

## ЛАБОРАТОРНА РОБОТА №20 «Обробка одновимірних масивів»

**Мета роботи:** вивчити основні функціональні можливості C++ Builder для обробки одновимірних масивів.

### Хід роботи

**Завдання 1 (4 бали).** У деякому масиві  $x$  дійсних чисел розміром  $n$  знайти мінімальний елемент та його індекс або максимальний елемент та його індекс, або за завданням користувача і мінімальний, і максимальний елементи та їхні індекси.

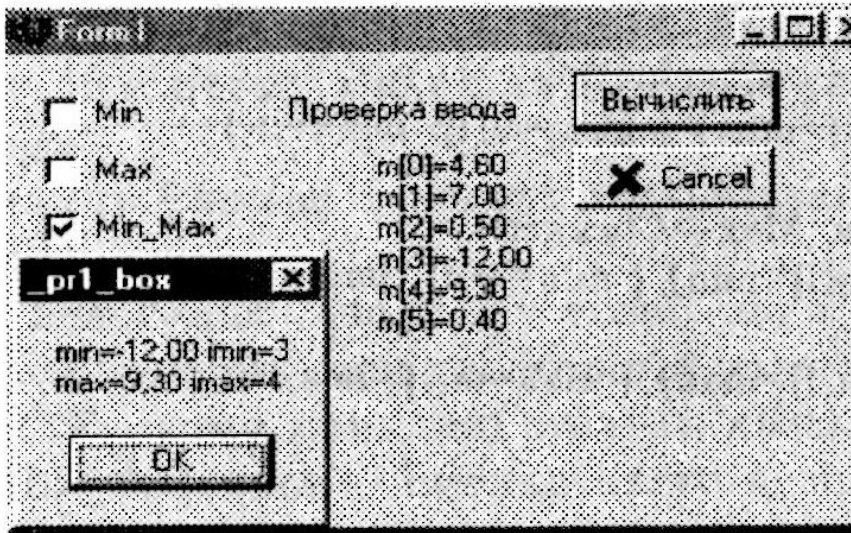
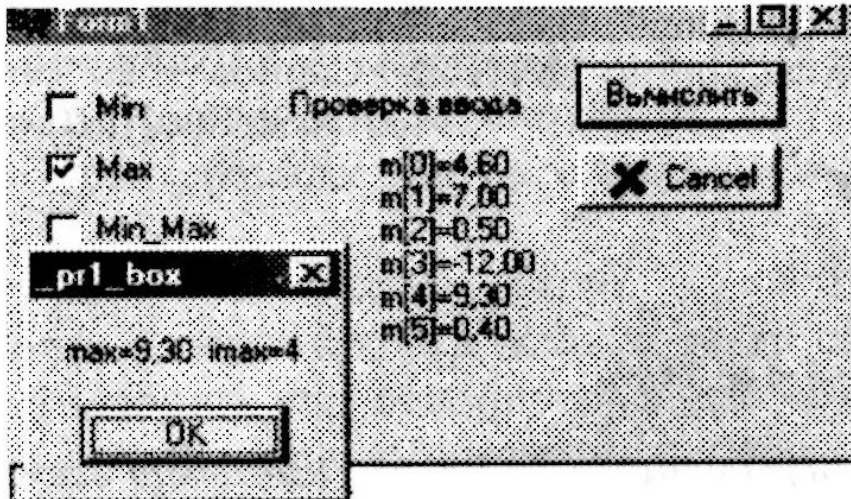
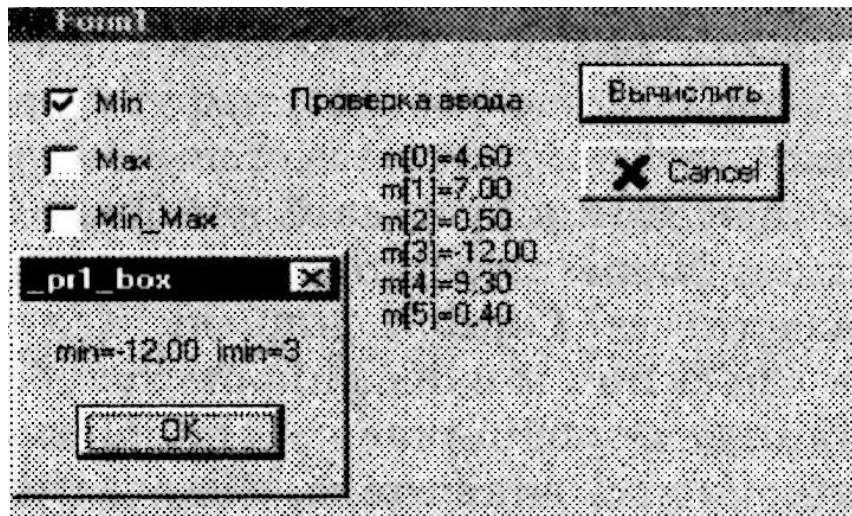
Розташуємо на формі такі компоненти: **Label1** — з текстом у полі **Caption** «Перевірка введення», **Label2** — з ім'ям **LbVivod** для виведення масиву у вікно форми, кнопку **Button1** з текстом «Обчислити» і кнопку **BitBtn** із зображенням та текстом «Cancel».

Крім того, вертикально розташуємо три незалежних одна від одної кнопки **CheckBox**, за допомогою яких можна задавати режими роботи програми в будь-якій комбінації їхнього активного стану. У полі **Name** цих кнопок введені імена: **Min**, **Max** і **Min\_Max**, а властивості **Caption** очищені. Для визначення стану компонента **CheckBox** можна використовувати властивість **Clicked**, яка має значення: «**true**» — якщо кнопка включена і «**false**» — якщо кнопка виключена, або властивість **State**, яка може мати три значення: **cbChecked** — якщо кнопка включена, **cbUnchecked** — якщо виключена і **cbGrayed** — якщо знаходиться в неактивному стані. Для ознайомлення з методикою роботи кнопок **CheckBox** у програмі використано обидві властивості (**Clicked** і **State**), що можуть визначити їх стан.

Після всіх підготовчих робіт програма запускається на виконання. З появою робочого вікна форми на екрані треба установити спочатку режими роботи програми шляхом включення відповідних кнопок. Для активації кожної з кнопок досить клацнути на ній мишкою. Повторне клацання скасовує виділення. Потім треба натиснути кнопку «Вычислить», тобто включити виконання програми в роботу. Введення елементів масиву здійснюється за допомогою функції **InputBox()**, що виводить відповідне вікно.

У поле введення вікна **InputBox** треба набрати необхідне число і натиснути або кнопку **OK**, або клавішу **Enter**. Якщо відбулася помилка введення, необхідно її скасувати, натиснувши кнопку **Cancel**, і повторити введення. Відразу в поле мітки **LbVivod** під заголовком «Перевірка введення» буде виводитися введене значення. По закінченні цього процесу на екрані з'явиться вікно **ShowMessage()** з результатом розв'язання задачі. Якщо була обрана не одна кнопка **CheckBox**, то після виведення чергового результату треба у вікні виведення натиснути **OK**, після чого з'явиться результат обробки наступного заданого режиму тощо. Вікно виведення може з'явитися в будь-якому місці форми, тому що функція **ShowMessage()** не містить його координат. Якщо це вікно закриває які-небудь потрібні елементи форми, його можна перетягнути в інше зручне місце, захопивши мишкою заголовок вікна.

Розглянутий додаток було тричі запущено на виконання при різних активних кнопках **CheckBox**, що видно з трьох виведених вікон і результатів роботи з програмним кодом.



```

#include <vcl.h>
#pragma hdrstop
#include<Dialogs.hpp>
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
const n=6;
AnsiString st;
int i, imin, imax;
float min, max, m[n];
for (i=0; i<n; i++)
{ st=InputBox("Ввести масив m", "Введіть " + IntToStr(i) +
"елемент", "");
m[i]=StrToFloat(st);
LbVivod->Caption=LbVivod->Caption+"m["+IntToStr(i)+"]=" +
FloatToStrF(m[i], ffFixed, 5, 2)+'\n';}
min=m[0];
imin=0;
max=m[0];
imax=0;
for(i=0;i<n;i++)
// цикл для знаходження максимального і мінімального елементів
{if (Min->Checked)
if(m[i]<min) {min=m[i];imin=i;}
else
{if (Max->Checked)
if(m[i]>max) {max=m[i];imax=i;} }
else
if (Min_Max->Checked)
{if (m[i]<min) {min=m[i];imin=i;}
else
if (m[i]>max) {max=m[i];imax=i;}
}
}
}
}

```

```
if (Min->State==cbChecked)
ShowMessage ("min="+FloatToStrF (min, ffFixed, 6, 2) +
imin="+IntToStr (imin) );
    if (Max->State==cbChecked)
        ShowMessage ("max="+FloatToStrF (max, ffFixed, 6, 2) +
imax="+IntToStr (imax) );
if (Min_Max->State==cbChecked)
ShowMessage ("min="+FloatToStrF (min, ffFixed, 6, 2) +
imin="+IntToStr (imin) +"\n"+
"max="+FloatToStrF (max, ffFixed, 6, 2) + " imax="+IntToStr (imax) );
}
```

У **C++ Builder** є кнопки, що завжди поєднуються в групи, але активною в групі може бути лише одна кнопка, при вмиканні якої інші автоматично вимикаються. До таких компонентів відноситься кнопка **RadioButton**, що одна ніколи не використовується, і група кнопок **RadioGroup**. Методику використання цих компонентів розглянемо в такому прикладі.

#### Завдання 2 (+1 бал)

**1. За вказівкою користувача обчислити кількість та суму додатних елементів або додаток від'ємних непарних елементів масиву  $m(15)$ .**

**2. За вказівкою користувача з додатних або від'ємних елементів матриці  $mt(4,5)$  утворити одновимірний масив, елементи якого розмістити в зворотній послідовності.**

**3. За вказівкою користувача обчислити або добуток та суму елементів розташованих на парних місцях масиву  $x(15)$ , або непарні елементи масиву відсортувати за спаданням.**