МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Одеська національна академія зв'язку ім. О. С. Попова

Кафедра інформаційних технологій

Створення багатомодульних програмних проектів для опрацювання даних у файлах засобами Visual C++

Методичні вказівки для виконання курсової роботи
з дисципліни
"Комп'ютерні технології та програмування"

Одеса 2016

УДК 004.43 План НМВ 2016 р.

Укладачі: **Трофименко О. Г., Прокоп Ю. В., Буката Л.М.**

Рецензент: к.т.н., доцент Флейта Ю. В.

Надано дидактичний матеріал для виконання курсової роботи з дисципліни "Комп’ютерні технології та програмування", а також 30 варіантів завдань, які дозволять всебічно засвоїти роботу зі створення багатомодульних проектів засобами Visual C++ та закріпити навики з програмного опрацювання даних, які зберігаються у різнотипних файлах. Для самостійного виконання курсової роботи розглянуто послідовність виконання цієї роботи на комп'ютері. Має значну кількість ілюстрацій і докладні описи функцій виконання конкретних завдань. Надано теоретичні відомості з докладним розглядом матеріалу.

Методичні вказівки будуть корисними студентам спеціальності "Автоматизація та комп’ютерно-інтегровані технології", які вивчають дисципліну "Комп’ютерні технології та програмування" для набуття навиків програмування з метою подальшого застосовування цих навиків у власній повсякденній і майбутній професійній діяльності; також стануть у нагоді користувачам персональних комп’ютерів, які бажають поглибити свої знання і навики з програмування засобами Visual C++.

СХВАЛЕНО
на засіданні кафедри
інформаційних технологій
і рекомендовано до друку.

Протокол № 4 від 14.12.2015 р.

ЗАТВЕРДЖЕНО
методичною радою академії зв’язку.

Протокол № 6 від 23.02.2016 р.

ЗМІСТ

ВСТУП

Популярне інтегроване середовище розробки програмного забезпечення Visual C++ є частиною комплекту Microsoft Visual Studio, а, крім того, воно постачається у вигляді безкоштовного функціонально обмеженого комплекту Visual C++ Express Edition. Стандартизована мова програмування Visual C++ дозволяє розробляти як консольні програми, так і програми з графічним інтерфейсом, у тому числі, з підтримкою технології Windows Forms як в "рідному", так і в керованому кодах для різних комп'ютерних систем. Поєднання простоти освоєння візуального середовища проектування і підтримка широкого спектра технологій роблять Visual C++ універсальним інструментом створювання програмних проектів будь-якого рівня складності як для платформи .NET Framework, так і для виконання у середовищі "чистої" Windows.

У цій роботі ми намагалися на конкретних прикладах доторкнутися до великих можливостей середовища розробки програмних проектів Visual C++ для створення багатомодульних проектів з опрацювання даних, які зберігаються у файлах. Тематика роботи зумовлена тим, що більшість комп'ютерних програм, які розв'язують ті чи інші практичні завдання, зберігають дані у вигляді різнотипних файлів, і тому виникає необхідність програмно створювати, видаляти, записувати читати, відкривати файли. Visual C++ пропонує широкий спектр засобів зі створення бінарних (двійкових) та текстових файлів. На практиці доцільним є створення основного файла (чи файлів) з даними у двійковому форматі, оскільки цей формат не потребує додаткового місця, а, отже, і пам'яті для символів-розділювачів (пробілів, табуляцій та ін.). А вже формування звітних документів слід створювати у текстовому форматі (типи файлів txt, rtf, doc тощо).

Актуальність теми курсової роботи "Створення багатомодульних програмних проектів для опрацювання даних у файлах засобами Visual C++" зумовлена повсюдною практичною затребуваністю програмних проектів, які зберігають дані у файлах. Тобто програма сама має створювати файли того чи іншого формату, записувати у них дані, на вимогу виводити або всі дані файла, або лише ті, які задовольняють певній умові, вміти редагувати і видаляти певні дані файла, надавати можливість створення нових файлів (звітних документів) з певними даними з основного файла та ін.

Об'єктом дослідження є технології Visual Studio зі створення багатомодульних проектів та технології опрацювання даних, які зберігаються у файлах.

Предмет дослідження – програмні засоби Visual C++ для створення файлів різних форматів, записування у них даних, виведення вмісту цих файлів, відбирання даних за умовою, можливості редагування файла і видалення певних даних з файла, створення нових файлів (звітних документів) з певними даними з основного файла, створення програмних проектів з декількома динамічними формами та взаємодії між ними.

Мета курсової роботи полягає у вивченні технологій Visual C++ програмного створення файлів даних з можливістю їх програмного редагування (до-

писування, змінювання чи видалення даних) та набуття навиків розробки програмних проектів з візуальним інтерфейсом, організованого за допомогою меню, діалогових вікон, декількох взаємодіючих форм та ін.

Відповідно до поставленої мети визначено такі завдання:

– розкрити сутність технологій Visual Studio зі створення багатомодульних проектів;

– вивчити наявні програмні засоби Visual C++, які дозволяють створювати файли різних форматів, записувати у них дані, переглядати вміст цих файлів, відбирати дані за умовою, редагувати дані у файлах;

– розробити програмний проект для розв’язання завдання відповідно до індивідуального варіанта, який передбачає створення, заповнення даними, відбір та редагування файлів різних форматів.

Курсова робота є самостійною роботою студента, яка передбачає закріплення та розширення теоретичних знань з дисципліни "Комп’ютерні технології та програмування".

ЗАВДАННЯ

Використовуючи середовище програмування Visual C++, створити програмний проект, який реалізує такі завдання:

1) програмне створення бінарного (двійкового) файла та заповнення його даними, структуру яких вибрати з табл. 1 (стовпець *Вміст бінарного файла*) згідно з індивідуальним варіантом;

2) програмний перегляд даних створеного файла;

3) можливість редагування та зберігання відредагованого файла;

4) впорядкування (сортування) даних згідно з параметром, заданим у табл. 1 (стовпець *Параметри сортування*);

5) відбір даних файла за певною умовою, заданою у табл. 1 (стовпець *Відбір даних за умовою*);

6) відбір з бінарного файла даних за умовою, яку вибрати з табл. 1 (стовпець *Умова на відбір даних з формуванням документа (текстового файла)*), та створення на основі відібраних даних текстового звітного документа;

7) створення трьох форм: 1) про автора, 2) про завдання та 3) формузаставку. Крім того, розробити програмний код для виведення, редагування та форматування параметрів шрифту текстового документа із завданням, виведення даних про розмір файла даних, дату його створення та час останнього редагування тощо. Доступ до цих функцій організувати за допомогою меню;

8) можливість видалення даних з бінарного файла даних за умовою, яку вибрати з табл. 1 (стовпець *Умова на видалення даних*).

Для виконання завдання пропонується три рівня складності: базовий, підвищений та високий. Завдання для базового і підвищеного рівнів складності наведені у табл. 1. Для базового рівня достатньо виконати п. 1 … 6, а для підвищеного – ще й п. 7 ... 8. Високий рівень складності передбачає використання нових для Вас (не розглянутих у базовому курсі навчання) програмних інструментів і технологій (узгодьте завдання цього рівня з викладачем).

Оформіть і здайте викладачеві виконану курсову роботу (у рукописному або друкованому вигляді).

Таблиця 1 – Індивідуальні варіанти завдань

№ вар.	Вміст бінарного файла	Параметри сортування	Відбір даних за умовою	Умова на відбір даних з формуванням документа (текстового файла)	Умова на видалення даних
1	Заявки на ремонт комп'ютерної техніки: найменування ПК, назва ремонту (список), ПІБ власника, дата здачі, кількість днів для ремонтних робіт, вартість ремонту, наявність передплати (так/ні)	За спаданням вартості	Передплачені заявки, для яких кількість днів ремонту є меншим 7-ми	Заявки на ремонт комп'ютерної техніки, з дати здачі на ремонт яких минуло понад 20 днів	Заявки з вартістю ремонтних робіт до 50 грн.
2	Дані про успішність студентів: прізвище та ініціали, дата народження, назва групи (список), оцінки з фізики, математики, історії, наявність стипендії в минулому семестрі (так/ні)	За зростанням оцінок з математики	Студентів, які не здали сесію (мають "двійку" хоча б з одного предмета) та мали стипендію в минулому семестрі	Студенти, яким виповнилось 18 років	Студенти, середній бал яких менше 60-ти
3	Абоненти телефонної станції: ПІБ абонента, номер телефону, дата встановлення телефону, розмір фіксованої абонплати в місяць (список), наявність пільг (так/ні), сума заборгованості	За спаданням заборгованості	Абонентів без пільг і боргом понад 200 грн.	Абоненти, телефони яких встановлені понад три роки тому	Абоненти, борг яких перевищив 500 грн.
4	Прайс-лист ноутбуків у магазині: фірма-виробник, тип (список: ноутбук, ультрабук, нетбук, трансформер тощо), діагональ дисплею у дюймах, ціна, дата поставки, наявність сенсорного дисплея (так/ні)	За зростанням ціни	Ноутбуки типу "ультрабук" із сенсорним дисплеєм	Ноутбуки, з дати поставки яких минуло понад чотири місяці	Ноутбуки, ціна яких менше 2500 грн.

Продовження табл. 1

№ вар.	Вміст бінарного файла	Параметри сортування	Відбір даних за умовою	Умова на відбір даних з формуванням документа (текстового файла)	Умова на видалення даних
5	Товари у магазині електроніки: код, назва, фірма-виробник (список), рік випуску, вартість, наявність гарантії (так/ні)	За спаданням вартості	Дані про телевізори фірми "Samsung" з наявною гарантією	Товари, випущені понад два роки тому з ціною понад 1000 грн.	Товари, випущені п’ять років тому
6	Список студентів: номер, ПІБ, група (список), дата народження, рейтинг з історії, математики, основ програмування, наявність пропусків занять без поважної причини (так/ні)	В алфавітному порядку прізвищ	Студенти, які претендують на стипендію: мають середній бал понад 75 балів та не мають пропусків занять без поважної причини (так/ні)	Студенти, які святкують день народження у цьому місяці	Студенти, які мають пропуски занять без поважної причини
7	Прокат автомобілів: державний номер, марка (список), кілометраж пробігу, дата здачі у прокат, сума платні, наявність сигналізації (так/ні)	За спаданням кілометражу	Автомобілі "Ford" без сигналізації	Автомобілі, здані в прокат понад тиждень тому	Автомобілі, кілометраж пробігу яких понад 100 000 км
8	Список робітників підприємства: табельний номер, ПІБ, посада (список), дата народження, наявність вищої освіти (так/ні), розмір зарплатні	За зростанням розміру зарплатні	Співробітників без вищої освіти, вік яких до 40 років	Робітники підприємства, які святкують у цьому році ювілей (вік кратний 5-ти)	Робітники підприємства на посаді "оператор"
9	Дані в ЖЕКу про мешканців: адреса, назва ділянки чи району (список), ПІБ власника, дата укладення договору про надання послуг, наявність субсидії (так/ні), розмір заборгованості	За спаданням розміру заборгованості	Боржників без субсидії, сума боргу яких понад 500 грн.	Мешканці, з якими договори укладені понад три роки тому	Мешканці, з якими договори укладені у листопаді-місяці

Продовження табл. 1

№ вар.	Вміст бінарного файла	Параметри сортування	Відбір даних за умовою	Умова на відбір даних з формуванням документа (текстового файла)	Умова на видалення даних
10	Список послуг операторів мобільного телефонного зв'язку: номер телефону, назва тарифу (список), дата ініціалізації пакета, наявність підключення до Інтернету (так/ні), баланс рахунку (грн.)	За спаданням номерів телефонів	Дані про абонентів без підключення до Інтернету і балансом рахунку понад 100 грн.	Дані про абонентів, телефони яких ініціалізовані понад рік тому	Дані про абонентів, баланс рахунку яких менше 1 грн.
11	Список аварійних будинків у районі міста: вулиця (список), номер будинку, кількість мешканців у будинку, рік побудови, дата постановки на облік, наявність проведення ремонту за останні 20 років (так/ні)	За зростанням року побудови	Вивести відомості про будинки без ремонту за останні 20 років, побудовані до 1941 року, в яких понад 50 мешканців	Аварійні будинки, постановлені на облік понад 10 років тому	Аварійні будинки на вулиці Пішонівська
12	Прайс-лист планшетів у магазині: фірма-виробник (список), модель, діагональ екрана в дюймах, ціна, дата поставки, наявність 3G-модуля (так/ні)	За спаданням значень розміру діагоналі екрана	Планшети Apple з 3G-модулем	Планшети з діагоналлю 7″, поставлені в цьому місяці	Планшети з ціною менше 100 грн.
13	Список підприємств, які виготовляють електронне обладнання: найменування підприємства, дата реєстрації підприємства, вид продукції (список), кількість продукції за останній квартал, рентабельність (так/ні)	За зростанням кількості продукції за останній квартал	Рентабельні підприємства, які виробляють TV-тюнери	Підприємства, зареєстровані понад 6 років тому	Підприємства, зареєстровані за останні півроку

Продовження табл. 1

№ вар.	Вміст бінарного файла	Параметри сортування	Відбір даних за умовою	Умова на відбір даних з формуванням документа (текстового файла)	Умова на видалення даних
14	Список робітників: табельний номер, прізвище та ініціали, дата народження, посада (список), оклад, наявність нагород (так/ні)	За спаданням окладу	Робітники пенсійного віку (понад 60 років), які мають нагороди	Робітники, оклади яких більше середньо-арифметичного всіх робітників	Робітники, день народження яких святкують у найближчі 20 днів
15	Список книг у бібліотеці: інвентарний номер, назва книги, автор, жанр (список), рік видання, ціна, дата інвентаризації, наявність CD/DVD для книги (так/ні)	За зростанням року видання	Книги, у назві яких є "C++", які в комплекті мають CD чи DVD	Книги, інвентаризовані в поточному календарному році	Книги, видані до 1950 року
16	Прайс-лист магазину мобільних телефонів: фірма-виробник (список), модель, діагональ екрана в дюймах, ціна, дата поставки, наявність двох SIM-карт (так/ні)	За спаданням ціни	Телефони Samsung з двома SIM-картами	Телефони, з дати поставки пройшло менше місяця	Телефони з ціною менше 400 грн.
17	Послуги турагенції: назва туру (список), країна, дата виїзду, кількість днів, вартість туру, наявність нічних переїздів (так/ні)	За зростанням вартості турів	Тури до Польщі без нічних переїздів	Тури, до дати виїзду яких лишилось менше двох тижнів	Тури, кількість днів яких понад 18 днів
18	Список комп'ютерів у комп'ютерному класі: номер, тип ПК (список), тактова частота, оперативна пам'ять, ємність диска, дата встановлення, наявність підключення до мережі Інтернет (так/ні)	За спаданням значень тактової частоти	Підключені до Інтернету ПК, в яких ємність диска не менше 80 Гб	Комп'ютери, встановлені у комп'ютерному класі понад чотири роки тому	Комп'ютери з розміром оперативної пам'яті до 512 Мб

Продовження табл. 1

№ вар.	Вміст бінарного файла	Параметри сортування	Відбір даних за умовою	Умова на відбір даних з формуванням документа (текстового файла)	Умова на видалення даних
19	Відомості про наявність квитків на автостанції: пункт призначення, модель автобуса (список), дата відправлення, час відправлення, ціна, наявність вільних місць (так/ні)	В алфавітному порядку назв пунктів призначення	Дані про рейси до Києва, які відправляються у першій половині дня	Автобусні рейси до Вільнюса, які відправляються у найближчі 10 днів	Автобусні рейси, які вже в минулому
20	Каталог ПК: фірма-виробник, тип (список: настільний, моноблок, нетоп, ігровий), ємність RAM, ємність диска, наявність ТВ-тюнера (так/ні)	За зростанням ємності диска	Настільні ПК з ТВ-тюнером	Неттопи фірм Acer або Lenovo	ПК з ємністю RAM до 1 Гб
21	Список автомобілів в автосалоні: марка (список), модель, рік випуску, дата реєстрації, пробіг, ціна, наявність гарантійного сервісу протягом року (так/ні)	За спаданням ціни	Автомобілі марки "Mazda" з гарантійним сервісом	Автомобілі, зареєстровані минулого місяця	Автомобілі з пробігом до 5 000 км
22	Прайс-лист WIFI-адаптерів у магазині: фірма-виробник (список), модель, швидкість WI-FI у Мбіт/с, ціна, дата поставки, наявність рознімів USB (так/ні)	За спаданням швидкості	WIFI-адаптери фірми Asus з наявністю рознімів USB	WIFI-адаптери, дата поставки яких відбулася у минулому місяці	WIFI-адаптери зі швидкістю до 54 Мбіт/с
23	Анкетні дані про співробітників: табельний номер, ПІБ, посада (список), дата прийому на роботу, сімейний стан, наявність неповнолітніх дітей	В алфавітному порядку ПІБ	Співробітники, які працюють на підприємстві понад 10 років	Сімейні співробітники, які мають неповнолітніх дітей	Співробітники, які зараховані на роботу у цьому році

Продовження табл. 1

№ вар.	Вміст бінарного файла	Параметри сортування	Відбір даних за умовою	Умова на відбір даних з формуванням документа (текстового файла)	Умова на видалення даних
24	Список книг у книжковому магазині: назва книги, автор, категорія (список), дата надходження, ціна, наявність акційної знижки (так/ні)	За спаданням ціни	Книги категорії "Програмування" з акційною знижкою	Книги, які надійшли за минулий календарний рік	Книги з категорії "Фантастика"
25	Список команди "Чорноморець": номер гравця, ПІБ, дата народження, амплуа (список: воротар, захисник тощо), кількість зіграних ігор за минулий сезон, наявність важких травм у цьому сезоні (так/ні)	За спаданням кількості зіграних ігор за минулий сезон	Дані про нападників команди, які не мали важких травм у цьому сезоні	Гравці, старші 30 років	Іменинників наступного місяця
26	Список принтерів для продажу: тип принтера, фірма-виробник (список), марка, ціна принтера, дата поставки, наявність у комплекті запасного картриджа (так/ні)	За зростанням ціни	Принтери фірми HP за ціною до 2000 грн., які мають запасний картридж	Принтери, дата поставки яких відбулася у цьому календарому місяці	Принтери з типом "матричний"
27	Список товарів на складі: код, назва, категорія (список), кількість, ціна за одиницю товару, дата завезення на склад, ознака того чи треба товар зберігати в холодильнику (так/ні)	В алфавітному порядку назв товарів	Товари, які зберігаються в холодильнику, завезені понад місяць тому	Товари з ціною менше 70 грн., кількість яких на складі перевищує 1000	Товари завезені у лютому-місяці

Закінчення табл. 1

№ вар.	Вміст бінарного файла	Параметри сортування	Відбір даних за умовою	Умова на відбір даних з формуванням документа (текстового файла)	Умова на видалення даних
28	Список студентів курсу: ПІБ, дата народження, група (список), середній бал успішності, проживання у гуртожитку (так/ні)	За зростанням середнього балу	Студенти з середнім балом понад 80, які мешкають у гуртожитку	Неповнолітні студенти (вік до 18-ти років)	Студентів, які святкують день народження восени
29	Співробітники підприємства: табельний номер, ПІБ, дата народження, посада, відділ (список), оклад, наявність премії минулого місяця	За спаданням табельних номерів	Робітники з відділу реалізації, які отримали премію минулого місяця	Співробітники, яким до пенсії (60 років) лишилося менше п’яти років	Співробітники з окладом до 1500 грн.
30	Список моніторів: модель, фірма-виробник (список), діагональ у дюймах, можливість підтримки 3D (так/ні), дата поставки	За зростанням розміру діагоналі	Монітори фірми LG без 3D	Монітори, дата поставки яких відбулася понад півроку тому	Монітори з діагоналлю 15 дюймів

ПРАВИЛА ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ

Курсова робота має бути виконана та оформлена з дотриманням вимог до наукових робіт. Вона оформлюється на аркушах формату А4 на одній стороні з відступами (полями) сторінок: зліва – 25 мм, зверху і знизу – 20 мм, праворуч – 15 мм. Виконана курсова робота здається викладачу у прошитому вигляді або в спеціальних (пластикових) теках.

Текст роботи рекомендується оформляти з такими параметрами шрифту та абзаців:

шрифт – Times New Roman, 14, колір чорний;

абзаци – міжрядковий інтервал – 1; відступ першого рядка – 1,25; відступи перед і після – 0; інтервал перед і після – 0.

Текст курсової роботи повинен містити такі обов'язкові розділи:

1) титульна сторінка;
2) зміст;
3) вступ, який включає і завдання із зазначенням індивідуального варіанта;
4) розділ 1 "Теоретичні відомості: інструментарій використовуваний для виконання завдання";
5) розділ 2 "Програмний код основного модуля з поясненнями";
6) розділ 3 "Створення додаткових форм та меню";
7) розділ 4 "Здобуті результати у вигляді форм проекту";
8) висновки з аналізом здобутих результатів;
9) список використаних джерел.

Оптимальний обсяг курсової роботи – 20 – 25 друкованих сторінок або 35 – 40 рукописних сторінок.

Курсова робота починається з **титульної сторінки** за формою, наведеною у додатку. Це перша сторінка курсової роботи, яку включають у загальну нумерацію сторінок, але не нумерують. Далі номер сторінки проставляють у правому нижньому куті.

За титульною сторінкою друкуються послідовно зміст, вступ, розділи у порядку подання, висновки, список використаних джерел, додатки. Кожну структурну частину роботи починають з нової сторінки. Заголовки структурних частин роботи "ЗМІСТ", "ВСТУП", "РОЗДІЛ", "ВИСНОВКИ", "СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ", "ДОДАТКИ" друкують великими літерами симетрично відносно тексту.

У **вступі** слід розкрити актуальність тематики роботи, тобто причини важливості дослідження вибраної тематики, зазначити об'єкт, предмет та мету дослідження, сформулювати завдання, за допомогою якого може бути досягнута мета дослідження. У завданні слід зазначити індивідуальний варіант на кшталт такого:

Засобами Visual C++ створити програмний проект, який включатиме створення та опрацювання бінарного і текстового файлів з даними відповідно до індивідуально-

го варіанта. При створенні програмного проекту передбачити виконання таких завдань:

1)програмне створення бінарного файла і заповнення його такими даними: номер мобільного телефону, дата підключення (активації) номера, вартість однієї хвилини розмов, назва тарифу, назва мобільного оператора;

2)програмний перегляд даних створеного бінарного файла;

3)можливість редагування та зберігання відредагованого файла;

4)впорядкування (сортування) за зростанням вартості 1 хв. розмов;

5)відбір даних файла за умовою: абоненти, які користуються мобільним зв'язком понад три роки;

6)можливість програмного відбору даних за умовою (абоненти, у назві тарифів яких є слово "вільний") та створення на основі відібраних даних текстового звітного документа;

7)з метою забезпечення інформаційної частини проекту, а саме: для надання інформації про автора роботи та для ознайомлення із завданням, яке виконує проект, створити три додаткові форми: 1) про автора, 2) форму для виведення, редагування та форматування параметрів шрифту текстового документа із завданням і 3) форму-заставку. Перехід до створених форм та додаткових функцій для роботи з файлом даних організувати за допомогою меню.

Крім індивідуального варіанта завдання, у вступі слід вказати структуру роботи, наприклад: *"Курсова робота складається зі вступу, чотирьох розділів, висновків та списку використаної літератури. Загальний обсяг курсової роботи – 24 сторінки".*

Текст курсової роботи викладають, поділяючи матеріал на розділи. Основна частина роботи складається з чотирьох розділів. Номер розділу ставлять після слова "РОЗДІЛ", причому крапка після номера не ставиться. Потім з нового рядка друкується заголовок розділу, наприклад:

РОЗДІЛ 3
СТВОРЕННЯ ДОДАТКОВИХ ФОРМ ТА МЕНЮ

Розділи можуть поділятися на підрозділи і пункти. Підрозділи нумеруються у межах кожного розділу. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, між ними ставиться крапка. У кінці номера підрозділу повинна стояти крапка, наприклад: "2.3." (третій підрозділ другого розділу). У тому самому рядку пишеться назва підрозділу малими літерами (крім першої прописної), наприклад:

2.3. Видалення даних бінарного файла за умовою

У кінці заголовків крапки не ставлять. Якщо заголовок складається з двох або більше речень, їх розділяють крапкою.

У **розділі 1** доцільно сформувати текст з описом тих засобів, які використовувалися при написанні програмного коду для виконання завдання. Цей роз-

діл є теоретичним, у ньому розкривається сутність досліджуваних процесів і використовуваних методів для розкриття сутності зазначеної роботи, може надаватися їх класифікація та різні трактування з позицій різних авторів у науковій літературі.

Розділ 2 містить розроблений автором для виконання поставлених завдань програмний код основного модуля з докладними коментарями.

Розділ 3 курсової роботи має назву "Створення додаткових форм та меню" та містить програмний код для переходу і роботи з додатковими формами у проекті та код додаткових функцій для роботи з файлом даних, організованих за допомогою меню.

Розділ 4 "Вигляд здобутих результатів у вигляді форм проекту" носить практичний характер і містить різноманітні вигляди форм (скріншоти) на різних етапах виконання програмного проекту.

Кожен з рисунків (скріншотів), наведених у роботі, повинен бути підписаний і пронумерований послідовно у межах розділу. При цьому підписи рисунків складаються з номера рисунка у цьому розділі (наприклад, номер 4.2 буде у другого рисунка в четвертому розділі роботи) та його назви. Розміщуються підписи безпосередньо під рисунком з вирівнюванням по центру, наприклад:

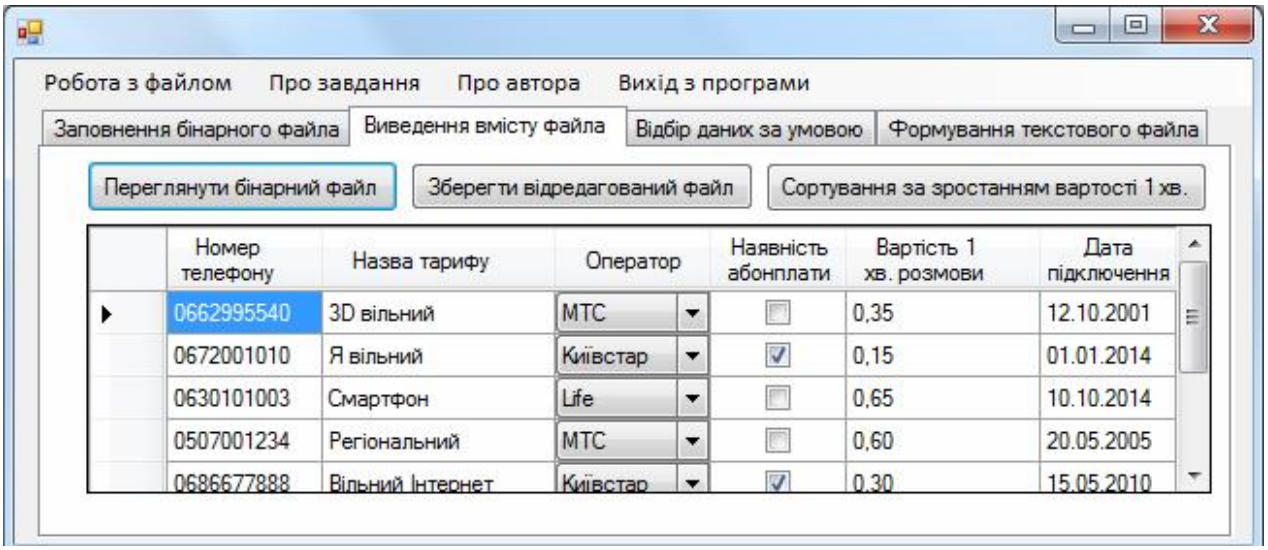


Рисунок 4.2 – Виведення вмісту файла після натиснення кнопки *Переглянути бінарний файл*

Висновки розміщують безпосередньо після викладення тексту, починаючи з нової сторінки. У висновках зробити аналіз здобутих результатів та набутих знань і навичок (при цьому заохочується самостійність суджень та оцінок), оцінити повноту виконання поставлених задавдань та рекомендації щодо можливостей використання матеріалів роботи.

Список використаної літератури має бути оформлений за вимогами ДСТУ (ГОСТ) 7.1:2006 "Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання" і містити 10 – 20 літературних джерел, включаючи посилання
на Інтернет-ресурси.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Загальні відомості про файли

Файлами є іменовані області пам'яті на зовнішньому носії, призначені для довготривалого зберігання інформації. Файли мають ***імена*** й є організовані в ієрархічну деревоподібну структуру з ***каталогів*** (тек) і простих файлів.

Часто використовується така метафора: якщо уявляти собі файли як книжки (лише зчитування) і блокноти (зчитування й записування), які стоять на полиці, то відкриття файла – це вибір книжки чи блокнота за заголовком на його обкладинці й відкриття обкладинки (на першій сторінці). Після відкриття можна читати, дописувати, викреслювати і правити записи, перегортати книжку. Сторінки можна порівняти з блоками файла, а полицю з книжками – з каталогом.

Для доступу до даних файла з програми в ній слід прописати функцію відкриття чи створення цього файла, тим самим встановити зв'язок між ім'ям файла і певною файловою змінною у програмі. Доступ до даних файла відбувається з так званої позиції зчитування-записування, яка автоматично просувається при операціях зчитування-записування. Існують, щоправда, функції для довільного змінювання цієї позиції.

C++ надає засоби опрацювання двох типів файлів: *текстових* і *бінарних (двійкових)*.

Текстові файли призначено для зберігання текстів, тобто сукупності символьних рядків змінної довжини. Кожен рядок завершується керувальною послідовністю `'\n'`, а розділювачами слів та чисел у рядку є пробіли й символи табуляції. Оскільки вся інформація текстового файла є символьною, програмне опрацювання такого файла полягає в читанні рядків, виокремлюванні з рядка слів і, за потреби, перетворюванні цифрових символьних послідовностей на числа відповідними функціями перетворювання. Створювати і редагувати текстові файли можна не лише в програмі, а і в якому завгодно текстовому редакторі, наприклад, Word, WordPad чи Блокноті.

Бінарні (двійкові) файли зберігають дані у тому самому форматі, в якому вони були оголошені, а їхній вигляд є такий самий, як і в пам'яті комп'ютера. І тому відпадає потреба у використанні розділювачів: пробілів, керувальних послідовностей, а отже, розмір використовуваної пам'яті порівняно з текстовими файлами з аналогічною інформацією є значно меншим. Окрім того, немає потреби у застосуванні функцій перетворювання числових даних. Але кожне опрацювання даних бінарних файлів можливе лише за наявності програми, якій має бути відомо, що саме і в якій послідовності зберігається у цьому файлі.

У цій роботі будуть розглянуті основні засоби створення й опрацювання текстових файлів з використанням спеціального простору імен System::IO платформи .NET Framework, хоча слід зазначити, що існують й інші підходи при роботі з файлами.

Деякі класи простору System::IO для роботи з файлами

Простір імен System::IO платформи .NET Framework містить декілька класів, які дозволяють здійснювати операції з потоками даних, файлами і каталогами (папками, директоріями), такі як читання і записування даних, пошук (визначення і змінення поточної позиції всередині потоку), архівація файлів і каталогів тощо. Наприклад, клас `StreamWriter` з цього простору імен має сучасні, потужні і доволі зручні засоби записування даних у файли як потоки даних.

Охарактеризуємо деякі з цих класів.

Клас	Опис
`File`	надає методи для створення, копіювання, видалення, переміщення і відкриття файлів
`FileInfo`	надає методи екземпляра для створення, копіювання, видалення, переміщення і відкриття файлів
`Directory`	надає статистичні методи для створення і переміщення у каталогах
`DirectoryInfo`	надає методи екземпляра для створення і переміщення у каталогах
`Path`	виконує операції щодо відомостей про шлях до файла чи каталогу (літеральний рядок (тип `String`), який вказує розташування файла у файловій системі – адресу каталогу)
`StreamReader`	надає засоби зчитування послідовності символів (байтів) з потоку у певному кодуванні
`StreamWriter`	надає засоби записування послідовності символів (байтів) у потік у певному кодуванні
`BinaryReader`	надає засоби зчитування простих типів даних (двійкових значень)
`BinaryWriter`	надає засоби записування простих типів даних (двійкових значень)

Розглянемо найпоширеніші методи цих класів, які дозволять створити файл, записати у нього дані, прочитати їх та ін.

Поширені дії при роботі з файлами

Дії	Методи
Відкрити файл у заданому режимі і доступі	`File::Open` `File::OpenRead` – відкрити файл у режимі читання `File::OpenWrite` – відкрити файл для записування
Закрити файл (потік)	`Close`
Записати у файл	`Write` – записати у потік значення аргументів `WriteLine` – записати рядок із символом `'\n'` (*Enter*)
Дописати у текстовий файл	`File::AppendText` `FileInfo::AppendText`

	методи відкривають існуючий файл і дописують у нього текст у кодуванні UTF-8 (якщо файла не існувало, він буде створений)

Дії	Методи
Зчитати з файла	`Read` – зчитати з потоку значення аргументів, записаних у дужках `ReadLine` – зчитати рядок і символ `'\n'` як ознаку кінця рядка `File::ReadAllText` – відкрити текстовий файл, зчитати всі рядки файла в один зазначений рядок і закрити файл
Створити текстовий файл	`File::CreateText` або `File::Create` `FileInfo::CreateText` або `FileInfo::Create`

Перемістити позицію файла на `offset` байтів відносно позиції
`SeekOrigin:`
`SeekOrigin:Begin` або
`SeekOrigin::End`
	`BaseStream::Seek(offset, SeekOrigin)` – для текстових файлів, а для бінарних файлів:
	`BinaryWriter::Seek(offset, SeekOrigin)`
	Наприклад:
	`// на початок файла`
	`f->BaseStream->Seek(0, SeekOrigin::Begin);`
	`// у кінець файла`
	`f->BaseStream->Seek(0, SeekOrigin::End);`

Повернути наступний доступний символ, але не використовувати його	`StreamReader::Peek` З цим методом можна організувати цикл для почергового опрацювання вмісту файла: `while (fr->Peek() > -1) . . .`
Перейменувати чи перемістити файл	`File::Move` або `FileInfo::MoveTo`
Видалити файл	`File::Delete` або `FileInfo::Delete`
Копіювати файл	`File::Copy` або `FileInfo::CopyTo`
Визначити наявність файла	`File::Exists`
Отримати дані про розмір файла	властивість `FileInfo::Length`
Отримати розширення імені файла	`Path::GetExtension`
Отримати повний шлях до файла	`Path::GetFullPath`
Отримати ім’я й розширення файла	`Path::GetFileName`
Змінити розширення файла	`Path::ChangeExtension`

Тобто для однієї дії при роботі з файлами існують декілька методів її реалізації. Розглянемо на прикладах деякі з цих методів.

Створити текстовий файл з ім’ям `MyTest.rtf` у поточному каталозі можна командою:

```
StreamWriter^ = File::CreateText("MyTest.rtf");
```

або командою:

```
TextWriter^ ft = gcnew
StreamWriter(File::OpenWrite("MyTest.rtf"));
```

Якщо створюваний файл слід розмістити не в поточному каталозі, а в якомусь іншому, то слід зазначити повний шлях до файла:

```
StreamWriter^ = File::CreateText("c:\\KP\\MyTest.rtf");
```

Створити бінарний файл з ім'ям `Test.dat` у поточному каталозі можна командою:

```
BinaryWriter^fb=gcnew
BinaryWriter(File::Open("Test.dat",FileMode::OpenOrCreate));
```

або двома командами:

```
FileStream^ data = gcnew FileStream("Test.dat",
FileMode::OpenOrCreate);
BinaryWriter^ fb = gcnew BinaryWriter(data);
```

При цьому режим відкриття `OpenOrCreate` дозволить не обнуляти вже існуючий файл (якщо такий є), а лише відкрити його.

Якщо створюваний файл слід розмістити не в поточному каталозі, а в якомусь іншому, слід зазначити повний шлях до файла.

Записати у файл рядок `s` можна за допомогою методу `WriteLine`:

```
String^ s = "Текст, який буде записано у файл окремим абза-
цом.";
ft->WriteLine(s);
```

Дописати дані у кінець файла можна командою

```
File::AppendAllText("MyTest.rtf", "Текст, який буде дописано у
файл\n");
```

При цьому, якщо файл з таким ім'ям не існував, він буде створений, тобто метод `AppendAllText` і створює файл, і відкриває його, і дописує у нього дані, поєднуючи можливості відразу декількох методів.

Для **зчитування рядків** текстового файла слід спочатку створити екземпляр класу `StreamReader`, після чого по черзі зчитувати всі рядки з файла:

```
StreamReader^ ft = File::OpenText("MyTest.rtf");
String^ s = "";
while ( s = ft->ReadLine() ) Console::WriteLine( s );
```

У наведеному циклі умовою припинення зчитування рядків з файла є нульове значення довжини зчитаного рядка. Така конструкція операторів дозволить переглянути вміст всього текстового файла у консольному режимі.

Переміщуватися по файлу можна не лише при зчитуванні даних, а й за допомогою методу `Seek`, наприклад:

```
ft->BaseStream->Seek(0, SeekOrigin::Begin); // перейти на по-
чаток файла
ft->BaseStream->Seek(0, SeekOrigin::End);   // перейти у кі-
нець файла
```

Закрити файл доречно командою

```
ft->Close();
```

Використання dataGridView при роботі з даними файлів

Елемент dataGridView на формі можна використовувати не лише для введення-виведення матриць, адже цей елемент має широкий спектр налаштовуваних властивостей, завдяки яким можна задавати різні властивості для різних стовпців елемента. Цю можливість доволі зручно використовувати для введення-виведення даних бінарних файлів, поля яких мають різні типи даних: числові, рядкові, логічні, списки та ін.

Розглянемо докладніше налаштування dataGridView. Після розміщення такого елемента на формі доцільно почергово створити на ньому необхідну кількість стовпців і при їх створенні відразу задати тип даних, заголовки та інші необхідні значення стовпців, наприклад:

– у властивості HeaderText доцільно вписати текст заголовка відповідного стовпця;

– у властивості ColumnType замість значення за замовчуванням (DataGridViewTextBoxColumn) для текстових і рядкових даних вибрати зі списку одне з можливих значень:

- DataGridViewComboBoxColumn – список з набором можливих текстових значень, для якого у властивості Items доцільно через *Enter* вписати текст можливих значень, наприклад: назви операторів мобільного зв'язку, назви посад співробітників або назви студентських груп;
- DataGridViewCheckBoxColumn – зазвичай відповідає логічному типу (значення true та false);
- DataGridViewImageColumn – використовується для виведення зображень;
- DataGridViewButtonColumn – використовується для виведення кнопок у клітинках таблиці;

– властивість AllowUserToAddRows дозволяє додавати нові рядки таблиці при заповненні значень останнього рядка таблиці (за замовчуванням значення true). Значення false забороняє автоматичне додавання нового рядка;

– властивість ColumnHeadersVisible дозволяє приховати заголовки стовпців при значенні false;

– RowHeadersVisible дозволяє приховати заголовки рядків при значенні false;

– властивість RowCount зберігає значення кількості рядків;

– властивість ColumnCount зберігає значення кількості стовпців;

– властивість Width задає ширину стовпця у пікселях;

– властивість ToolTipText містить текст підказки, яка виводитиметься при наведенні покажчика миші на заголовок стовпця;

– властивість AutoSizeMode дає можливість вибрати режим автоматичного формування ширини того чи іншого стовпця. Наприклад, значення AllCells задасть автоматичний підбір ширини стовпця за максимальним вмістом стовпця;

– властивість `ReadOnly` вмикає-вимикає дозвіл редагування даних у стовпці;

– властивість `SortMode` надає можливість задавати режим автоматичного сортування даних у стовпці.

– `Columns` – колекція стовпців;

– `Rows` – колекція рядків;

– `Font` – параметри шрифту.

ПРИКЛАД ВИКОНАННЯ КУРСОВОЇ РОБОТИ

1. Хід виконання роботи

Хід виконання буде розглянуто на конкретному прикладі створення програмного проекту, який включатиме створення та опрацювання бінарного і текстового файлів з такими даними: номер телефону, назва тарифного пакета, мобільний оператор, наявність абонементної плати, вартість хвилини розмов.

Крім створення і заповнення бінарного файла даними, у проекті будуть виконуватись такі завдання:

– відбір з файла даних про абонентів МТС без абонплати з вартістю понад 50 коп;

– можливість сортування даних за зростанням значень у стовпці з номерами телефонів;

– створення текстового документа (файла) з даними про абонентів, у назві яких є слово "вільний". Виведення вмісту даного файла на форму в окремий dataGridView;

– з метою забезпечення інформаційної частини програмного проекту, а саме: для надання інформації про автора роботи та для ознайомлення із завданням, яке розв’язує проект, будуть створені три додаткові форми. Доступ до цих форм, а також до команд виведення довідкових даних про файл даних (розмір, дату створення, час останнього редагування тощо) та до засобів виведення, редагування та форматування параметрів шрифту текстового документа буде організовано за допомогою меню.

Далі буде наведено програмний код усіх функцій для реалізації цих завдань.

2. Програмний код основного модуля

2.1. Робота з бінарним файлом

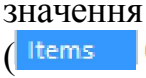
Створення програмного проекту слід розпочати зі створення на налаштування форми, розмістивши на ній відповідні командні кнопки та таблиці (елементи dataGridView) для введення-виведення даних бінарного файла. Але перед розміщенням на формі декількох dataGridView-ів доцільно створити елемент tabControl, який дозволить не захаращувати форму декількома таблицями, а розмістити їх на різних вкладках залежно від виконуваної частини завдання: чи то введення даних у файл, чи виведення вмісту всього файла, чи відбір певних даних файла за умовою (запитом). Відтак після розміщення на формі tabControl1 слід створити на ньому три вкладки, скориставшись командою *Добавить вкладку*, яку можна вибрати з контекстного меню або натиснувши на стрілочку у верхньому правому кутку елемента.

Для першої вкладки створити надпис Заповнення бінарного файла, вписавши його у властивості Text. Для другої та третьої вкладок створити надписи: Виведення вмісту файла та Відбір даних за умовою.

На першій вкладці розмістити кнопку button1 з надписом Записати дані у бінарний файл на перший елемент dataGridView. Після розміщення dataGridView1 на формі слід почергово створити на ньому п’ять стовпців з кількістю полів у завданні і при їх створенні відразу задати тип даних і заголовки стовпців:

– для 1-го стовпця – у властивості HeaderText вписати Номер телефону;

– для 2-го стовпця – у властивості HeaderText вписати Назва тарифу;

– для 3-го стовпця – у властивості HeaderText вписати Оператор, а для властивості ColumnType замість значення за замовчуванням вибрати зі списку значення DataGridViewComboBoxColumn. Після цього у властивості Items (

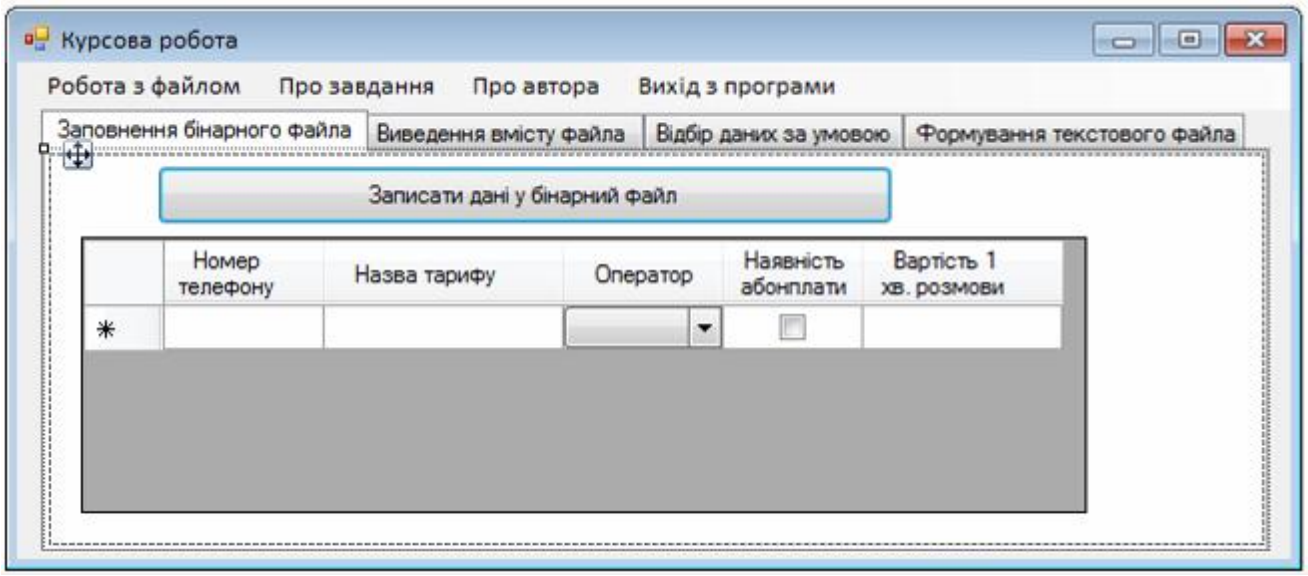
) вписати через *Enter* назви операторів, наприклад: МТС, Київстар, Life;

– для 4-го стовпця – у властивості HeaderText вписати Наявність абонплати, а для властивості ColumnType замість значення за замовчуванням вибрати зі списку значення DataGridViewCheckBoxColumn;

– для 5-го стовпця – у властивості HeaderText вписати Вартість 1 хв. розмови.

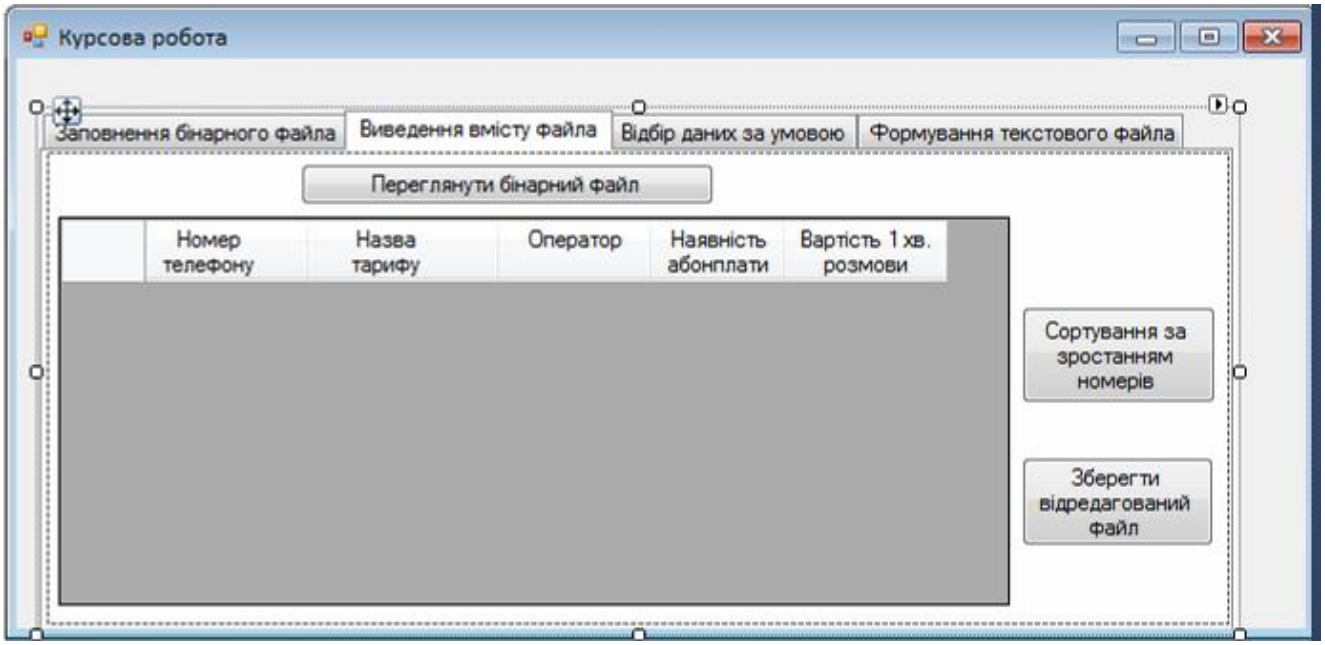
Тепер для всього dataGridView1 доцільно задати стиль заголовка стовпців у колекції властивостей ColumnHeadersDefaultCellStyle для Alignment вибрати значення MiddleCenter. Можна також для кожного зі стовпців задати потрібну ширину у властивості Width, наприклад: для першого і третього стовпців – 80, для четвертого стовпця – 70.

Тепер можна скопіювати dataGridView1 і вставити його копії на другу та третю вкладки, створивши таким чином вже налаштовані dataGridView2 та dataGridView3, для яких залишилось для властивості AllowUserToAddRows задати значення false.

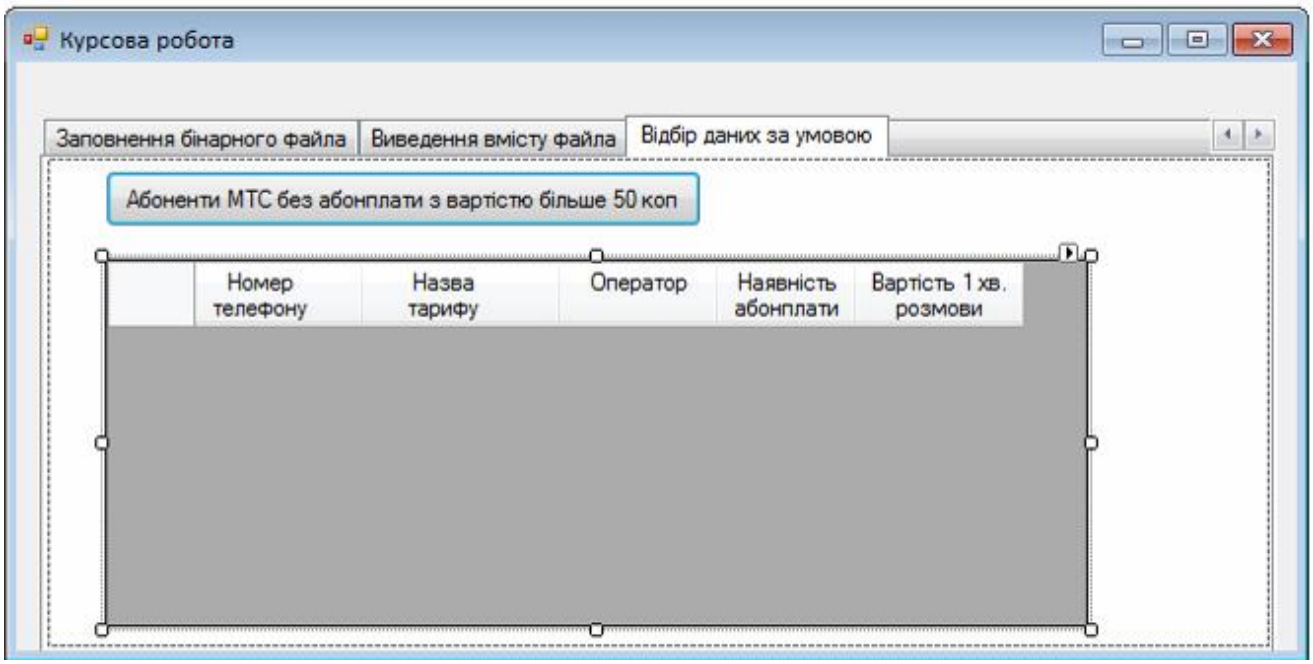


На другій вкладці слід розмістити три кнопки `button2`, `button3` та `button4`, вписати для них текст `Переглянути бінарний файл`, `Сортування за зростанням номерів` та `Зберегти відредагований файл`.

Після таких налаштувань елементів на формі вона набуде вигляду:



На третій вкладці розмістити ще одну кнопку `button5`, вписати для неї текст `Абоненти МТС без абонплати з вартістю понад 50 коп`.



Специфікою файла Form1.h з програмним кодом є наявність значної кількості рядків з командами налаштування елементів на формі, які автоматично створюються у цьому файлі при розміщенні нових елементів на формі та налаштуванні їх властивостей. Тому, щоби вписати команду для зазначення використання простору імен з програмними засобами файлового введення-виведення, слід піднятися на початок файла, віднайти рядок `using namespace System;` і під ним вписати команду:

```
using namespace System::IO;
```

Після цього слід спустися вниз і повернутися до створення програмного коду.

Оскільки ім'я файла буде використовуватись у програмному коді функцій майже всіх кнопок, доцільно для цього оголосити глобальну змінну (наприклад, `bfname`), а при створенні форми (функція `Form1_Load`) надати їй значення імені опрацьовуваного бінарного файла. Шаблон функції `Form1_Load` можна утворити подвійним клацанням миші на формі, подібно до того як подвійним клацанням по відповідній кнопці створюються шаблони функцій `button_Click`.

Програмний код:

```
. . . . . .
using namespace System;
using namespace System::IO;
. . . . . .
String ^bfname; // глобальна змінна  - ім'я  бінарного файла
private: System::Void Form1_Load(System::Object^ sender,
System::EventArgs^ e)
{
  bfname = "telephons.dat";
}
```

```
//------- Вкладка "Заповнення бінарного файла" ----------
// Записати дані у бінарний файл
private: System::Void
button1_Click(System::Object^sender,System::EventArgs^e)
{
  BinaryWriter^ fb = gcnew
BinaryWriter(File::Open(bfname,FileMode::OpenOrCreate));
  // або замість цієї команди можна скористатися двома такими
  // FileStream^ phons = gcnew FileStream(bfname,
FileMode::OpenOrCreate);
  // BinaryWriter^ fb = gcnew BinaryWriter(phons);
  fb->Seek(0, SeekOrigin::End); // перейти у кінець файла для допису-
вання даних
  try
  {
    int k=dataGridView1->Rows-
>GetRowCount(DataGridViewElementStates::Visible);
    for (int i=0; i < k-1; i++)
    { 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();
  }
  finally
  { fb->Close();  }
}
//------- Вкладка "Виведення вмісту файла" ----------
// Переглянути бінарний файл
private: System::Void
button2_Click(System::Object^sender,System::EventArgs^e)
{
  int i = 0;                          // номер рядка (запису)
  dataGridView2->Rows->Clear();       // очищення dataGridView2
  if (!File::Exists(bfname)) { MessageBox::Show("file not exists");
return;}
  BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(bfname));
  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();
      dataGridView2->Rows->Add(номер, тариф, оператор, абонплата,
                               String::Format("{0:0.00}", vart));
      dataGridView2->Rows[i]->HeaderCell->Value = (i+1).ToString();
      i++;
    }
  }
  finally
  { fb->Close();  }
}
// Сортування за зростанням вартості 1 хв. розмов
private: System::Void
button3_Click(System::Object^sender,System::EventArgs^e)
{
  button2_Click(button2, e);
  dataGridView2->Sort(dataGridView2->Columns[4],
ListSortDirection::Ascending);
  // Для сортування за спаданням тип Ascending слід замінити на
Descending
 }
// Зберегти відредагований файл
private: System::Void
button4_Click(System::Object^sender,System::EventArgs^e)
{
   BinaryWriter^ fb = gcnew
BinaryWriter(File::Open(bfname,FileMode::OpenOrCreate));
   fb->Seek(0, SeekOrigin::Begin); // Перейти на початок файла
   try
   {
     int kol=dataGridView2->Rows-
>GetRowCount(DataGridViewElementStates::Visible);
     for (int i = 0; i < kol; i++)
     { String^ номер = Convert::ToString(dataGridView2[0,i]->Value);
       String^ тариф = Convert::ToString(dataGridView2[1,i]->Value);
       String^ оператор = Convert::ToString(dataGridView2[2,i]-
>Value);
       bool абонплата = Convert::ToBoolean(dataGridView2[3,i]-
>Value);
       double vart = Convert::ToDouble(dataGridView2[4,i]->Value);
       fb->Write(номер);       fb->Write(тариф);   fb-
>Write(оператор);
       fb->Write(абонплата);   fb->Write(vart);
     }
   } finally { fb->Close(); }
}
//------- Вкладка "Відбір даних за умовою" ----------
// Абоненти МТС без абонплати з вартістю понад 50 коп.
```

```
private: System::Void
button5_Click(System::Object^sender,System::EventArgs^e)
{
   int i = 0;
   dataGridView3->Rows->Clear();
   if (File::Exists(bfname) == false)
    { MessageBox::Show("file not exists");  return; }
   BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(bfname));
   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 == "МТС" && vart>0.5 && !абонплата)
       {
         dataGridView3->Rows->Add(номер, тариф, оператор, абонплата,
                                  String::Format("{0:0.00}", vart));
         dataGridView3->Rows[i]->HeaderCell->Value =
(i+1).ToString();
         i++;
       }
     }
  }
  finally
  { fb->Close();
  }
}
```

Наведемо декілька прикладів конструювання умов для поля з даними типу "дата". Наприклад, відбір даних абонентів зі стажем підключення понад 3 роки можна організувати так:

```
    . . . . . .  .
    while (fb->BaseStream->Position < fb->BaseStream->Length)
    { . . . . . .  .
      String^ d = fb->ReadString();
      DateTime dd = Convert::ToDateTime(d); // Перетворення рядка до
дати
      TimeSpan  y = DateTime::Today - dd;    // обчислити різницю дат
      if (y.Days/365.25 >= 3)
        dataGridView3->Rows->Add(номер, тариф, оператор, абонплата,
                                 String::Format("{0:0.00}",vart), d);
    }
    . . . . . .
```

Ще одним варіантом формування цієї умови може бути такий:

```
if (DateTime::Today.Year - dd.Year >= 3) . . .
```

Якщо ж в умові йдеться про абонентів, підключених у поточному календарному місяці, умова може бути такою:

```
if (dd.Month == DateTime::Today.Month) . . .
```

Коли ж мова йде про визначення різниці між двома датами у днях, наприклад, для абонентів, підключених менше двох тижнів тому:

```
int days = (DateTime::Today - Convert::ToDateTime(d)).Days;
if (days < 14) . . .
```

2.2. Відбір даних бінарного файла за умовою з формуванням текстового документа

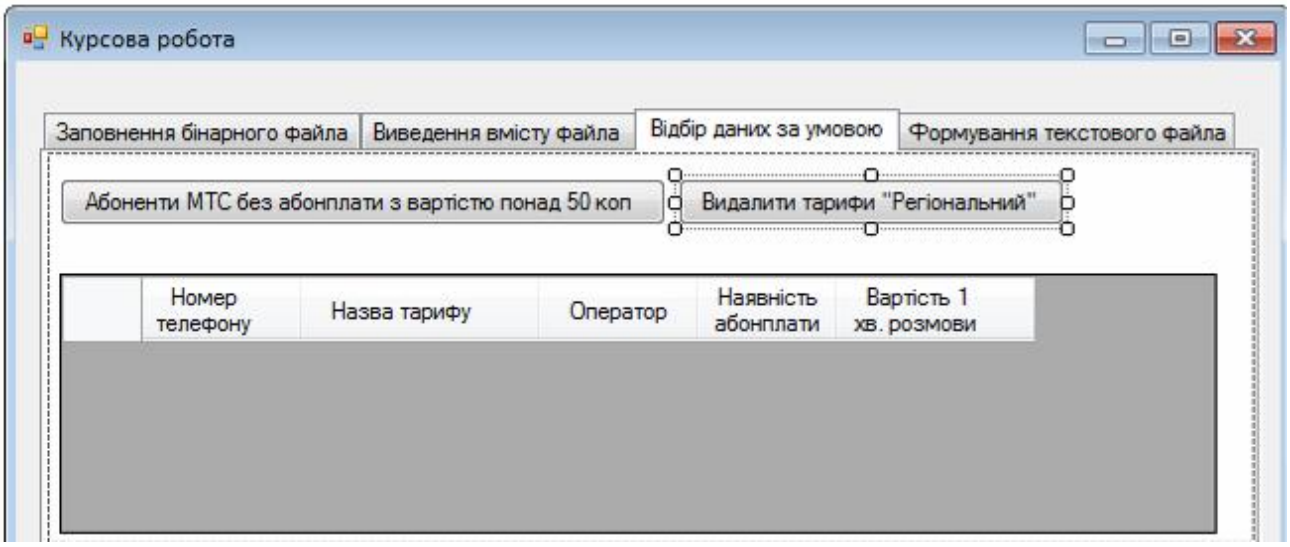
Для налаштування форми слід створити на елементі tabControl1 ще одну вкладку та вписати надпис Формування текстового файла. З попередньої вкладки скопіювати і вставити на щойно створену вкладку елемент dataGridView. Крім того, розмістити на цій вкладці кнопку з надписом "Абоненти, у назві тарифів яких є слово "вільний"". Після цього можна приступити до написання програмного коду для створеної кнопки.

```
//------- Вкладка "Формування текстового файла" ----------
// Текстовий файл: абоненти, у назві яких є слово "вільний"
private: System::Void
button6_Click(System::Object^sender,System::EventArgs^e)
{
  int i = 0;
  dataGridView4->Rows->Clear();
  if (File::Exists(bfname) == false)
   { MessageBox::Show("file not exists");  return; }
  BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(bfname));
  TextWriter^ ft = gcnew
StreamWriter(File::OpenWrite("абоненти.doc"));
  try
  { ft->WriteLine("Абоненти, у назві яких є слово вільний");
    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 (тариф->ToLower()->IndexOf("вільний",0) >= 0)
     {
       dataGridView4->Rows->Add(номер, тариф, оператор, абонплата,
                              String::Format("{0:0.00}", vart));
       dataGridView4->Rows[i]->HeaderCell->Value = (i+1).ToString();
       i++;
       String^ s = номер+"\t" + тариф + "\t" + оператор + "\t" +
                   Convert::ToString(абонплата) + "\t"+
```

```
                    Convert::ToString(String::Format("{0:0.00}",
vart));
        ft->WriteLine(s);
      }
    }
  }
  finally
  { fb->Close();   ft->Close();   }
}
```

2.3. Видалення даних з бінарного файла за умовою

У програмному проекті передбачимо можливість програмного видалення даних за умовою, наприклад видалення тарифів "Регіональний". Для цього на формі слід створити нову командну кнопку, розмістивши її на вкладці `Відбір даних за умовою`. На розміщеній кнопці вписати текст `"Видалити тарифи "Регіональний""`. Після цього можна приступити до написання програмного коду для створеної кнопки.



Структура програмного коду для видалення даних буде такою:

1) при читанні даних з файла у додатковий тимчасовий файл будуть записуватись тільки ті дані, які треба залишити у файлі, а дані, які задовольняють умові видалення, виводитимуться на форму у таблицю (елемент `dataGridView3`);

2) якщо виявиться, що у файлі не було зазначених в умові даних, про це виведеться відповідне повідомлення, а тимчасовий файл буде знищено. Інакше буде сформовано діалогове вікно із запитом на підтвердження видалення даних і кнопками "Так" і "Ні";

3) у разі натиснення користувачем кнопки "Так", вихідний файл буде замінений (метод `File::Replace`) на щойно сформований тимчасовий файл з даними без видалених. Крім того, користувачеві буде запропоновано перейти на другу вкладку, щоб переглянути новий вміст файла;

4) якщо ж користувач натиснув кнопку "Ні", сформований тимчасовий файл буде видалено, тобто вихідний файл залишиться без змін.

Програмний код:

```
// Видалити тарифи "Регіональний"
private: System::Void button7_Click(System::Object^sender,
System::EventArgs^ e)
{
   int i = 0;
   dataGridView3->Rows->Clear();
   if (File::Exists(bfname) == false)
    {  MessageBox::Show("file not exists");  return;  }
   BinaryReader^ fb = gcnew BinaryReader(File::OpenRead(bfname));
   BinaryWriter^ tmp = gcnew BinaryWriter(File::Open("tmp.bin",
                                                  FileMode::Create));
   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 (тариф == "Регіональний")
       {
         dataGridView3->Rows->Add(номер, тариф, оператор, абонплата,
                                  String::Format("{0:0.00}", vart));
         dataGridView3->Rows[i]->HeaderCell->Value = (i+1).ToString();
         i++;
       }
       else
       {  tmp->Write(номер);     tmp->Write(тариф);  tmp-
>Write(оператор);
          tmp->Write(абонплата); tmp->Write(vart);
     } }
     tmp->Close();
     if (!i)
     { MessageBox::Show("Таких записів у файлі не існує");
       File::Delete("tmp.bin");
     }
     else
      if (MessageBox::Show (L"Ви дійсно бажаєте видалити ці записи з
файла?",
                          L"Видалити дані з файла?",

System::Windows::Forms::MessageBoxButtons::YesNo) ==
                       System::Windows::Forms::DialogResult::Yes)
      {  fb->Close();
         File::Replace("tmp.bin", bfname,"Copy\\"+ bfname);
```

```
            dataGridView3->Rows->Clear();
            MessageBox::Show("Для перегляду нового вмісту файла перейдіть
                              на другу вкладку.");
        }
        else File::Delete("tmp.bin");
    }
    finally
    { fb->Close();
}  }
```

3. Створення додаткових форм та меню

Роботу з декількома формами в одному програмному проекті розглянемо, створивши три нові форми з різним наповненням та різним цільовим призначенням:

1) форма з довідковою інформацією про автора роботи;

2) форма, яка дасть можливість ознайомитись із завданням на курсову роботу (у вигляді виведення вмісту текстового документа). Програмний код для цієї форми надасть додаткові можливості редагування, форматування параметрів шрифту та зберігання змін у документі;

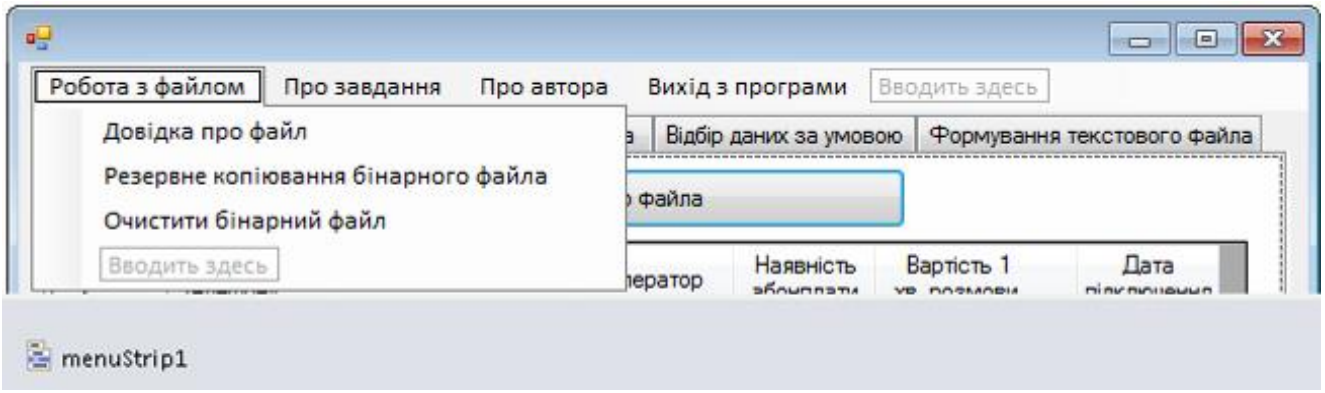
3) форма-заставка з тематичним рисунком, яка автоматично з'являтиметься після запуску програми та через декілька секунд "розчиниться у повітрі".

Програмний перехід до створених форм та додаткових функцій для роботи з файлом даних доцільно організувати за допомогою меню такої структури:

- Робота з файлом:
  - Довідка про файл;
  - Резервне копіювання бінарного файла;
  - Очистити;
- Про завдання;
- Про автора;
- Вихід з програми.

3.1. Функції основного модуля для команд меню

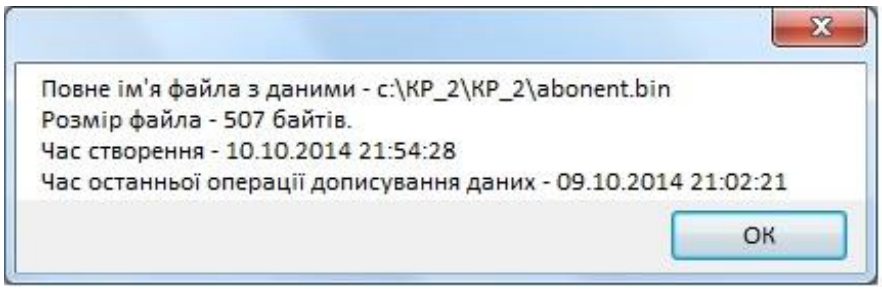
На формі програмного проекту слід розмістити елемент-меню `menuStrip1` та вписати такі надписи:



Програмний код для меню:

```
// Робота з файлом – Довідка про файл
private: System::Void довідкаПроФайлToolStripMenuItem_Click
                        (System::Object^ sender, System::EventArgs^ e)
{
  if (!File::Exists(bfname)) { MessageBox::Show("file not exists");
return; }
  FileInfo^ f = gcnew FileInfo(bfname);
  MessageBox::Show("Повне ім'я файла з даними - " + f->FullName +
                   "\nРозмір файла - " + (f->Length).ToString() +
```

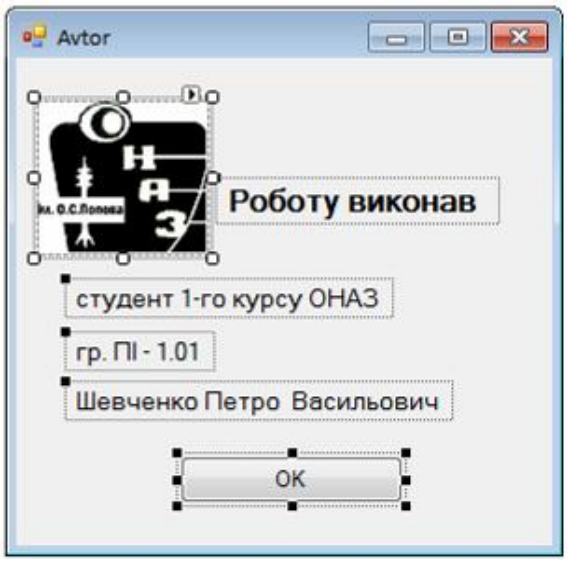
```
                    " байтів.\nЧас створення - " + f-
>CreationTime.ToString()+
                    "\nЧас останньої операції дописування даних - " +
                    f->LastWriteTime.ToString());
}
// Робота з файлом – Резервне копіювання бінарного файла
private: System::Void резервнеКопіюванняToolStripMenuItem_Click
                          (System::Object^ sender, System::EventArgs^ e)
{ if (!Directory::Exists("Copy"))
    Directory::CreateDirectory("Copy"); // Створити каталог
  FileInfo^ f = gcnew FileInfo(bfname);
  f->CopyTo("Copy\\"+fname,true);
}
// Робота з файлом – Очистити бінарний файл
private: System::Void очиститиФайлToolStripMenuItem_Click
                          (System::Object^ sender, System::EventArgs^ e)
{ BinaryWriter^ fb = gcnew BinaryWriter(File::Open(bfname,
FileMode::Create));
  fb->Close();
  dataGridView2->Rows->Clear();
}
// Про автора
private: System::Void проАвторaToolStripMenuItem_Click
                          (System::Object^ sender, System::EventArgs^ e)
{ Avtor^ fr2 = gcnew Avtor();
  fr2->ShowDialog();
}
// Про завдання
private: System::Void проЗавданняToolStripMenuItem_Click
                          (System::Object^ sender, System::EventArgs^ e)
{  Zavdan^ fr3 = gcnew Zavdan();
   fr3->ShowDialog();
}
// Вихід з програми
private: System::Void вихідЗПрограмиToolStripMenuItem_Click
                          (System::Object^ sender, System::EventArgs^ e)
{  Close();
}
```



Робота з файлом / Довідка про файл

3.2. Функції для модуля (форми) "Про автора"

Для створення і долучення нової форми з інформацією про автора слід виконати команду *Проект / Добавить новый элемент / Форма Windows Forms*, ввести ім'я форми – `Avtor` та натиснути кнопку *Добавить*. За допомогою чотирьох елементів `label`, кнопки `button` та `pictureBox1` створити форму на кшталт показаної праворуч. Зображення для елемента `pictureBox1` можна вибрати за допомогою властивості `Image`, а параметри шрифту для елементів `label` можна задати за допомогою колекції властивості `Font`. Після цього слід двічі клацнути по кнопці `button1` на цій формі та вписати команду:



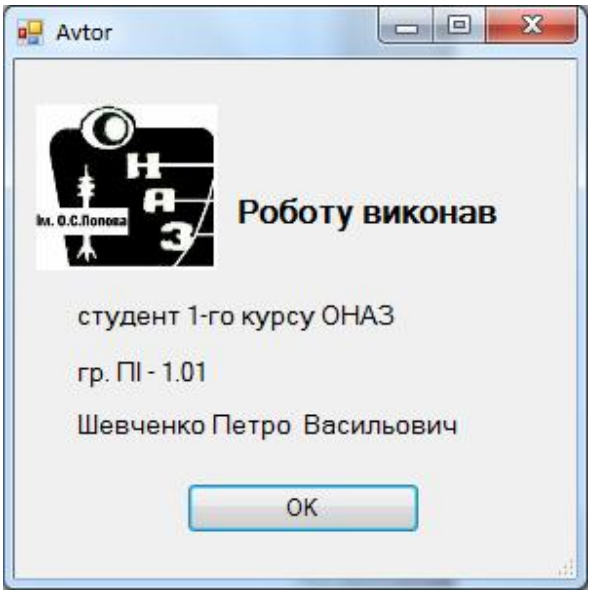

```
Close();
```

Тепер треба повернутися до файла Form1.h, піднятися на початок проекту і після команди `#pragma once` вписати команду долучення щойно створеної форми:

```
#include "Avtor.h"
```

Програмний код для командної кнопки буде доволі простим:

```
private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
   Close();
}
```

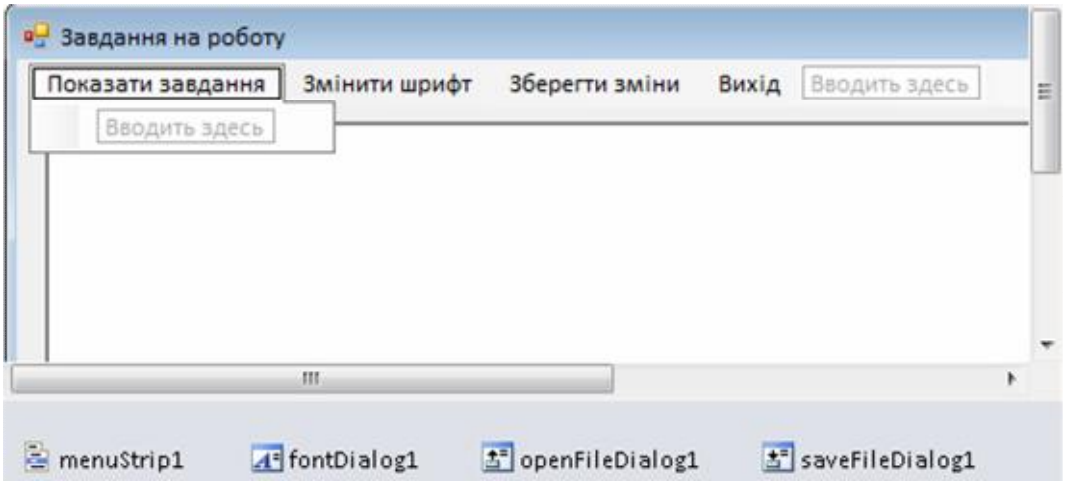


Про автора

3.3. Створення модуля (форми) "Завдання"

Для створення і долучення нової форми слід виконати команду *Проект / Добавить новый элемент / Форма Windows Forms*, ввести ім'я форми – Zavdan та натиснути кнопку *Добавить*.

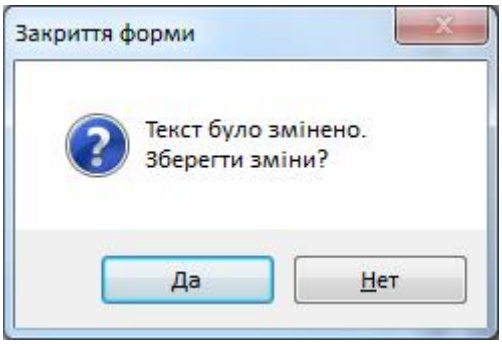
Налаштування форми Zavdan можна розпочати зі збільшення її ширини та формування надпису на ній Завдання на роботу (властивість Text), після чого розташувати на ній елементи richTextBox1, menuStrip1, fontDialog1, openFileDialog1 та saveFileDialog1. В меню (елемент menuStrip1) цієї форми ввести текст чотирьох команд: 1) Показати завдання, 2) Змінити шрифт, 3) Зберегти зміни та 4) Вихід. Після цього форма Zavdan набуде такого вигляду:



Подвійним клацанням по командам меню створити шаблони функцій та вписати в них такий програмний код:

```
// Показати завдання
private: System::Void показатиЗавданняToolStripMenuItem_Click_1
                          (System::Object^ sender, System::EventArgs^ e)
{ openFileDialog1->Filter = "Doc files(*.rtf)|*.rtf";
  if (openFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)
    richTextBox1->LoadFile(openFileDialog1->FileName);
}
// Змінити шрифт
private: System::Void змінитиШрифтToolStripMenuItem_Click
                          (System::Object^ sender, System::EventArgs^ e)
{ fontDialog1->ShowColor = true;
  fontDialog1->Font = richTextBox1->Font;
  fontDialog1->Color = richTextBox1->ForeColor;
  if (fontDialog1->ShowDialog() !=
System::Windows::Forms::DialogResult::Cancel)
  { richTextBox1->SelectionFont = fontDialog1->Font;
    richTextBox1->SelectionColor = fontDialog1->Color;
  }
}
```

```
// Зберегти зміни
private: System::Void зберегтиЗміниToolStripMenuItem_Click
                          (System::Object^ sender, System::EventArgs^ e)
{ saveFileDialog1->Filter = "Doc files(*.rtf)|*.rtf";
  if (saveFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)
    richTextBox1->SaveFile(saveFileDialog1->FileName);
  richTextBox1->Modified = false;
}
// Вихід
private: System::Void вихідToolStripMenuItem_Click
                          (System::Object^ sender, System::EventArgs^ e)
{   // Якщо були зміни, вивести повідомлення з кнопками Да-Нет
    // Якщо користувач натиснув Да, викликати функцію меню "Зберегти
зміни"
    if (richTextBox1->Modified)
      if (MessageBox::Show("Текст було змінено.\nЗберегти зміни?",
          "Закриття форми", MessageBoxButtons::YesNo,
          MessageBoxIcon::Question) ==
          System::Windows::Forms::DialogResult::Yes)
   зберегтиЗміниToolStripMenuItem_Click(зберегтиЗміниToolStripMenuItem,
                                                                  e);
    Close();
 }
```



Тепер треба повернутися до файла Form1.h, піднятися на початок проекту і після команди долучення форми `Avtor` вписати команду долучення щойно створеної форми Zavdan:

```
#include "Zavdan.h"
```

Можна запустити проект та впевнитись у правильності роботи всіх створених функцій, але для того, щоб можна було побачити текст файла із завданням, такий файл слід попередньо створити у текстовому редакторі, наприклад MS Word, записати у нього текст завдання і задати тип файла `rtf`.

3.4. Створення модуля (форми) "Заставка"

Тепер займемося створенням форми-заставки. Для її створення слід виконати команду *Проект / Добавить новый элемент / Форма Windows Forms*, ввести ім'я форми – `Zastavka` та натиснути кнопку *Добавить*. Розмістити на формі елементи `timer1` та `pictureBox1`, зображення до якого завантажити за допомого властивості `Image`.

Подвійним клацанням по таймеру та по формі створити шаблони функцій і вписати у них такий програмний код:

```
private: System::Void
Zastavka_Load(System::Object^sender,System::EventArgs^e)
{ this->timer1->Start();
}
private: System::Void timer1_Tick(System::Object^
sender,System::EventArgs^e)
{ this->Opacity -= 0.02; // кожної 0,1 секунди прозорість форми зменшуватиметься на 2%
```

```
  if (this->Opacity == 0) Close();
}
```

Тепер треба повернутись до файла Form1.h, піднятися на початок проекту і долучити модуль щойно створеної форми Zastavka:

```
#include "Zastavka.h"
```

Крім того, слід віднайти функцію Form1_Load та дописати до того оператора, що там є, ще три оператори для запуску форми-заставки:

```
private: System::Void Form1_Load(System::Object^ sender,
System::EventArgs^ e)
{ bfname = "telephons.dat";
  this->Hide();
  Zastavka^ fr4 = gcnew Zastavka();
  fr4->ShowDialog();
}
```

Запустити проект та впевнитись у правильності роботи всіх створених функцій.

4. Здобуті результати у вигляді форм (скріншоти)

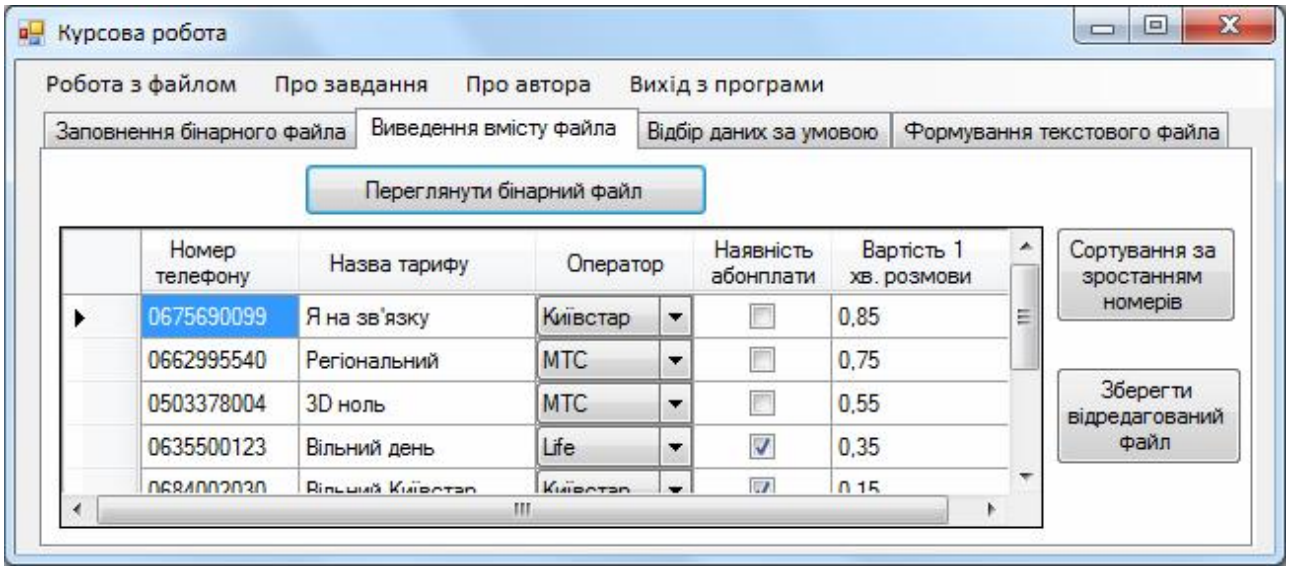


Рисунок 4.1 – Вигляд головної форми після натиснення кнопки "Переглянути бінарний файл" на вкладці "Виведення вмісту файла"

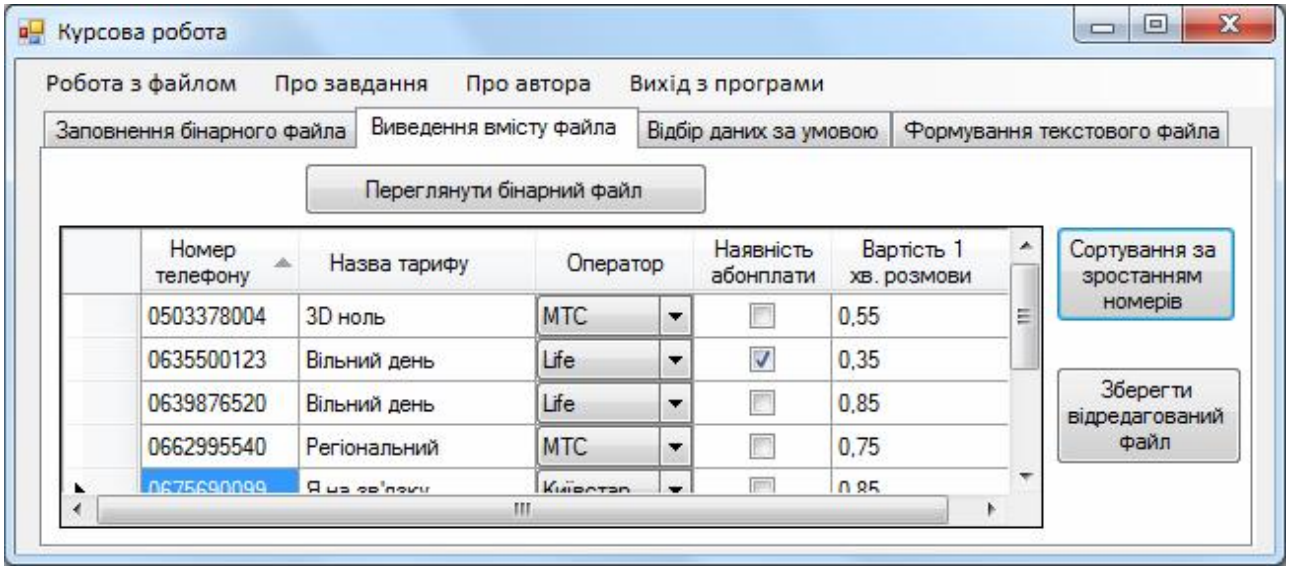


Рисунок 4.2 – Вигляд головної форми після натиснення кнопки "Сортування за зростанням номерів" на вкладці "Виведення вмісту файла"

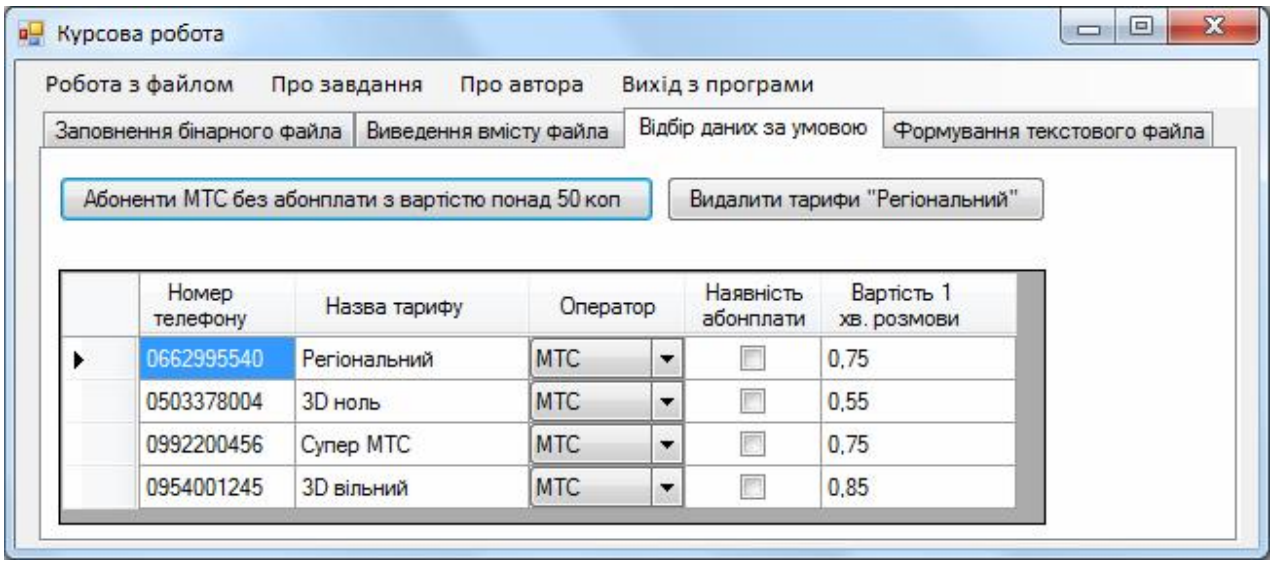


Рисунок 4.3 – Вигляд головної форми після натиснення кнопки "Абоненти МТС без абонплати з вартістю понад 50 коп" на вкладці "Відбір даних за умовою"

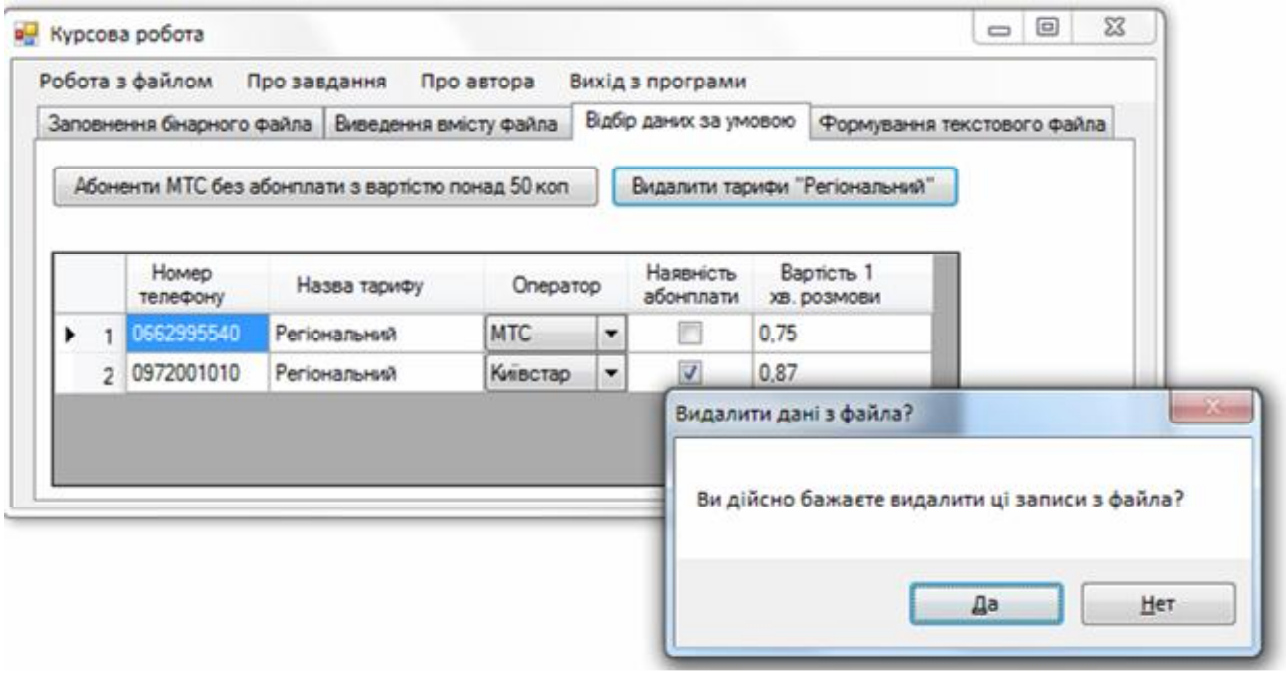


Рисунок 4.4 – Вигляд головної форми після натиснення кнопки "Видалити тарифи "Регіональний"" на вкладці "Відбір даних за умовою"

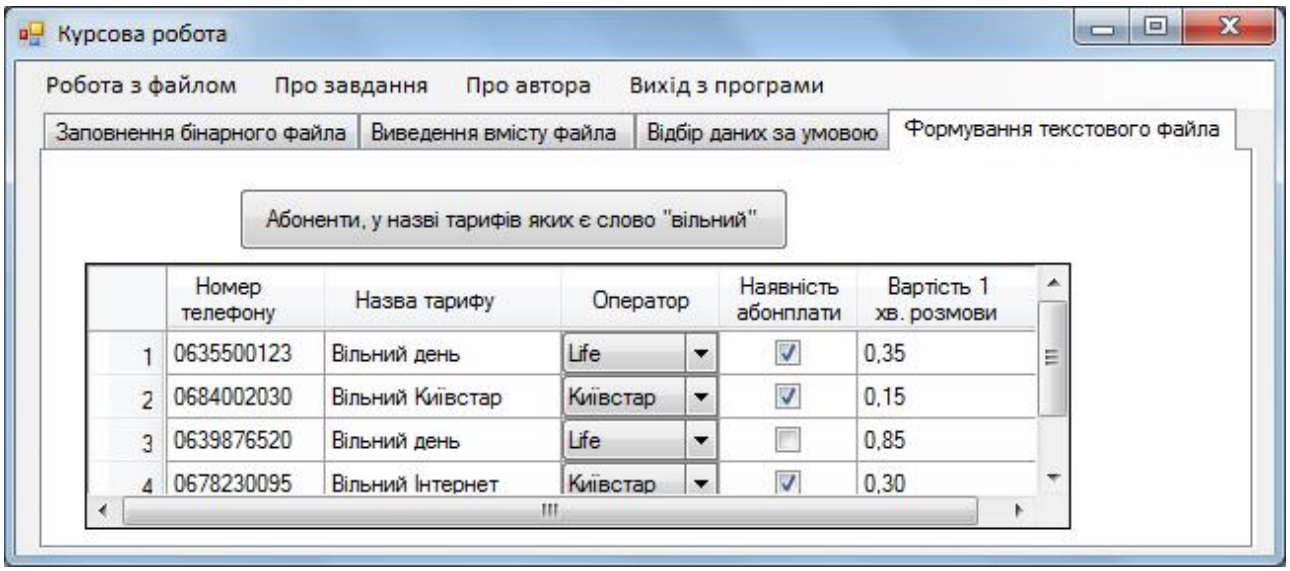


Рисунок 4.5 – Вигляд головної форми після натиснення кнопки "Абоненти, у назві тарифів яких є слово "вільний"" на вкладці "Формування текстового файла"

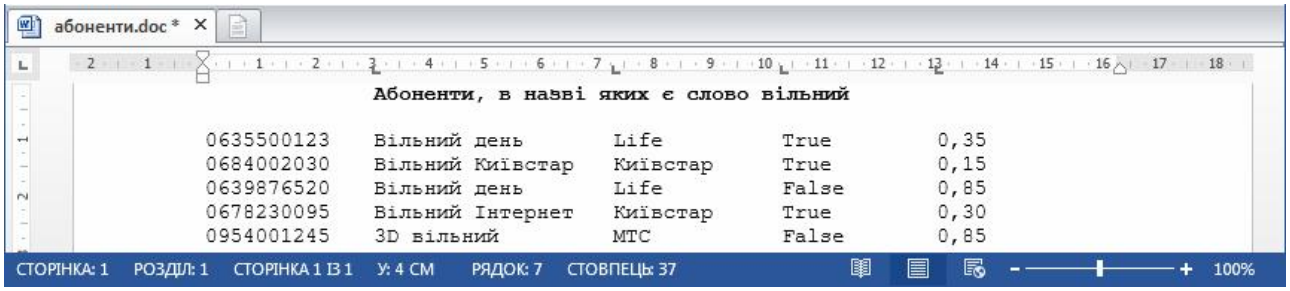


Рисунок 4.6 – Вигляд сформованого текстового документа з ім’ям "абоненти.doc"

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Библиотека классов платформы .NET Framework [Электронный ресурс]. – Режим доступа: http://msdn.microsoft.com/ru-ru/library/gg145045(v=vs.110).aspx. – Название с экрана.

2. Зиборов В. В. MS Visual C++ 2010 в среде .NET. Библиотека программиста / Зиборов В. В. – СПб. : Питер, 2012. – 320 с.

3. Хортон А. Visual C++ 2010: полный курс.; пер. с англ. / Хортон А. – М. : ООО "И.Д. Вильямс", 2011. – 1216 с.

4. Довбуш Г. Ф. Visual C++ на примерах / Г.Ф. Довбуш, А.Д. Хомоненко ; под ред. проф. А.Д. Хомоненко. – СПб. : БХВ-Петербург, 2007. – 528 с.

5. Visual C++ .NET: пособие для разработчиков C++ / [А. Корера, С. Фрейзер, С. Джентайл и др.]. – М. : ЛОРИ, 2003. – 398 с.

6. Дейтел Х. М. Как программировать на C++; пер с англ. / Х. М. Дейтел, П. Дж. Дейтел. – М. : ООО "Бином-Пресс", 2008. – 1456 с.

7. Стивен Прата. Язык программирования C++. Лекции и упражнения : учебник ; пер с англ. / Стивен Прата. – СПб.: ООО "ДиаСофтЮП", 2005. – 1104 с.

8. Страуструп Б. Язык программирования C++. Специальное издание ; пер. с англ. / Страуструп Б. – М. : ООО "Бином-Пресс", 2006. – 1104 с.

9. C++. Основи програмування. Теорія та практика: підручник / [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред. О. Г. Трофименко. – Одеса : Фенікс, 2010. – 544 с.

10. C++. Теорія та практика: навч. посіб. з грифом МОНУ/ [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред. О. Г. Трофименко. – Одеса : ВЦ ОНАЗ, 2011. – 587 с.

11. Трофименко О. Г. Основи програмування. Базові алгоритми: метод. вказівки для лаб. і практ. робіт / О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката. – Ч. 1. – Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2014. – 108 с.

12. Трофименко О. Г. Основи програмування. Опрацювання структурованих типів: метод. вказівки для лаб. і практ. робіт / О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката. – Ч. 2. – Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2014. – 130 с.

13. Трофименко О. Г. Основи програмування. Програмне опрацювання файлів: метод. вказівки для лаб. і практ. робіт / О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката. – Ч. 3. – Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2015. – 80 с.

14. Трофименко О. Г. Створення багатомодульних програмних проектів для опрацювання даних у файлах засобами Visual C++: метод. вказівки для виконання курсової роботи з дисципліни "Основи програмування" / О. Г. Трофименко, Ю. В. Прокоп. – Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2015. – 44 с.

ДОДАТОК

Приклад оформлення титульної сторінки курсової роботи

Міністерство освіти і науки України

Одеська національна академія зв'язку ім. О.С. Попова

Кафедра інформаційних технологій

К У Р С О В А Р О Б О Т А

з дисципліни: "Комп'ютерні технології та програмування"
(назва дисципліни)

на тему: "Створення багатомодульних програмних проектів зі створенням та опрацювання даних у файлах засобами Visual C++"

студента 1-го курсу, групи КТ-1.18
напряму підготовки 6.050202
"Автоматизація та комп'ютерно-інтегровані технології"
Шевченка Костянтина Миколайовича

Керівник доцент, к.т.н. Трофименко О.Г.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____
Кількість балів: ____ Кількість балів _____
_____ Трофименко О.Г.
(підпис) (прізвище та ініціали)

Одеса 2016

Навчально-методичне видання

Трофименко О. Г.,
Прокоп Ю. В.,
Буката Л.М.

Створення багатомодульних програмних проектів для опрацювання даних у файлах засобами Visual C++

Методичні вказівки для виконання курсової роботи
з дисципліни
"Комп'ютерні технології та програмування"

Коректор Кодрул Л.А.

Комп'ютерне верстання Гардиман Ж.А.