

Лабораторна робота 8

Тема: Робота з бінарними файлами. Використання DataGridView при роботі з даними файлів

Мета роботи: набути практичних навиків програмного створення та редагування бінарних файлів засобами Visual C++, закріпити практичні навики програмного створення та редагування бінарних файлів за допомогою елемента dataGridView.

Хід роботи

Завдання до роботи

1. Розглянути наведені приклади

Приклад 1. Створити програму, яка дозволить зберігати у бінарному файлі двійкові дані: номер мобільного телефону, дату підключення (активації) номера, вартість однієї хвилини розмов, назvu тарифу, назvu мобільного оператора. Крім програмного створення файла і заповнення його даними, передбачити програмний перегляд даних зі створеного файла.

Відібрати абонентів, які користуються мобільним зв'язком понад 10 років без зміни номера телефону.

Лістинг 8.1 – Код програми

```
using namespace System;
using namespace System::IO;
void Zapis(String^ fname) {
    BinaryWriter^ fb = gcnew BinaryWriter(File::Open(fname,
    FileMode::OpenOrCreate));
    fb->Seek(0, SeekOrigin::End);
    try
    {
        String ^номер = nullptr, ^тариф = nullptr, ^оператор = nullptr;
        double плата;
        DateTime d;
        do
        {
Console::Write("\nВведіть номер телефона (Enter для завершення): ");
```

```

номер = Console::ReadLine(); if (номер != String::Empty) {
Console::Write("Введіть дату підключення номера (dd.MM.yyyy): ");
d = Convert::.ToDateTime(Console::ReadLine()); Console::Write("Введіть
вартість 1 хв розмов: "); плата = Convert::ToDouble(Console::ReadLine())
Console::Write("Введіть назву тарифу: "); тариф = Console::ReadLine();
Console::Write("Введіть оператора: "); оператор = Console::ReadLine();
fb->Write(номер); fb->Write(d.Ticks); fb->Write(плата); fb-
>Write(тариф); fb->Write(оператор);
}
} while (номер != String::Empty);
}
finally
{
if (fb != nullptr)
fb->Close(); }
}

void Prosmotr(String^ fname)
{
if (!File::Exists(fname)) {
Console::WriteLine("Файл не знайдено!");
return; }
BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(fname));
try
{
Console::WriteLine("\nПерегляд файла: ");
Console::WriteLine("Номер\tДата підкл.\tВартість
(грн)\tТариф\tОператор");
while (fb->BaseStream->Position < fb->BaseStream->Length)
{
String^ номер = fb->ReadString(); DateTime d = DateTime(fb-
>ReadInt64()); double плата = fb->ReadDouble(); String^ тариф = fb-
>ReadString();
String^ оператор = fb->ReadString(); Console::WriteLine(номер + "\t" +
d.ToString("d") + "\t" +
String::Format("{0:0.00}", плата) + "\t\t" +
тариф + "\t" + оператор);
} }
finally
{
fb->Close(); }
}

void ThisYear(String^ fname) {
if (!File::Exists(fname)) {
Console::WriteLine("Файл не знайдено!"); return;
}
BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(fname));
try
{

```

```

Console::WriteLine(" \nАбоненти, піключені понад 10 років тому:");
while (fb->BaseStream->Position < fb->BaseStream->Length)
{
    String^ номер = fb->ReadString(); DateTime d = DateTime(fb-
>ReadInt64()); double плата = fb->ReadDouble(); String^ тариф = fb-
>ReadString(); String^ оператор = fb->ReadString(); if (d.Year + 10 <=
DateTime::Now.Year) {
    Console::WriteLine(номер + "\t" + d.ToString("d") + "\t" +
String::Format("{0:0.00}", плата) + "\t\t" +
тариф + "\t" + оператор);
} }
}
finally
{
    fb->Close(); }
}

int main(array<System::String^> ^args) {
String^ fname = "telephons.dat";
Console::WriteLine(L"Введіть дані абонентів. Для завершення вводу номера
двічі натисніть Enter.");
Zapis(fname); Prosmotr(fname); ThisYear(fname); Console::ReadLine();
return 0;
}

```

Введіть дані абонентів. Для завершення вводу номера дзвічі натисніть Enter.

Введіть номер телефона (Enter для завершення): 068251257
 Введіть дату підключення номера (dd.MM.yyyy): 07.03.2020
 Введіть вартість 1 хв розмов: 25
 Введіть назву тарифу: Окуп
 Введіть оператора: Life

Введіть номер телефона (Enter для завершення):

Перегляд файла:

Номер	Дата підкл.	Вартість (грн)	Тариф	Оператор
0988124061	12.09.2010	10,00	Vodafone	Vodafone
068251257	07.03.2020	25,00	Окуп	Life

Абоненти, піключені понад 10 років тому:

Номер	Дата підкл.	Вартість (грн)	Оператор
0988124061	12.09.2010	10,00	Vodafone

Рисунок 8.1 – Вікно результатів програми

Приклад 2.

Створити програму, яка дозволить зберігати у бінарному файлі двійкові дані: номер мобільного телефону, назву тарифу, назву мобільного оператора, наявність абонплати, вартість одної хвилини розмов.

Крім програмного створення файла і заповнення його даними, передбачити програмний перегляд даних зі створеного файла. Відібрати дані про номери телефонів на Київстарі та обчислити їхню кількість і середню вартість 1 хв. розмов. Відібрати тарифи без абонплати. Визначити найдешевший тариф без абонплати.

Лістинг 8.2 – Код програми

```
#pragma endregion
private: System::Void MyForm_Load(System::Object^ sender,
System::EventArgs^ e)
{
fname = "connect.dat";
dataGridView2->Visible = false; }
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
BinaryWriter^ fb = gcnew BinaryWriter(File::Open(fname,
FileMode::OpenOrCreate));
fb->Seek(0, SeekOrigin::End);
try
{
int k = dataGridView1->Rows-
>GetRowCount(DataGridViewElementStates::Visible);
for (int i = 0; i < k - 1; i++)
{
if (dataGridView1[0, i]->Value == nullptr || dataGridView1[1, i]->Value
== nullptr || dataGridView1[2, i]->Value == nullptr || dataGridView1[4,
i]->Value == nullptr)
continue;
String ^номер = Convert::ToString(dataGridView1[0, i]->Value);
String ^тариф = Convert::ToString(dataGridView1[1, i]->Value);
String ^оператор = Convert::ToString(dataGridView1[2, i]->Value);
bool абонплата = Convert::.ToBoolean(dataGridView1[3, i]->Value);
double vart = Convert::ToDouble(dataGridView1[4, i]->Value);
fb->Write(номер); fb->Write(тариф); fb->Write(оператор); fb-
>Write(абонплата); fb->Write(vart);
}
dataGridView1->Rows->Clear();
MessageBox::Show("Дані успішно записано у файл " + fname, "Запис
завершено");
}
catch (Exception^ ex)
{
MessageBox::Show("Помилка при записі: " + ex->Message, "Помилка!"); }
finally
```

```
{  
if (fb != nullptr) fb->Close();  
} }  
private: System::Void button2_Click(System::Object^ sender,  
System::EventArgs^ e)  
{  
dataGridView1->Rows->Clear(); dataGridView2->Visible = false;  
if (!File::Exists(fname)) {  
MessageBox::Show("Файл " + fname + " не знайдено.");  
return; }  
BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(fname));  
try  
{  
while (fb->BaseStream->Position < fb->BaseStream- >Length)  
{  
String^ номер = fb->ReadString(); String^ тариф = fb->ReadString();  
String^ оператор = fb->ReadString(); bool абонплата = fb->ReadBoolean();  
double vart = fb->ReadDouble();  
dataGridView1->Rows->Add(номер, тариф, оператор, абонплата,  
String::Format("{0:0.00}", vart)); }  
}  
finally  
{  
if (fb != nullptr) fb->Close();  
} }  
private: System::Void button3_Click(System::Object^ sender,  
System::EventArgs^ e)  
{  
int k = 0; double s = 0; dataGridView2->Visible = true; dataGridView2-  
>Rows->Clear();  
if (!File::Exists(fname)) return;  
BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(fname)); try  
{  
while (fb->BaseStream->Position < fb->BaseStream- >Length)  
{  
String^ номер = fb->ReadString(); String^ тариф = fb->ReadString();  
String^ оператор = fb->ReadString(); bool абонплата = fb->ReadBoolean();  
double vart = fb->ReadDouble();  
if (оператор == "Київстар")  
{  
dataGridView2->Rows->Add(номер, тариф, оператор, абонплата,  
String::Format("{0:0.00}", vart)); s += vart;  
k++; }  
}  
finally  
{  
if (k > 0) s /= k;  
MessageBox::Show("Всього абонентів на Київстарі: " + (k).ToString() +
```

```
"\nСередня вартість 1 хв розмов: " + String::Format("{0:0.00}", s) + "
грн.",  
"Результат відбору"); if (fb != nullptr) fb->Close();  
} }  
private: System::Void button4_Click(System::Object^ sender,  
System::EventArgs^ e)  
{  
    dataGridView2->Visible = true; dataGridView2->Rows->Clear();  
    if (!File::Exists(fname)) return;  
    BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(fname)); try  
{  
    while (fb->BaseStream->Position < fb->BaseStream- >Length)  
{  
        String^ номер = fb->ReadString(); String^ тариф = fb->ReadString();  
        String^ оператор = fb->ReadString(); bool абонплата = fb->ReadBoolean();  
        double vart = fb->ReadDouble(); if (!абонплата)  
            dataGridView2->Rows->Add(номер, тариф, оператор, абонплата,  
String::Format("{0:0.00}", vart)); }  
    MessageBox::Show("Виведено тарифи без абонплати у нижню таблицю.",  
"Відбір завершено");  
}  
finally  
{  
    if (fb != nullptr) fb->Close();  
}  
}  
private: System::Void button5_Click(System::Object^ sender,  
System::EventArgs^ e)  
{  
    dataGridView2->Visible = true; dataGridView2->Rows->Clear();  
    if (!File::Exists(fname)) return;  
    BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(fname)); try  
{  
    double vart, min = 10000.0;  
    String ^nommin = "", ^tarmin = "", ^opmin = "";  
    while (fb->BaseStream->Position < fb->BaseStream- >Length)  
{  
        String^ номер = fb->ReadString(); String^ тариф = fb->ReadString();  
        String^ оператор = fb->ReadString(); bool абонплата = fb->ReadBoolean();  
        vart = fb->ReadDouble(); if (!абонплата && vart < min)  
{  
            min = vart; nommin = номер; tarmin = тариф; opmin = оператор;  
        } }  
        if (nommin != "")  
{  
            dataGridView2->Rows->Add(nommin, tarmin, opmin, false,  
String::Format("{0:0.00}", min));  
        MessageBox::Show("Найдешевший тариф без абонплати: " + tarmin + " (" +  
String::Format("{0:0.00}", min) + " грн/хв)", "Результат пошуку");  
    }
```

```

}
else
{
MessageBox::Show("Не знайдено тарифів без абонплати.", "Результат
пошуку");
} }
finally
{
if (fb != nullptr) fb->Close();
} }
};

```

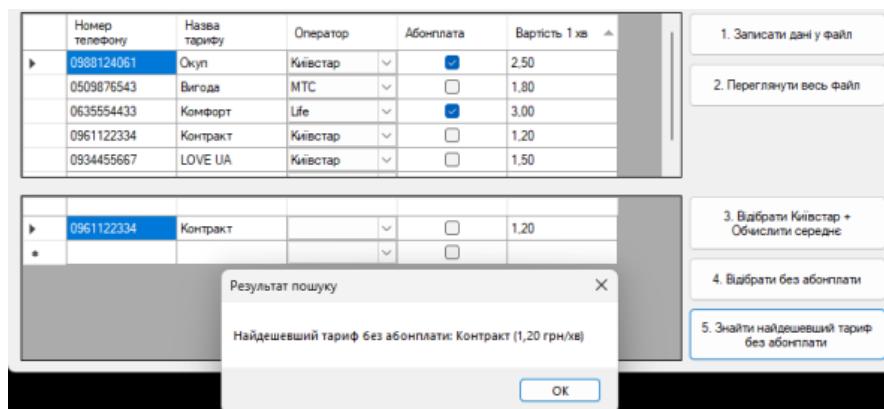


Рисунок 8.2 – Вікно результатів програми

Створити програму, яка дозволить зберігати у бінарному файлі двійкові дані для реалізації завдання згідно таблиці 8.1.

У звіті лабораторної роботи навести програмний код для створення бінарного файла з ім'ям Вашого прізвища і розширенням dat, заповнення файла даними, переглядання вмісту сформованого файла та виконання індивідуального завдання.

5	Список робітників підприємства: табельний номер, ПІБ, посада, стать (ч або ж), адреса	Вивести відомості про середній вік чоловіків та жінок, а також скласти список жінок, яким залишився рік до пенсії
---	---	--

Лістинг 8.3 – Код програми

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <fstream>
#include <vector>
#include <string>

```

```
#include <iostream>
#include <ctime>
#include <iomanip>
#include <locale>

using namespace std;

struct Worker {
    int id;
    char pib[50];
    char position[50];
    char gender;
    char address[150];
    int birth_year;
};

const string FILENAME = "Chavurska.dat";

void createTestData() {
    ofstream outFile(FILENAME, ios::binary | ios::trunc);

    if (!outFile.is_open()) {
        cerr << "Помилка відкриття файлу для запису!" << endl;
        return;
    }

    cout << "Створення тестових даних у файлі " << FILENAME << endl;

    vector<Worker> test_data = {
        {101, "Петрова О. В.", "Менеджер", 'Ж', "пр. Перемоги, 12",
1971},
        {202, "Іванов А. П.", "Програміст", 'Ч', "вул. Сонячна, 5",
1990},
        {303, "Коваленко В. С.", "Керівник", 'Ч', "пр. Науки, 8", 1975},
        {404, "Мельник І. Р.", "Директор", 'Ч', "вул. Центральна, 1",
1960},
        {505, "Сидорова С. В.", "Бухгалтер", 'Ж', "вул. Миру, 22",
1985},
        {606, "Антомириська Ю. С.", "Аналітик", 'Ж', "пр. Свободи, 1",
1998}
    };

    for (const auto& w : test_data) {
        outFile.write(reinterpret_cast<const char*>(&w),
sizeof(Worker));
    }

    outFile.close();
}
```

```
        cout << "У файл успішно записано " << test_data.size() << "
записів." << endl;
}

void readAndDisplay() {
    ifstream inFile(FILENAME, ios::binary | ios::in);

    if (!inFile.is_open()) {
        cerr << "Помилка: файл " << FILENAME << " не знайдено!" << endl;
        return;
    }

    cout << "\n--- Перегляд вмісту файлу " << FILENAME << " ---" <<
endl;
    cout << setw(8) << left << "Таб. №" << setw(30) << "ПІБ" << setw(20)
<< "Посада"
        << setw(8) << "Стать" << "Рік Народження" << endl;
    cout << "-----"
        << endl;

    Worker w;
    while (inFile.read(reinterpret_cast<char*>(&w), sizeof(Worker))) {
        cout << setw(8) << left << w.id
            << setw(30) << w.pib
            << setw(20) << w.position
            << setw(8) << w.gender
            << w.birth_year << endl;
    }

    inFile.close();
}

void processIndividualTasks() {
    ifstream inFile(FILENAME, ios::binary | ios::in);

    if (!inFile.is_open()) {
        cerr << "Помилка: файл " << FILENAME << " не знайдено для
обробки!" << endl;
        return;
    }

    time_t now = time(nullptr);
    tm* ltm = localtime(&now);
    int current_year = 1900 + ltm->tm_year;

    cout << "\n--- Результат Індивідуального Завдання (Поточний рік: "
<< current_year << ") ---" << endl;
```

```

cout << "\nЗавдання 2: Єдине жіноче ім'я, яке залишилося 1 рік до
пенсії (55 років)" << endl;
cout << setw(8) << left << "Таб. №" << setw(30) << "ПІБ" << setw(20)
<< "Рік Народження" << endl;
cout << "-----"
<< endl;

bool found_pensioner = false;
int male_age_sum = 0;
int female_age_sum = 0;
int male_count = 0;
int female_count = 0;

Worker w;
while (inFile.read(reinterpret_cast<char*>(&w), sizeof(Worker))) {
    int age = current_year - w.birth_year;

    if (w.gender == 'Ч' || w.gender == 'ч') {
        male_age_sum += age;
        male_count++;
    }
    else if (w.gender == 'Ж' || w.gender == 'ж') {
        female_age_sum += age;
        female_count++;

        // Жінки йдуть на пенсію у 55 років (якщо 1 рік до пенсії,
        to вік 54)
        if (age == 54) {
            cout << setw(8) << left << w.id
                << setw(30) << w.pib
                << w.birth_year << endl;
            found_pensioner = true;
        }
    }
}

if (!found_pensioner) {
    cout << "(У поточному наборі даних таких жінок немає)" << endl;
}

inFile.close();

// Завдання 3: Обчислення та виведення середнього віку
cout << "\nЗавдання 3: Обчислення та виведення середнього віку" <<
endl;

```

```
        double avg_male_age = (male_count > 0) ? (double)male_age_sum /  
male_count : 0.0;  
        double avg_female_age = (female_count > 0) ? (double)female_age_sum  
/ female_count : 0.0;  
  
    cout << fixed << setprecision(2);  
  
    cout << "Середній вік чоловіків: " << avg_male_age << " років" <<  
endl;  
    cout << "Середній вік жінок: " << avg_female_age << " років" <<  
endl;  
}  
  
int main() {  
    setlocale(LC_ALL, "ukr");  
  
    createTestData();  
    readAndDisplay();  
    processIndividualTasks();  
  
    cout << "\nНатисніть Enter для виходу...";  
    cin.ignore();  
    cin.get();  
    return 0;  
}
```

Створення тестових даних у файл Chavurska.dat				
У файл успішно записано 6 записів.				
--- Перегляд вмсту файлу Chavurska.dat ---				
Таб. №	П?Б	Посада	Стать	Р?к Народження
101	Петрова О. В.	Менеджер	Ж	1971
202	?ванов А. П.	Програм?ст	Ч	1990
303	Коваленко В. С.	Кер?вник	Ч	1975
404	Мельник ?. Р.	Директор	Ч	1960
505	Сидорова С. В.	Бухгалтер	Ж	1985
606	Антомириська Ю. С.	Анал?тик	Ж	1998

--- Результат ?ндив?дуального Завдання (Поточний р?к: 2025) ---

Завдання 2: Єдине ж?ноче ?м'я, яке залишилося 1 р?к до пенс?ї (55 рок?в)

Таб. №	П?Б	Р?к Народження
101	Петрова О. В.	1971

Завдання 3: Обчислення та виведення середнього в?ку
 Середн?й в?к чолов?ків: 50.00 рок?в
 Середн?й в?к ж?нок: 40.33 рок?в

Натисн?ть Enter для виходу...|

Рисунок 8.3 – Результат виконання завдання

Висновок: На цій лабораторній роботі я набула практичних навиків програмного створення та редагування бінарних файлів засобами Visual C++, закріпiti практичнi навики програмного створення та редагування бінарних файлів за допомогою елемента dataGridView.