Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

Кафедра радіофізики та комп’ютерних технологій

ЛАБОРАТОРНА РОБОТА № 14

з курсу “Алгоритмізація та програмування”

**«Ряд Фур’є із структурою і файлом у VC++»**

Виконав:

Cт. Гр. ФЕІ-15

Кочержук Богдан Романович

Перевірив:

Ас. Дзіковський В. Є.

Львів 2021

Звіт

**Тема:** ряд Фур’є із структурою і файлом у VC++

**Мета роботи:** познайомитись із принципами використання структур і файлів у VС++.

**ХІД РОБОТИ**

1. **Запускаємо середовище у режимі роботи Пустий Проект**
2. **Створюємо новий віконний проєкт**
3. **Текст програми:**

struct Zapys

{

double Xe[1000];

double Ye[1000];

double Yg[1000];

double c[50];

double Kx, Ky, Zx, Zy;

double a1, b1, Tp;

int Ne, Ngr, Ng;

double minYg, maxYg, maxx, maxy, minx, miny;

double krx, kry, xx, yy, Gx, Gy;

int KrokX, KrokY, L;

};

Zapys F\_GRF;

#pragma endregion

double f(double x)

{

F\_GRF.Tp = F\_GRF.b1 - F\_GRF.a1;

if (x < F\_GRF.Tp / 2.0)return 2.0;

else {

if ((x >= F\_GRF.Tp / 2.0) && (x < 3.0 \* F\_GRF.Tp / 4.0))

return 4.0 \* (F\_GRF.Tp - 2.0 \* x) / F\_GRF.Tp;

else return 8.0 \* (x - F\_GRF.Tp) / F\_GRF.Tp;

}

}

void TabF(double Xe[1000], double Ye[1000])

{

double h;

int i;

h = (F\_GRF.b1 - F\_GRF.a1) / F\_GRF.Ne;

Xe[0] = F\_GRF.a1;

for (i = 0; i <= F\_GRF.Ne - 1; i++) {

F\_GRF.Ye[i] = f(F\_GRF.Xe[i]);

F\_GRF.Xe[i + 1] = F\_GRF.Xe[i] + h;

}

}

void Furje(double Xe[1000], double Ye[1000], int Ne, double Yg[1000], double c[50], double TP)

{

double a[50]; // Масив a із коефіцієнтами ряду Фур’є

double b[50]; // Масив b із коефіцієнтами ряду Фур’є

double w, KOM, S, G, D;

int i, k;

F\_GRF.Ng = Convert::ToInt32(textBox3->Text); // Вводимо кількість гармонік

F\_GRF.a1 = Convert::ToDouble(textBox1->Text);

F\_GRF.b1 = Convert::ToDouble(textBox2->Text);

F\_GRF.Tp = F\_GRF.b1 - F\_GRF.a1;

w = 6.2831853 / F\_GRF.Tp;

for (k = 1; k <= F\_GRF.Ng; k++)

{

KOM = k \* w;

G = 0.0;

D = 0.0;

for (i = 1; i <= F\_GRF.Ne; i++)

{

S = KOM \* F\_GRF.Xe[i];

G = G + F\_GRF.Ye[i] \* Math::Cos(S);

D = D + F\_GRF.Ye[i] \* Math::Sin(S);

}

a[k] = 2 \* G / F\_GRF.Ne;

b[k] = 2 \* D / F\_GRF.Ne;

c[k] = Math::Sqrt(a[k] \* a[k] + b[k] \* b[k]);

}

a[0] = 0.0;

for (i = 1; i <= F\_GRF.Ne; i++)

a[0] = a[0] + F\_GRF.Ye[i];

a[0] = a[0] / Ne;//

for (i = 0; i <= F\_GRF.Ne - 1; i++) {

S = 0;

D = F\_GRF.Xe[i] \* w;

for (k = 1; k <= F\_GRF.Ng; k++) {

KOM = k \* D;

S = S + b[k] \* Math::Sin(KOM) + a[k] \* Math::Cos(KOM);

}

F\_GRF.Yg[i] = a[0] + S;

}

return; // Завершення тіла функції Furje

}

void Garm(int Ng, double c[50])

{

int i, KrokXG, x;

double MaxC, KyC, w;

Graphics^ g = pictureBox1->CreateGraphics();

Pen^ pen1 = gcnew Pen(Color::Black, (float)(numericUpDown1->Value));

Pen^ pen2 = gcnew Pen(Color::Blue, (float)(numericUpDown2->Value));

Pen^ pen3 = gcnew Pen(Color::Silver, (float)(numericUpDown3->Value));

Pen^ pen4 = gcnew Pen(Color::Green, (float)(numericUpDown4->Value));

int pb\_Height = pictureBox1->Height;

int pb\_Width = pictureBox1->Width;

KrokXG = (pb\_Width - 2 \* F\_GRF.L) / Ng;

MaxC = c[1];

for (i = 2; i <= Ng; i++)

if (c[i] > MaxC) MaxC = c[i];

KyC = (pb\_Height / 2) / MaxC;

g->DrawLine(pen2, F\_GRF.L, F\_GRF.L + 20, F\_GRF.L + 10, F\_GRF.L + 10);

g->DrawLine(pen2, F\_GRF.L + 20, F\_GRF.L + 20, F\_GRF.L + 10, F\_GRF.L + 10);

g->DrawLine(pen2, F\_GRF.L + 10, pb\_Height - 50, pb\_Width - 20, pb\_Height - 50);

g->DrawLine(pen2, F\_GRF.L + 10, pb\_Height - 50, F\_GRF.L + 10, F\_GRF.L + 10);

g->DrawLine(pen2, pb\_Width - 40, pb\_Height - 60, pb\_Width - 20, pb\_Height - 50);

g->DrawString("C", gcnew Drawing::Font("Times", 14),

Brushes::Black, (float)F\_GRF.L - 15, (float)F\_GRF.L + 5);

g->DrawString("W", gcnew Drawing::Font("Times", 14), Brushes::Black, (float)pb\_Width - 60.0f,

(float)pb\_Height - 50.0f);

x = KrokXG + 20;

w = 6.2831853 / (F\_GRF.b1 - F\_GRF.a1);

for (i = 1; i <= Ng; i++)

{

g->DrawLine(pen1, (int)x + 3, pb\_Height - 50, x + 3, pb\_Height - 50 - (int)(KyC \* c[i]));

String^ s = String::Format("{0:F3}", KyC \* c[i]);

g->DrawString(s, gcnew Drawing::Font("Times", 12), Brushes::Black, (float)x,

(float)pb\_Height - (float)(KyC \* c[i]) - 70.0f);

g->DrawEllipse(pen2, (int)x, pb\_Height - (int)(KyC \* c[i]) - 55, 5, 5);

g->DrawString(Convert::ToString(i), gcnew Drawing::Font("Times", 12), Brushes::Black,

(float)x - 5.0f, (float)pb\_Height - 50.0f);

x = x + KrokXG;

}

x = KrokXG + 19;

String^ s = String::Format("W={0:F3}", w);

g->DrawString(s, gcnew Drawing::Font("Times", 12), Brushes::Black, (float)x - 20.0f,

(float)pb\_Height - 35.0f);

return;

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

int i;

Pen^ pen1 = gcnew Pen(Color::Red, (float)(numericUpDown1->Value));

Pen^ pen2 = gcnew Pen(Color::Blue, (float)(numericUpDown2->Value));

Pen^ pen3 = gcnew Pen(Color::Silver, (float)(numericUpDown3->Value));

Pen^ pen4 = gcnew Pen(Color::Green, (float)(numericUpDown4->Value));

Graphics^ g = pictureBox1->CreateGraphics();

g->Clear(Color::White);

int pb\_Height = pictureBox1->Height;

int pb\_Width = pictureBox1->Width;

F\_GRF.L = 40;

F\_GRF.Ne = Convert::ToInt32(textBox4->Text);

F\_GRF.a1 = Convert::ToDouble(textBox1->Text);

F\_GRF.b1 = Convert::ToDouble(textBox2->Text);

F\_GRF.Ngr = F\_GRF.Ne;

TabF(F\_GRF.Xe, F\_GRF.Ye);

Furje(F\_GRF.Xe, F\_GRF.Ye, F\_GRF.Ne, F\_GRF.Yg, F\_GRF.c, F\_GRF.Tp);

F\_GRF.minYg = F\_GRF.Yg[0];

F\_GRF.maxYg = F\_GRF.Yg[0];

for (i = 1; i <= F\_GRF.Ngr - 1; i++) {

if (F\_GRF.maxYg < F\_GRF.Yg[i])F\_GRF.maxYg = F\_GRF.Yg[i];

if (F\_GRF.minYg > F\_GRF.Yg[i])F\_GRF.minYg = F\_GRF.Yg[i];

}

F\_GRF.minx = F\_GRF.Xe[0];

F\_GRF.maxx = F\_GRF.Xe[F\_GRF.Ne - 1];

F\_GRF.miny = F\_GRF.Ye[0];

F\_GRF.maxy = F\_GRF.Ye[0];

for (i = 1; i <= F\_GRF.Ne - 1; i++) {

if (F\_GRF.maxy < F\_GRF.Ye[i])F\_GRF.maxy = F\_GRF.Ye[i];

if (F\_GRF.miny > F\_GRF.Ye[i])F\_GRF.miny = F\_GRF.Ye[i];

}

if (F\_GRF.maxy < F\_GRF.maxYg)F\_GRF.maxy = F\_GRF.maxYg;

if (F\_GRF.miny > F\_GRF.minYg)F\_GRF.miny = F\_GRF.minYg;

F\_GRF.Kx = (pb\_Width - 2 \* F\_GRF.L) / (F\_GRF.maxx - F\_GRF.minx);

F\_GRF.Ky = (pb\_Height - 2 \* F\_GRF.L) / (F\_GRF.miny - F\_GRF.maxy);

F\_GRF.Zx = (pb\_Width \* F\_GRF.minx - F\_GRF.L \* (F\_GRF.maxx + F\_GRF.minx)) / (F\_GRF.minx - F\_GRF.maxx);

F\_GRF.Zy = (pb\_Height \* F\_GRF.maxy - F\_GRF.L \* (F\_GRF.miny + F\_GRF.maxy)) / (F\_GRF.maxy - F\_GRF.miny);

std::ofstream binout("Grf\_file.dat", std::ios::binary);

binout.write((char\*)&F\_GRF, sizeof(Zapys));

binout.close();

MessageBox::Show("Дані записано у файл");

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

int i;

Pen^ pen1 = gcnew Pen(Color::Red, (float)(numericUpDown1->Value));

Pen^ pen2 = gcnew Pen(Color::Blue, (float)(numericUpDown2->Value));

Pen^ pen3 = gcnew Pen(Color::Silver, (float)(numericUpDown3->Value));

Pen^ pen4 = gcnew Pen(Color::Green, (float)(numericUpDown4->Value));

Graphics^ g = pictureBox1->CreateGraphics();

g->Clear(Color::White);

int pb\_Height = pictureBox1->Height;

int pb\_Width = pictureBox1->Width;

std::ifstream bincin("Grf\_file.dat", std::ios::binary);

bincin.read((char\*)&F\_GRF, sizeof(Zapys));

bincin.close();

if (F\_GRF.minx \* F\_GRF.maxx <= 0.0)F\_GRF.Gx = 0.0;

if (F\_GRF.minx \* F\_GRF.maxx > 0.0)F\_GRF.Gx = F\_GRF.minx;

if (F\_GRF.minx \* F\_GRF.maxx > 0.0 && F\_GRF.minx < 0.0)F\_GRF.Gx = F\_GRF.maxx;

if (F\_GRF.miny \* F\_GRF.maxy <= 0.0)F\_GRF.Gy = 0.0;

if (F\_GRF.miny \* F\_GRF.maxy > 0.0 && F\_GRF.miny > 0.0)F\_GRF.Gy = F\_GRF.miny;

if (F\_GRF.miny \* F\_GRF.maxy > 0.0 && F\_GRF.miny < 0.0)F\_GRF.Gy = F\_GRF.maxy;

F\_GRF.KrokX = (pb\_Width - 2 \* F\_GRF.L) / 10;

F\_GRF.KrokY = (pb\_Height - 2 \* F\_GRF.L) / 10;

for (i = 1; i <= 7; i++) {

g->DrawLine(pen3, F\_GRF.L, Math::Round(F\_GRF.Ky \* F\_GRF.Gy + F\_GRF.Zy, 4) + i \* F\_GRF.KrokY, pb\_Width - F\_GRF.L,

Math::Round(F\_GRF.Ky \* F\_GRF.Gy + F\_GRF.Zy, 4) + i \* F\_GRF.KrokY);

g->DrawLine(pen3, F\_GRF.L, Math::Round(F\_GRF.Ky \* F\_GRF.Gy + F\_GRF.Zy, 4) - i \* F\_GRF.KrokY, pb\_Width - F\_GRF.L,

Math::Round(F\_GRF.Ky \* F\_GRF.Gy + F\_GRF.Zy, 4) - i \* F\_GRF.KrokY);

}

for (i = 1; i <= 10; i++) {

g->DrawLine(pen3, Math::Round(F\_GRF.Kx \* F\_GRF.Gx + F\_GRF.Zx, 4) + i \* F\_GRF.KrokX, F\_GRF.L - 15,

Math::Round(F\_GRF.Kx \* F\_GRF.Gx + F\_GRF.Zx, 4) + i \* F\_GRF.KrokX, pb\_Height - F\_GRF.L + 6);

g->DrawLine(pen3, Math::Round(F\_GRF.Kx \* F\_GRF.Gx + F\_GRF.Zx, 4) - i \* F\_GRF.KrokX, F\_GRF.L,

Math::Round(F\_GRF.Kx \* F\_GRF.Gx + F\_GRF.Zx, 4) - i \* F\_GRF.KrokX, pb\_Height - F\_GRF.L);

}

F\_GRF.xx = F\_GRF.minx;

F\_GRF.yy = F\_GRF.maxy;

F\_GRF.krx = (F\_GRF.maxx - F\_GRF.minx) / 10.0;

F\_GRF.kry = (F\_GRF.maxy - F\_GRF.miny) / 11.0;

for (i = 0; i < 12; i++) {

g->DrawString(Convert::ToString(Math::Round(F\_GRF.xx, 1)), gcnew Drawing::Font("Times", 8),

Brushes::Black, F\_GRF.L + 4 + i \* F\_GRF.KrokX,

Math::Round(F\_GRF.Ky \* F\_GRF.Gy + F\_GRF.Zy, 4) - F\_GRF.L + 43.0f);

g->DrawString(Convert::ToString(Math::Round(F\_GRF.yy, 1)), gcnew Drawing::Font("Times", 8),

Brushes::Black, Math::Round(F\_GRF.Kx \* F\_GRF.Gx + F\_GRF.Zx, 4) - F\_GRF.L + 10.0f,

(float)(F\_GRF.L + i \* F\_GRF.KrokY) - 5.0f);

F\_GRF.xx = F\_GRF.xx + F\_GRF.krx;

F\_GRF.yy = F\_GRF.yy - F\_GRF.kry;

}

g->DrawLine(pen2, F\_GRF.L, Math::Round(F\_GRF.Ky \* F\_GRF.Gy + F\_GRF.Zy, 4), Math::Round(pb\_Width - 10, 4),

Math::Round(F\_GRF.Ky \* F\_GRF.Gy + F\_GRF.Zy, 4));

g->DrawLine(pen2, Math::Round(F\_GRF.Kx \* F\_GRF.Gx + F\_GRF.Zx, 4), 10, Math::Round(F\_GRF.Kx \* F\_GRF.Gx + F\_GRF.Zx, 4),

Math::Round(pb\_Height - 10, 4));

for (i = 1; i <= F\_GRF.Ne - 1; i++) {

g->DrawLine(pen4, Math::Round(F\_GRF.Kx \* (F\_GRF.Xe[i - 1]) + F\_GRF.Zx, 4), Math::Round(F\_GRF.Ky \* (F\_GRF.Yg[i - 1]) + F\_GRF.Zy, 4),

Math::Round(F\_GRF.Kx \* (F\_GRF.Xe[i]) + F\_GRF.Zx, 4), Convert::ToInt32(Math::Round(F\_GRF.Ky \* (F\_GRF.Yg[i]) + F\_GRF.Zy, 4)));

g->DrawLine(pen1, Math::Round(F\_GRF.Kx \* F\_GRF.Xe[i - 1] + F\_GRF.Zx, 4), Math::Round(F\_GRF.Ky \* F\_GRF.Ye[i - 1] + F\_GRF.Zy, 4),

Math::Round(F\_GRF.Kx \* F\_GRF.Xe[i] + F\_GRF.Zx, 4), Convert::ToInt32(Math::Round(F\_GRF.Ky \* F\_GRF.Ye[i] + F\_GRF.Zy, 4)));

}

MessageBox::Show("Показати гармоніки");

g->Clear(Color::White);

Garm(F\_GRF.Ng, F\_GRF.c);

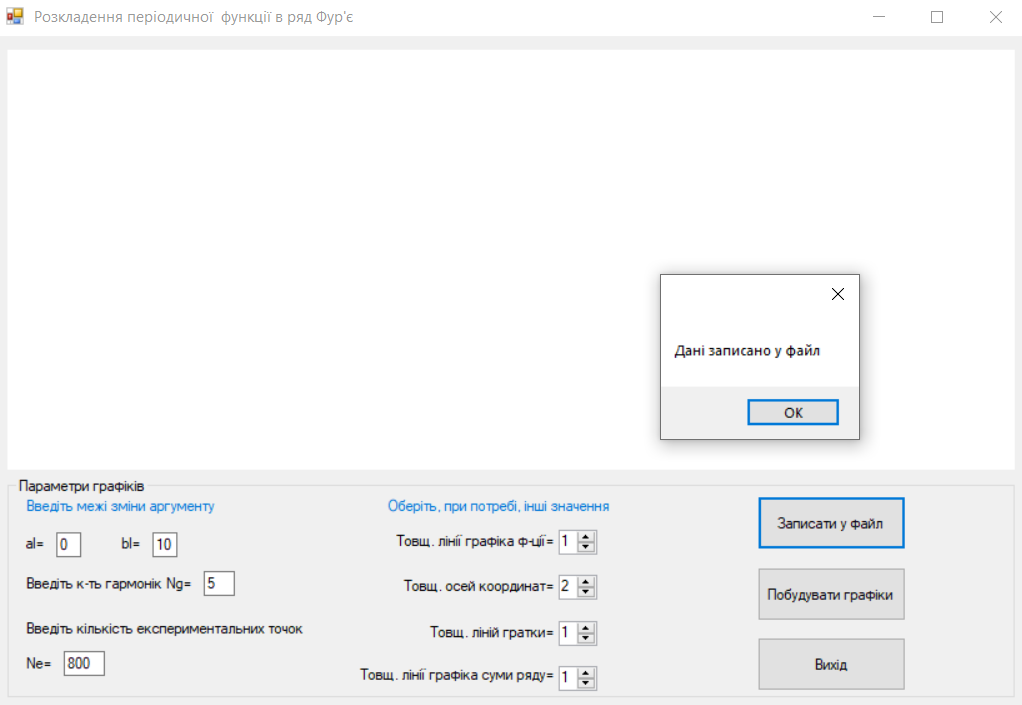
}

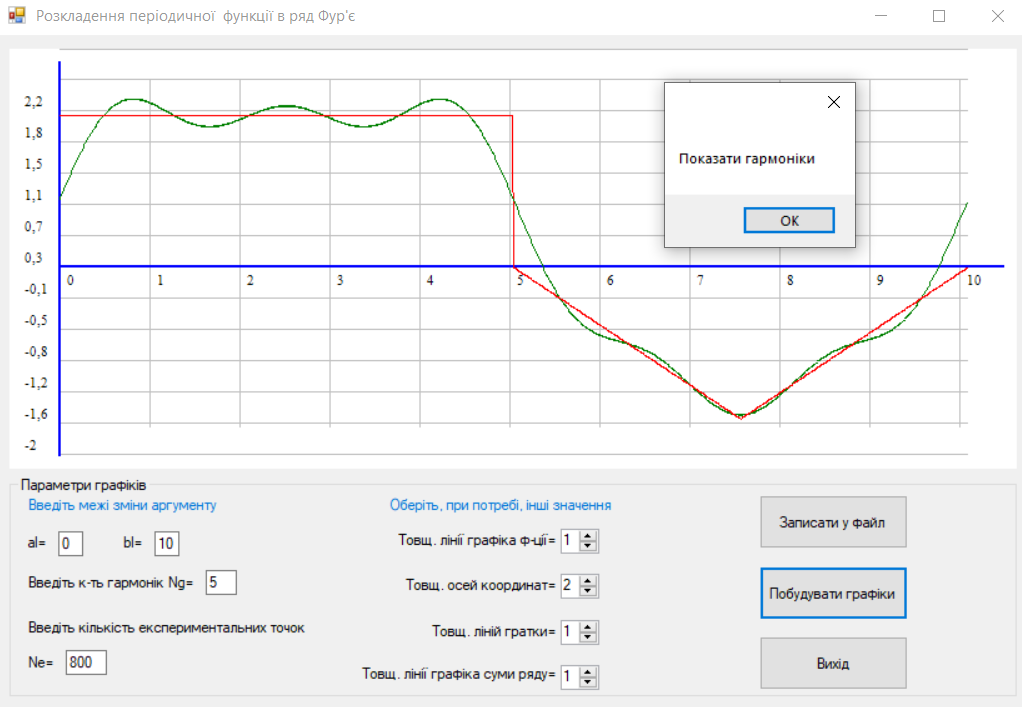
private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

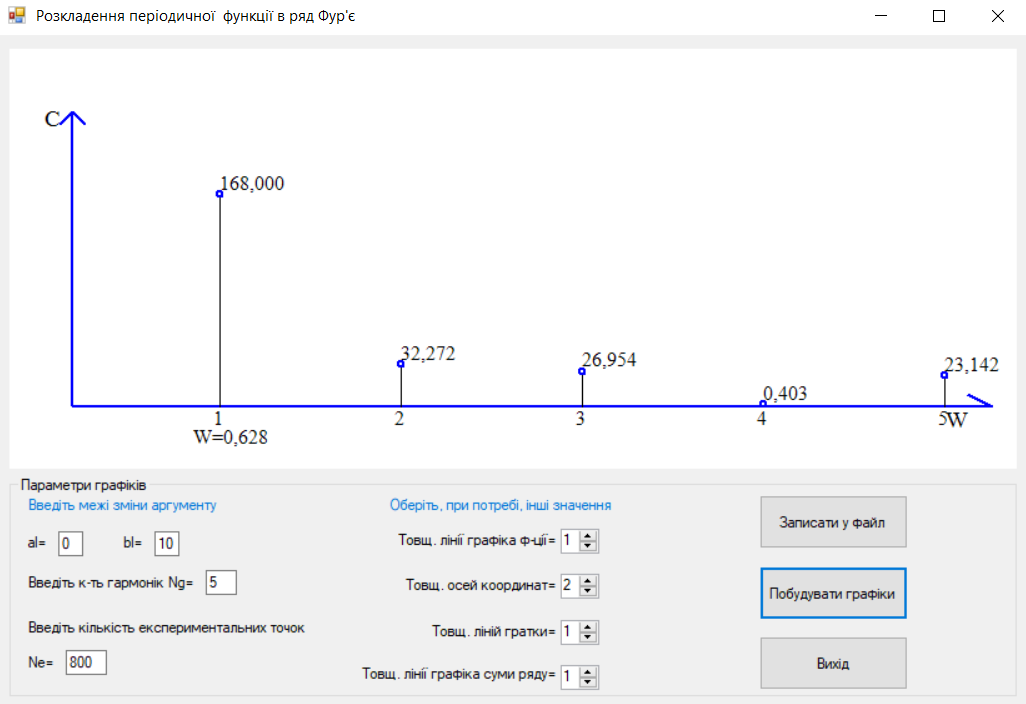
Close();

}

1. **Результати виконання програми**







**Висновки:**

Програма працює справно та без багів. Я познайомився із принципами використання структур і файлів у VС++.