ru");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

without\_fields h;

int sizes = 0, EL[ARRAY\_SIZE];

h.Riad\_FL(sizes, EL);

system("cls");

cout << "Стандартні типи данних"<<endl;

h.printEl(sizes, EL);

h.print\_min\_Element(sizes, EL);

//index();

h.sortEL(sizes, EL);

h.printEl(sizes, EL);

system("pause&cls");

cout << "Типи даних створених користувачем."<<endl;

real size = 6;

vegas array[6] = { 10, -2, 2.25, 9.1, -9.11, 0.25 };

h.printEl(size, array);

h.print\_min\_Element(sizes, array);

//index();

h.sortEL(size, array);

h.printEl(size, array);

return 0;

}

**Результати роботи програми:**

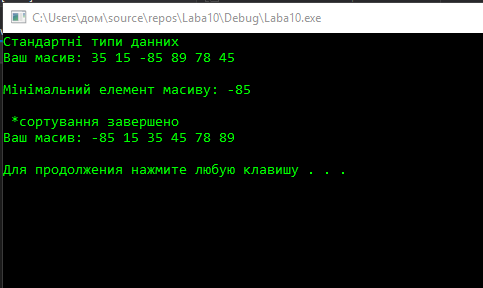
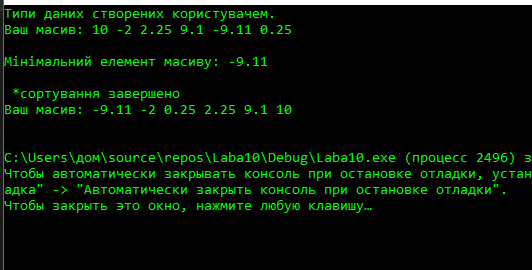


Рисунок 1 – Результати роботи програми з стандартним типом int



РиРисунок 2 – Результати роботи програми з типом dooble

**Висновки:**

В результаті виконання лабораторної роботи отримав базові знання про шаблонізацію (узагальнення) на основі шаблонних функцій