

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА
ПОЯСНЮВАЛЬНА ЗАПИСКА
з дисципліни “Об’єктно-орієнтоване програмування”
ЗАПИСНА КНИЖКА

Керівник проф.

Бондарєв В.М.

Студентка гр. ПЗП-22-1

Зозуля О.Ю.

Комісія:

Проф. Бондарєв В.М.,

Ст. викл. Черепанова Ю.Ю.,

Ст. викл. Ляпота В.М.

Харків 2023

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра *програмної інженерії*

Рівень вищої освіти *перший (бакалаврський)*

Дисципліна *Об'єктно-орієнтоване програмування*

Спеціальність *121 Інженерія програмного забезпечення*

Освітня програма *Програмна інженерія*

Курс *1.*

Група *ПЗПІ-22-1.*

Семестр *2.*

ЗАВДАННЯ

на курсовий проект студентки

Зозулі Олени Юріївни

1 Тема проекту: *Записна книжка*

2 Термін здачі студентом закінченого проекту: *“16” - червня - 2023 р.*

3 Вихідні дані до проекту:

Специфікація програми, методичні вказівки до виконання курсової роботи.

4 Зміст розрахунково-пояснювальної записки:

Вступ, опис вимог, проектування програми, інструкція користувача, висновки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапу	Термін виконання
1	Видача теми, узгодження і затвердження теми	13.02.2023 - 14.03.2023 р.
2	Формулювання вимог до програми	28.04.2023 - 30.04.2023 р.
3	Розробка моделей	30.04.2023 р.
4	Розробка функцій зберігання, очищення, завантаження даних	09.05.2023 р.
5	Розробка функцій додавання, видалення, редагування даних	12.05.2023 р.
6	Розробка функції пошуку	22.05.2023 р.
7	Розробка функції пошуку сортування даних і виводу привітань з днем народження	26.05.2023 – 29.05.2023 р.
8	Тестування і доопрацювання розробленої програмної системи	29.05.2023 - 30.05.2023 р.
9	Оформлення пояснювальної записки, додатків, графічного матеріалу	28.05.2023 - 31.05.2023 р.
10	Захист	31.05.2023 р.

Студентка

Зозуля Олена Юріївна

Керівник

Бондарев Володимир Михайлович

«29» травня 2023 року

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 46 с., 1 табл., 48 рис., 2 джерела.

ДРУЗІ, СПИСОК ДРУЗІВ, ЗАПИСНА КНИЖКА, МОВА ПРОГРАМУВАННЯ C#, ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПЛАТФОРМА .NET, WINDOWS FORMS.

Метою роботи є розробка програми «Записна книжка», яка буде надавати користувачу можливість взаємодіяти зі списком друзів.

В результаті отримана програма, що дозволяє створювати список друзів. Є можливості додавання друзів за певними ознаками, такими як: прізвище, ім'я, по батькові, дата народження, адреса, номер телефону, місце праці/навчання, посада, характер знайомства, редагування даних, видалення осіб зі списку. Є можливість пошуку друзів за довільним шаблоном, перегляду інформації про кожного друга, сортування за алфавітом та датою останнього коригування. Також, розроблено автоматичне формування привітання з днем народження і можливість копіювати текст побажань до буфера обміну.

В процесі розробки використано середовища Microsoft Visual Studio 2022, фреймворк Windows Forms, платформи .NET Framework 4.7.2, мова програмування C#.

ЗМІСТ

ВСТУП	6
1 ОПИС ВИМОГ	7
1.1 Сценарії використання	7
1.2 Функціональні вимоги	11
2 ПРОЕКТУВАННЯ ПРОГРАМИ	25
3 ІНСТРУКЦІЯ КОРИСТУВАЧА	28
3.1 Встановлення програми	28
3.2 Видалення програми	28
3.3 Робота з програмою	28
ВИСНОВКИ	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	46
ДОДАТКИ	47

ВСТУП

Метою цієї роботи є створення програми за допомогою фреймворку Windows Forms для ведення записів про друзів.

Даний програмний продукт має містити список друзів користувача. У даній програмі мають бути реалізовані можливості додавання нового друга до списку, редагування інформації про кожного, видалення і перегляд даних, має бути функція пошуку за довільним шаблоном, функція сортування за алфавітом і за датою останнього коригування, а також, функція автоматичного формування привітання з днем народження.

Програма, отримана в результаті курсової роботи, призначена для зручного керування персональними контактами. Основним призначенням цієї програми є забезпечення користувача зручним інструментом для зберігання контактної інформації, такої як імена, адреси, телефонні номери та інші деталі про своїх друзів.

Загалом, програма "Записна книжка" призначена для усіх людей планети, будь-хто може користуватися цією програмою, починаючи зі школяра і закінчуючи літньою людиною, яка вміє користуватися гаджетами.

1 ОПИС ВИМОГ

1.1 Сценарії використання

Сценарій 1. Додавання друга

Передумова

Користувач запускає програму, відкривається головне вікно.

Основний сценарій

1. На верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку обирає пункт “Add”.
2. Програма відкриває форму з полями для заповнення.
3. Користувач заповнює форму, надаючи дані про людину. Натискає кнопку “ОК”.
4. Виводиться діалогове вікно і запрошує підтвердження додавання людини.
5. Користувач підтверджує додавання.
6. Програма додає людину, виводить повідомлення про успішне додавання людини і оновлює список.

Додатковий сценарій

1. На верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку обирає пункт “Add”.
2. Програма відкриває форму з полями для заповнення.
3. Користувач заповнює форму, надаючи дані про людину. Натискає кнопку “ОК”.

4. Виводиться діалогове вікно і запрошує підтвердження додавання людини.

5. Користувач не підтверджує додавання.

Додатковий сценарій

1. На верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку обирає пункт “Add”.

2. Програма відкриває форму з полями для заповнення.

3. Користувач заповнює форму, надаючи дані про людину. Натискає кнопку “ОК”.

4. Програма перевіряє введену користувачем інформацію, знаходить помилки і виводить користувачу повідомлення про них.

Сценарій 2. Редагування інформації про друга

Передумова

Користувач запускає програму, відкривається головне вікно, список друзів не пустий.

Основний сценарій

1. Користувач обирає людину зі списку, дані якої хоче змінити, на верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку обирає пункт “Edit”.

2. Програма відкриває форму для редагування інформації.

3. Користувач редагує необхідні дані. Натискає кнопку “ОК”.

4. Виводиться діалогове вікно і запрошує підтвердження редагування даних.

5. Користувач підтверджує редагування.
6. Програма редагує дані, виводить повідомлення про успішне редагування людини і оновлює список.

Додатковий сценарій

1. Користувач обирає людину зі списку, дані якої хоче змінити, на верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку обирає пункт “Edit”.
2. Програма відкриває форму для редагування інформації.
3. Користувач редагує необхідні дані. Натискає кнопку “OK”.
4. Програма перевіряє введену користувачем інформацію, знаходить помилки і виводить користувачу повідомлення про них.

Додатковий сценарій

1. Користувач обирає людину зі списку, дані якої хоче змінити, на верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку обирає пункт “Edit”.
2. Програма відкриває форму для редагування інформації.
3. Користувач редагує необхідні дані. Натискає кнопку “Cancel”.
4. Виводиться діалогове вікно і запрошує підтвердження виходу з форми для редагування людини.
5. Користувач підтверджує вихід.

Сценарій 3. Видалення друга

Передумова

Користувач запускає програму, відкривається головне вікно, список друзів не пустий.

Основний сценарій

1. Користувач обирає людину зі списку, дані якої хоче змінити, на верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку обирає пункт “Delete”.

2. Виводиться діалогове вікно і запрошує підтвердження видалення людини зі списку.

3. Користувач підтверджує видалення.

4. Програма видаляє обрану людину зі списку і оновлює список.

Додатковий сценарій

1. Користувач обирає людину зі списку, дані якої хоче змінити, на верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку обирає пункт “Delete”.

2. Виводиться діалогове вікно і запрошує підтвердження видалення людини зі списку.

3. Користувач не підтверджує видалення.

Сценарій 4. Перегляд інформації про доданих друзів

Передумова

Користувач запускає програму, відкривається головне вікно, список друзів не пустий.

Основний сценарій

1. Користувач обирає необхідну людину зі списку, якщо список не пустий, далі на верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку обирає пункт “View info”.
2. Програма відкриває форму з інформацією про обрану людину.

Сценарій 5. Пошук друга

Передумова

Користувач запустив програму та відкрив головне вікно.

Основний сценарій

1. Користувач вводить дані на панель пошуку.
2. Програма автоматично здійснює пошук.

1.2 Функціональні вимоги

Функція 1. Додавання друга

Функція додавання здійснюється з головного вікна програми (рис. 1.1). На верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку (рис. 1.2) обирає пункт “Add”.

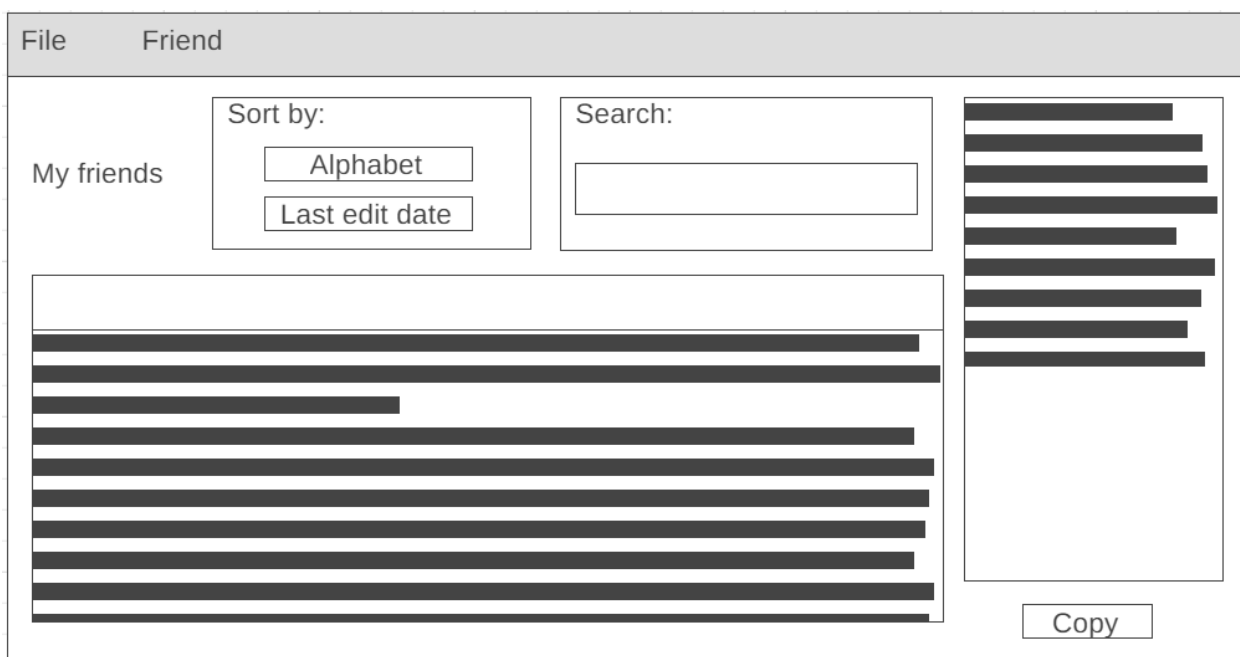


Рисунок 1.1 – Головне вікно програми

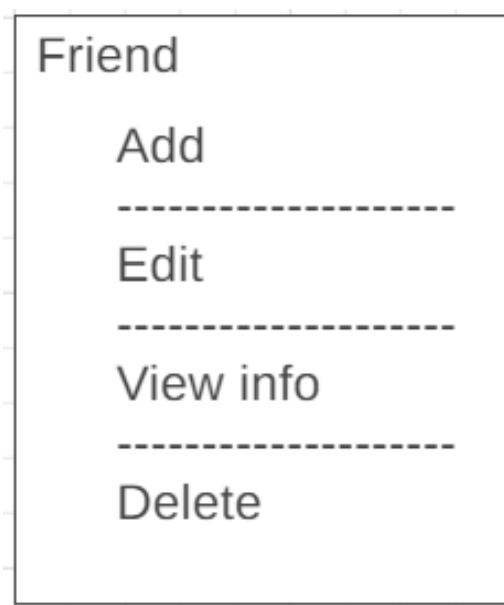
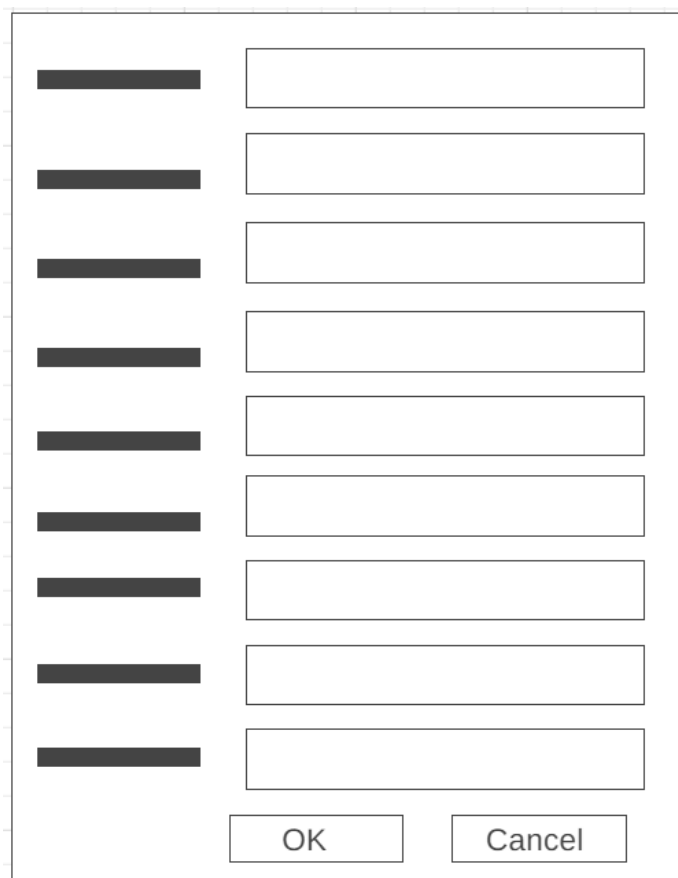


Рисунок 1.2 – Випадний список

Програма відкриває форму з полями для заповнення (рис. 1.3).



The image shows a graphical user interface window titled "Add Friend". It contains a vertical list of nine dark rectangular labels on the left side. To the right of each label is a horizontal text input field. At the bottom right of the window, there are two buttons labeled "OK" and "Cancel".

Рисунок 1.3 – Вікно додавання друга

Щоб додати людину до записної книжки, користувач має заповнити 9 полів:

1. Last name,
2. Name,
3. Surname,
4. Date of birth,
5. Address,
6. Phone number,
7. Place of work/study,
8. Position,
9. Acquaintance.

Усі поля, окрім дати народження та номеру телефона, текстові. У полях прізвища, ім'я та по батькові має бути перевірка, щоб користувач не вводив числа. У цих полях має бути не менше 2х літер і не більше 50ти, в усіх останніх не більше 100. У полі дати народження, користувач має вводити дату ось так: DD/MM/YY або DD/MM/YYYY. У полі номеру телефона, користувач має вводити ось так: +38XXXXXXXXXX. Усі поля є обов'язковими.

Користувач натискає кнопку "ОК". Після цього програма фіксує поточний час в останній колонці таблиці. Це знадобиться для функції сортування.

Якщо користувач робить некоректний ввід даних, то програма про це сповіщає і виводить текст помилки (рис. 1.4).

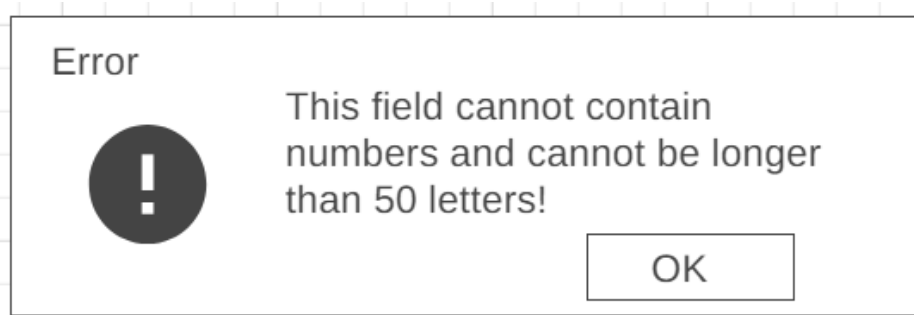


Рисунок 1.4 – Помилка вводу даних

Після успішної операції, програма буде виводити повідомлення (рис. 1.5).

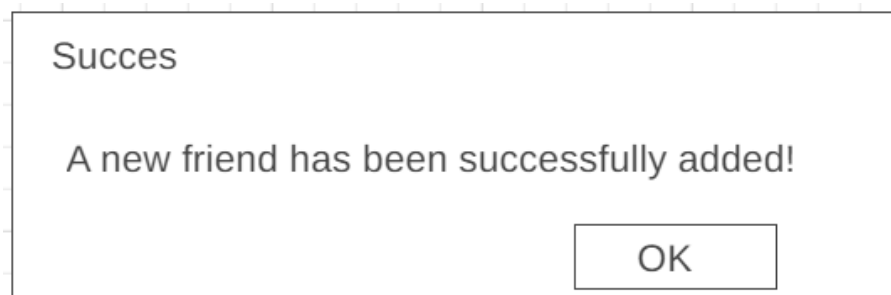


Рисунок 1.5 – Повідомлення після успішної операції

Після цього користувач повертається на головне вікно програми (рис. 1.1).

Функція 2. Редагування інформації про друга

Функція редагування здійснюється з головного вікна програми (рис. 1.1). На верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку (рис. 1.2) обирає пункт “Edit”, попередньо обравши необхідну людину зі списку.

Програма відкриває форму з полями, в які передаються дані про обрану людину, з можливістю зміни даних (рис. 1.6).

<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

OK Cancel

Рисунок 1.6 – Вікно редагування інформації про друга

Щоб редагувати дані про людину, користувач може змінювати будь-яке з 9 полів:

1. Last name,
2. Name,
3. Surname,
4. Date of birth,
5. Address,
6. Phone number,
7. Place of work/study,
8. Position,
9. Acquaintance.

Усі поля, окрім дати народження та номеру телефона, текстові. У полях прізвища, ім'я та по батькові має бути перевірка, щоб користувач не вводив числа. У цих полях має бути не менше 2х літер і не більше 50ти, в усіх останніх не більше 100. У полі дати народження, користувач має вводити дату ось так: DD/MM/YY або DD/MM/YYYY. У полі номеру телефона, користувач має вводити ось так: +38XXXXXXXXXX. Усі поля є обов'язковими.

Користувач натискає кнопку "ОК". Після цього програма фіксує поточний час в останній колонці таблиці. Це знадобиться для функції сортування.

Якщо користувач робить некоректний ввід даних, то програма про це сповіщає і виводить текст помилки (рис. 1.4).

Після успішної операції, програма буде виводити повідомлення (рис. 1.7).

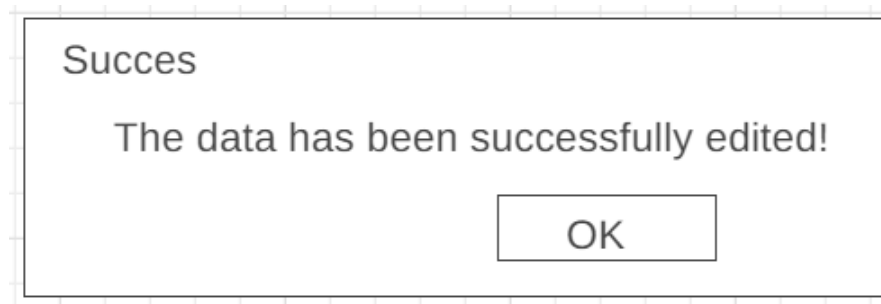


Рисунок 1.7 – Повідомлення після успішної операції

Після цього користувач повертається на головне вікно програми (рис. 1.1).

Функція 3. Видалення друга

Функція видалення здійснюється з головного вікна програми (рис. 1.1). На верхній панелі програми, користувач натискає кнопку "Friend" і у випадному списку (рис. 1.2) обирає пункт "Delete", попередньо обравши необхідну людину зі списку.

Програма запрошує підтвердження видалення (рис. 1.8).

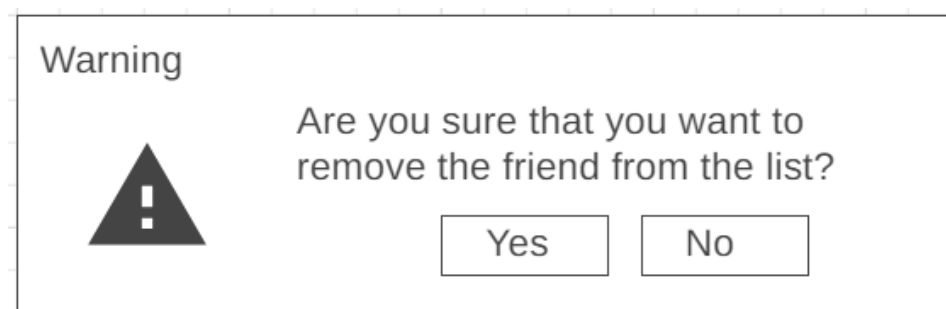


Рисунок 1.8 – Підтвердження операції

Після успішної операції, програма буде виводити повідомлення (рис. 1.9).

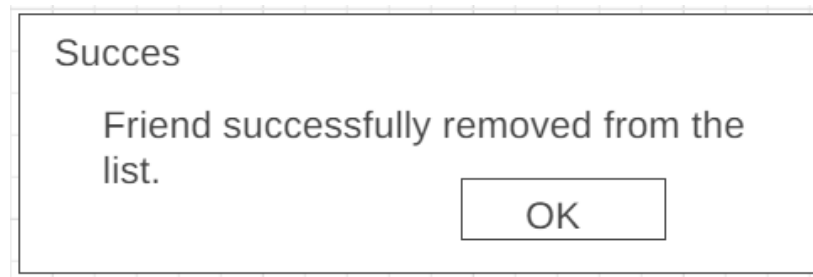


Рисунок 1.9 – Повідомлення після успішної операції

Після цього користувач повертається на головне вікно програми (рис. 1.1).

Функція 4. Перегляд інформації про доданих друзів

Функція перегляду здійснюється з головного вікна програми (рис. 1.1). На верхній панелі програми, користувач натискає кнопку “Friend” і у випадному списку (рис. 1.2) обирає пункт “View info”, попередньо обравши необхідну людину зі списку.

Програма відкриває форму з полями, в які передаються дані про обрану людину, без можливості зміни даних (рис. 1.10).

A rectangular form with a thin border. It contains two columns of text input fields. The left column has ten fields, each preceded by a thick black horizontal bar. The right column also has ten fields. The eighth field from the top in the right column is highlighted with a blue border. At the bottom center of the form is a button labeled 'OK'.

Рисунок 1.10 – Вікно перегляду інформації про друга

Натиснувши кнопку “ОК”, користувач повертається на головне вікно програми (рис. 1.1).

Функція 5. Пошук друга

Панель пошуку (рис. 1.11) знаходиться на головному вікні програми (рис. 1.1).

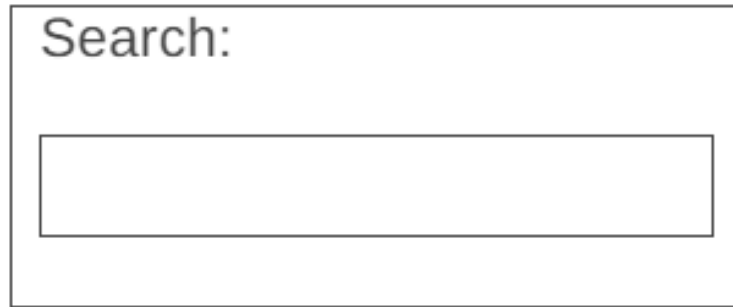


Рисунок 1.11 – Панель пошуку

Панель пошуку – текстове поле, в якому пошук здійснюється за такими критеріями:

1. Last name,
2. Name,
3. Surname,
4. Place of work/study,
5. Position,
6. Acquaintance.

Як тільки користувач починає вводити текст, програма автоматично здійснює пошук. У текстовому полі регістр неважливий.

Функція 6. Створення автоматичного привітання з днем народження

Поле для нагадування (рис. 1.12) знаходиться на головному вікні програми (рис. 1.1).

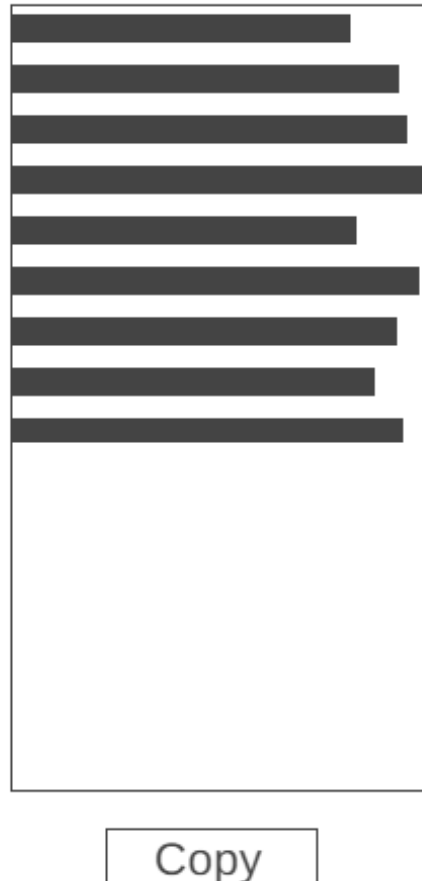


Рисунок 1.12 – Поле для нагадування

Якщо список пустий, то у текстовому полі написано "Today is not anyone's birthday. Have a great day!".

Якщо у списку є люди, у яких сьогодні день народження, у текстовому полі написано "Happy birthday, {LastName} {Name} {Surname}! I wish you happiness, health, and a lot of money!" Користувач може скопіювати текст, натиснувши на відповідну кнопку "Copy".

Текстове поле оновлюється після кожної маніпуляції за списком.

Функція 7. Сортування за алфавітом та за датою останнього коригування

Кнопки для сортування (рис. 1.13) знаходяться на головному вікні програми (рис. 1.1).

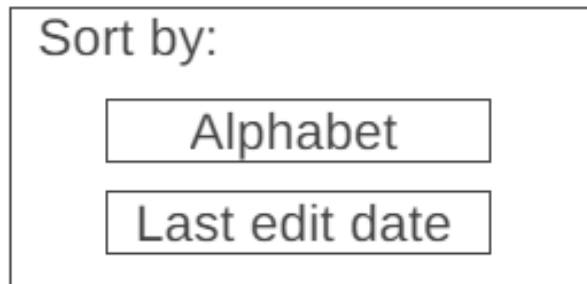


Рисунок 1.13 – Кнопки сортування списку

Якщо користувач натискає кнопку “Alphabet”, відбувається сортування по прізвищам за алфавітом.

Якщо користувач натискає кнопку “Last edit date”, відбувається сортування за датою останнього коригування. Цю дату програма отримує після додавання друга та після редагування інформації про друга.

Функція 8. Збереження даних

Функція збереження даних здійснюється з головного вікна програми (рис. 1.1). На верхній панелі програми, користувач натискає кнопку “File” і у випадному списку (рис. 1.14) обирає пункт “Save”.

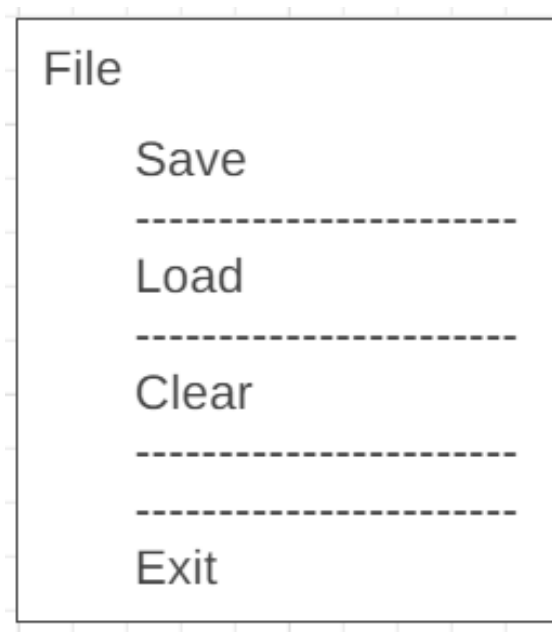


Рисунок 1.14 – Випадний список

Програма запрошує підтвердження збереження даних (рис. 1.15).

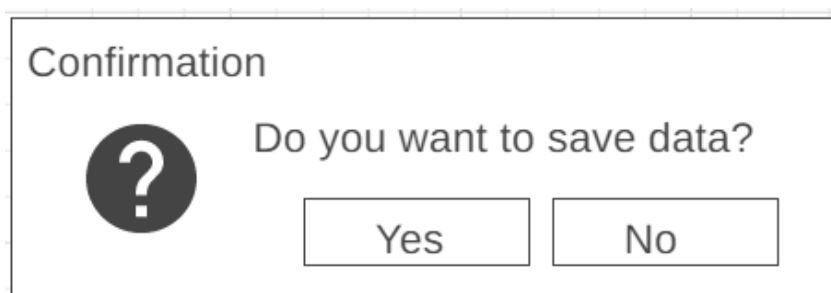


Рисунок 1.15 – Підтвердження збереження даних

Після успішної операції, програма буде виводити повідомлення (рис. 1.16).

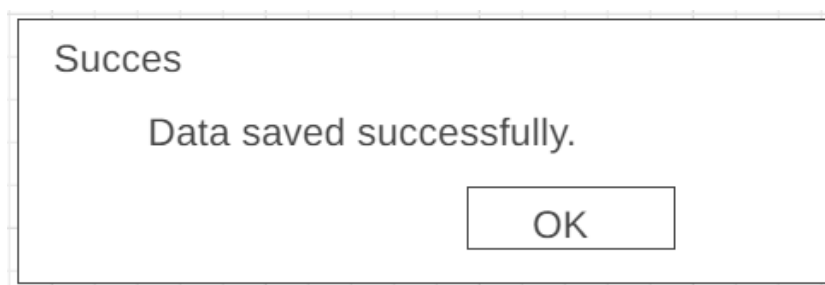


Рисунок 1.16 – Повідомлення після успішної операції

Таким чином дані будуть збережені у постійній пам'яті.

Функція 9. Завантаження даних

Функція завантаження даних здійснюється з головного вікна програми (рис. 1.1). На верхній панелі програми, користувач натискає кнопку “File” і у випадному списку (рис. 1.14) обирає пункт “Load”.

Програма запрошує підтвердження завантаження даних (рис. 1.17).

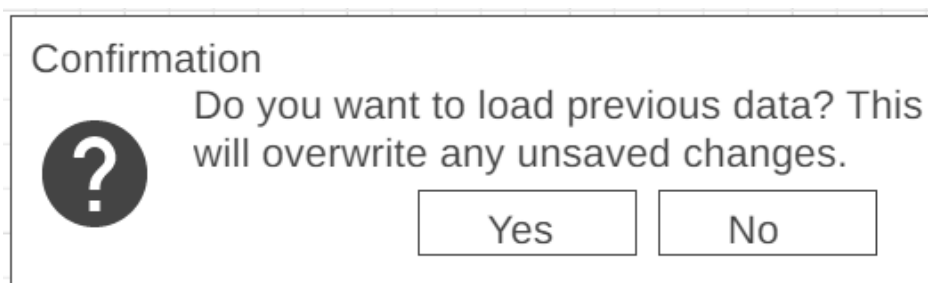


Рисунок 1.17 – Підтвердження завантаження даних

Після успішної операції, програма завантажує попередні дані, які останні були збережені у постійній пам'яті.

Функція 10. Очищення даних

Функція очищення даних здійснюється з головного вікна програми (рис. 1.1). На верхній панелі програми, користувач натискає кнопку “File” і у випадному списку (рис. 1.14) обирає пункт “Clear”.

Програма запрошує підтвердження очищення даних (рис. 1.18).

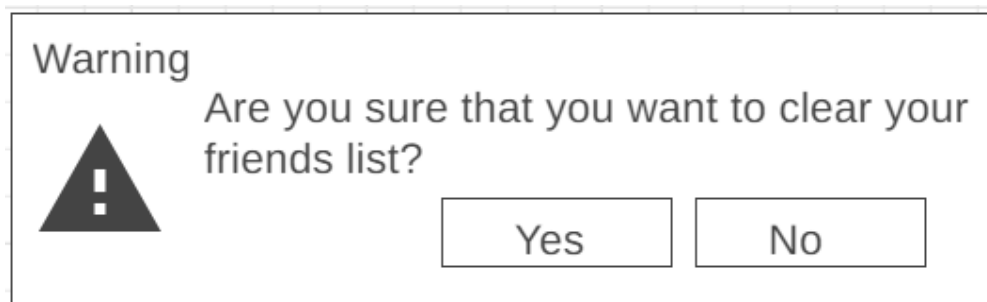


Рисунок 1.18 – Підтвердження завантаження даних

Після успішної операції, програма очищає увесь список друзів.

2 ПРОЕКТУВАННЯ ПРОГРАМИ

Для реалізації програми була обрана настільна архітектура з Graphical User Interface. Для зручності розподілили усі файли проекту на три групи: Data, Forms, Models (рис. 2.1).

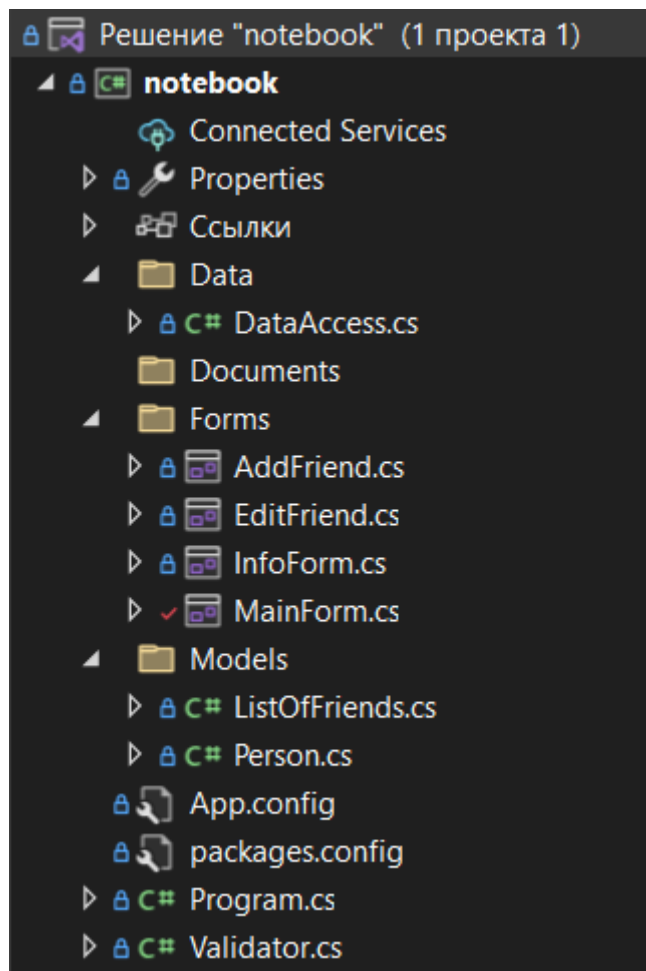


Рисунок 2.1 – Структура файлів проекту

Дані зберігатимуться у форматі JSON.

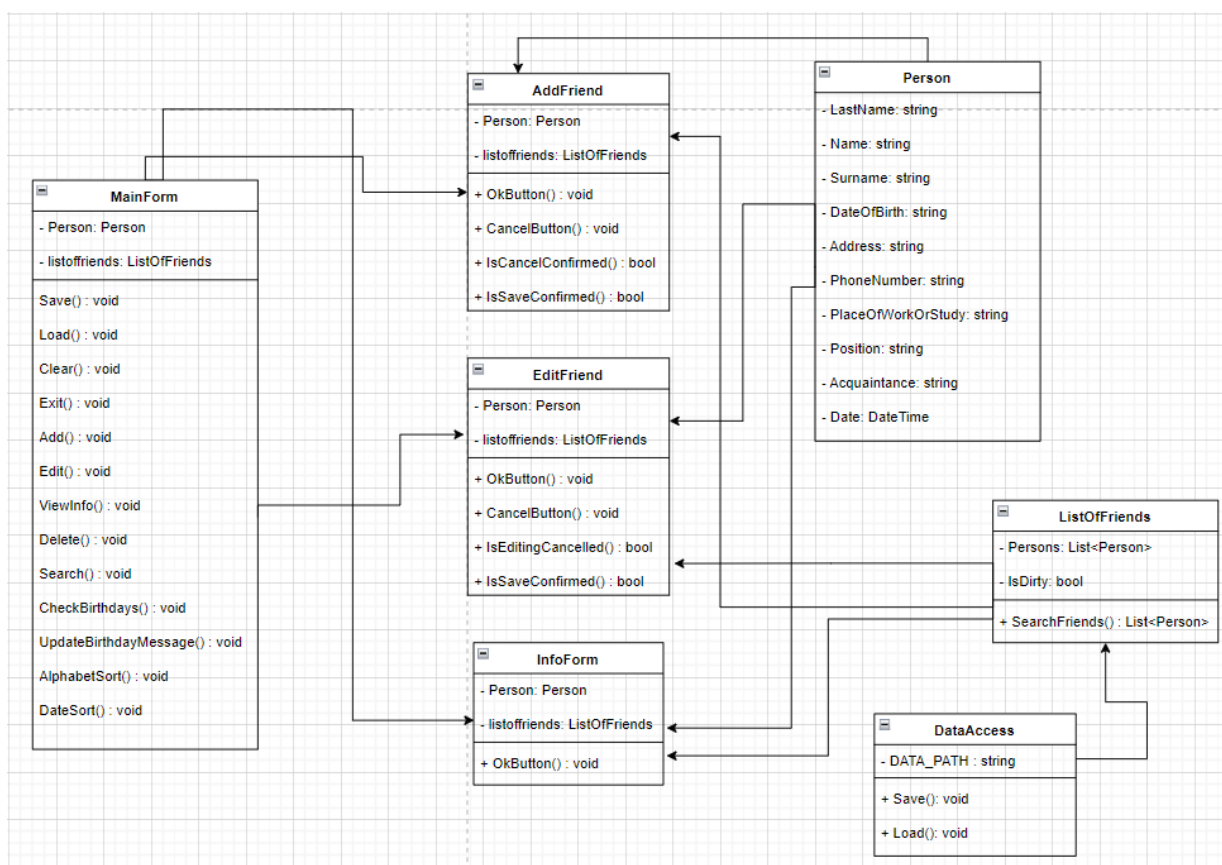


Рисунок 2.2 – Взаємодія класів у програмі

У проєкті є такі класи:

1. Клас "MainForm":

Це головне вікно програми, яке з'являється при запуску. З нього робляться усі маніпуляції зі списком.

2. Клас "AddFriend":

Це вікно відкривається з головної сторінки. Використовується для додавання нових людей до списку.

3. Клас "EditFriend":

Це вікно відкривається з головної сторінки. Використовується для редагування даних про людей зі списку.

4. Клас "InfoForm":

Це вікно відкривається з головної сторінки. Використовується для відображення інформації про людей зі списку.

5. Клас "Person":

Представляє об'єкт, що описує особу в програмі.

6. Клас "ListOfFriends":

Клас, що містить список друзів.

7. Клас "Validator":

Клас, який потрібен для перевірки правильності вводу даних користувачем.

3 ІНСТРУКЦІЯ КОРИСТУВАЧА

3.1 Встановлення програми

Процедура встановлення програми протікає таким чином:

1. Завантажити .zip.
2. Розархівувати його.
3. Далі шукаємо .exe у папках. Notebook > obj > Debug > notebook.exe.

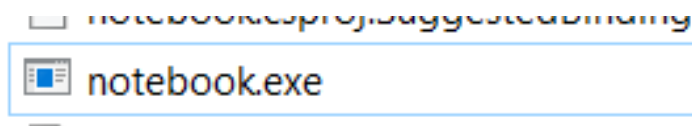


Рисунок 3.1.1 – Вигляд у папці

3.2 Видалення програми

Для цього необхідно просто видалити папку проекту з Вашого комп'ютеру.

3.3 Робота з програмою

Запускаємо програму. Відкривається головне вікно (рис. 3.3.1), з якого проводяться усі маніпуляції.

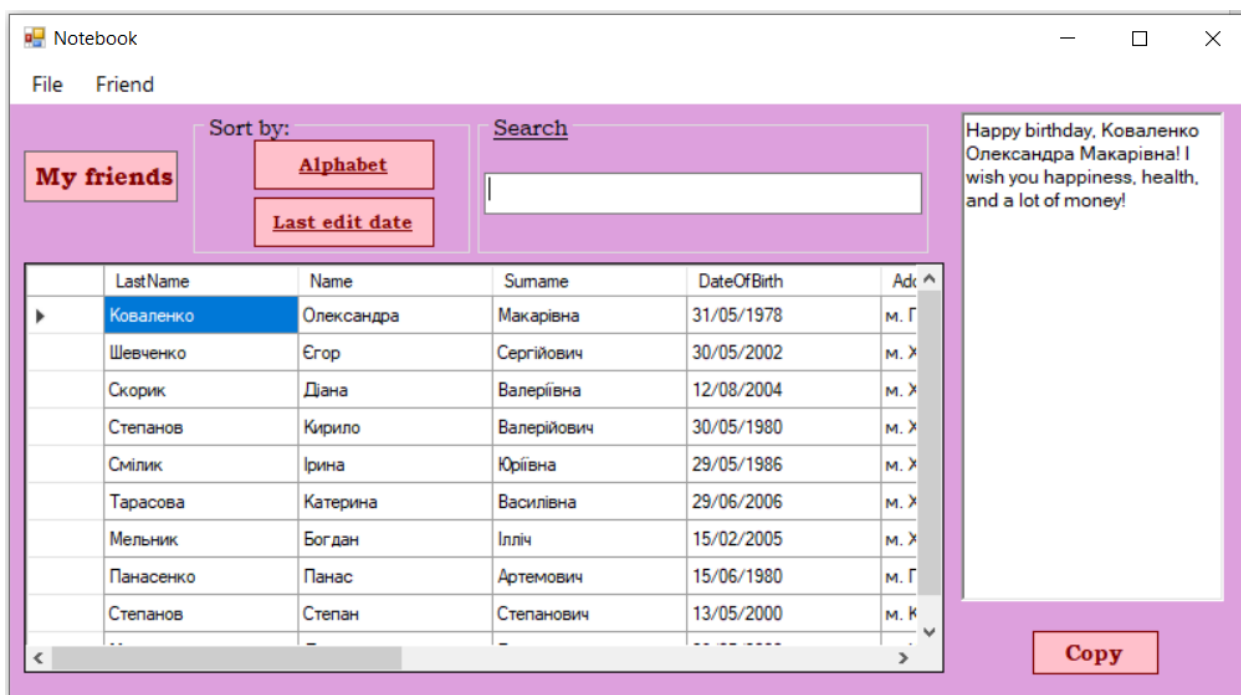


Рисунок 3.3.1 – Вигляд головного вікна програми

У верхній частині програми знаходиться панель, на якій ми можемо побачити два елементи меню: “File”, “Friend”. Елемент “File” потрібен для роботи з даними (рис. 3.3.2), “Friend” (рис. 3.3.3) – безпосередньо для роботи зі списком друзів.

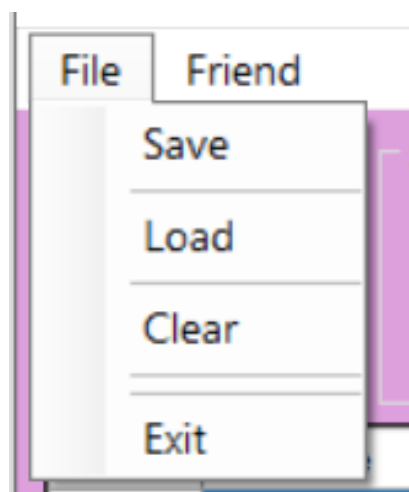


Рисунок 3.3.2 – Вигляд випадного списку елемента “File”

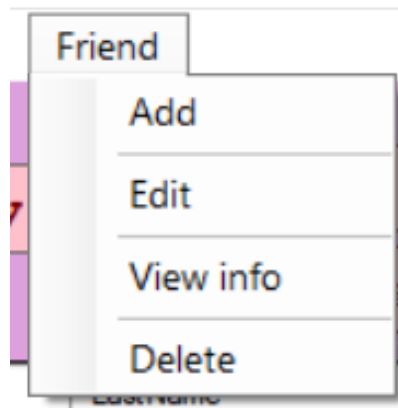


Рисунок 3.3.3 – Вигляд випадного списку елемента “Friend”

Розглянемо можливості елемента “File” (рис. 3.3.2).

Перший елемент меню – “Save”. Він потрібен для того, щоб зберегти дані.

Натиснувши на кнопку, з’являється діалогове вікно (рис. 3.3.4).

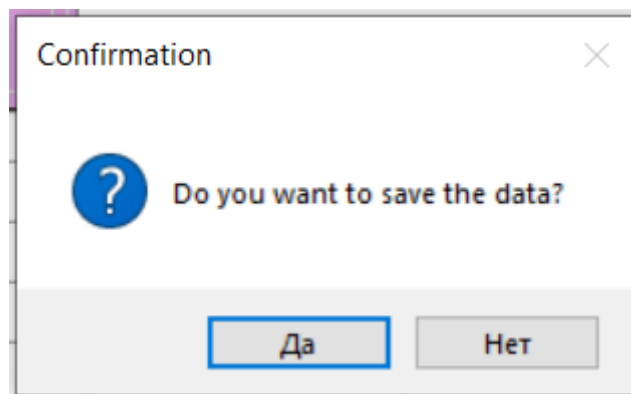


Рисунок 3.3.4 – Діалогове вікно

При підтвердженні збереження, з’являється повідомлення (рис. 3.3.5) про успішне збереження даних.

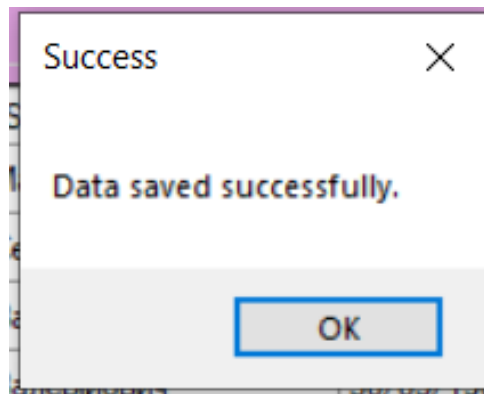


Рисунок 3.3.5 – Успішне виконання операції

Наступним елементом меню є кнопка “Load”.

За допомогою цієї кнопки ми можемо повернутися до попередніх змін.

Натиснувши на кнопку, з’являється діалогове вікно (рис. 3.3.6).

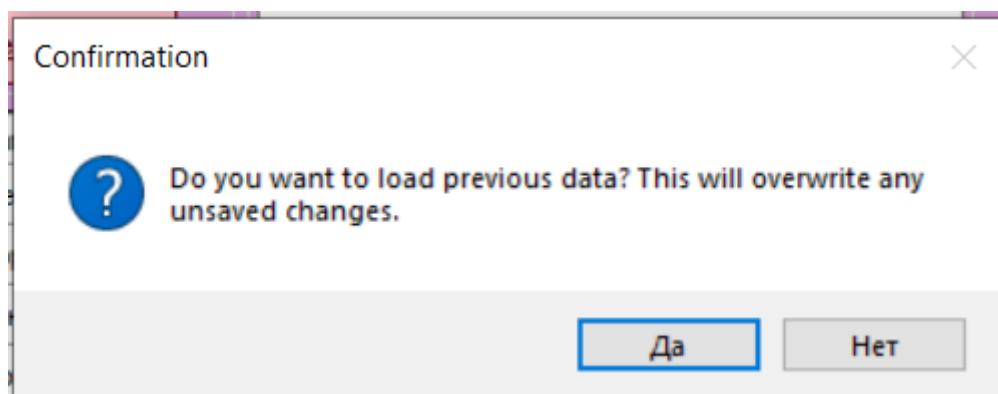


Рисунок 3.3.6 – Діалогове вікно

При підтвердженні завантаження, Ви побачите зміни, які були збережені останні.

Наступним елементом меню є кнопка “Clear”.

Натиснувши на кнопку, з’являється діалогове вікно (рис. 3.3.7).

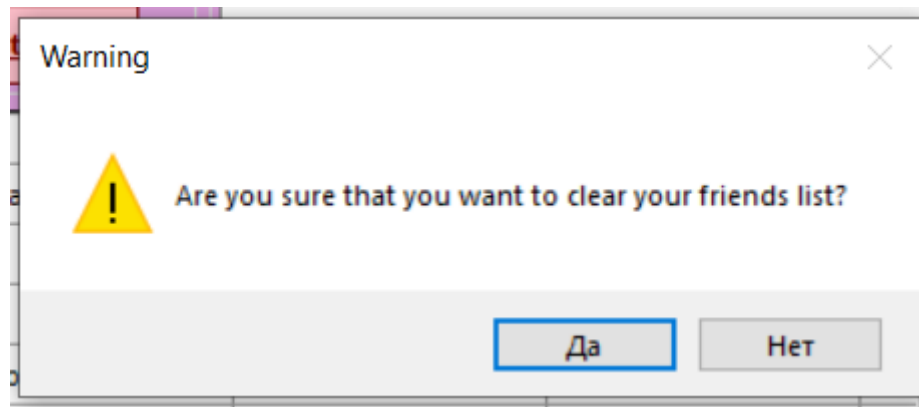


Рисунок 3.3.7 – Діалогове вікно

При підтвердженні очищення, список буде повністю очищений (рис. 3.3.8).

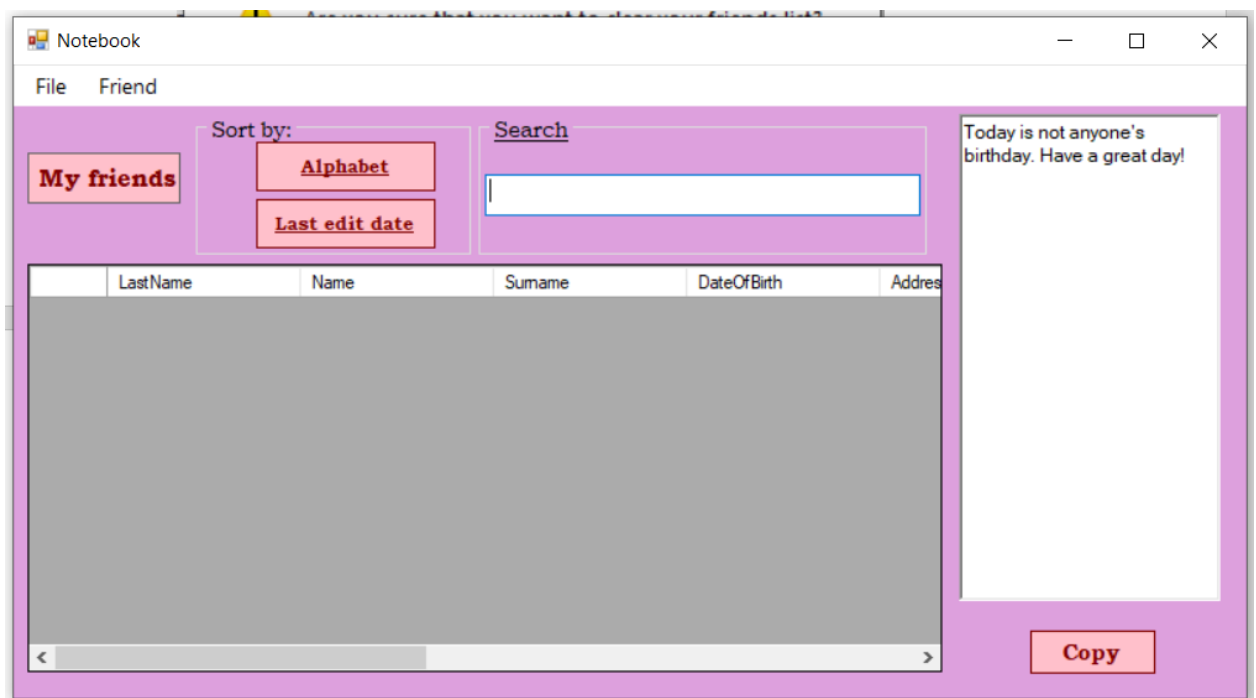


Рисунок 3.3.8 – Результат операції

Останнім елементом меню є кнопка "Exit".

Програма подбає про те, щоб Ви зберегли дані, перед тим як вийти із застосунку (рис. 3.3.9).

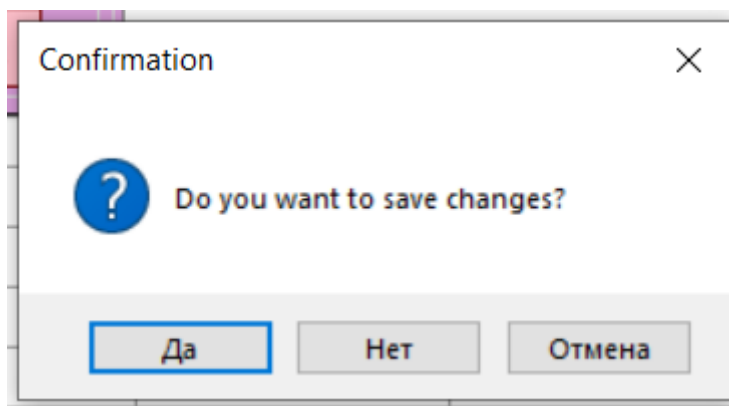


Рисунок 3.3.9 – Діалогове вікно

Далі Вас запитують, чи дійсно хочете вийти з програми (рис. 3.3.10).

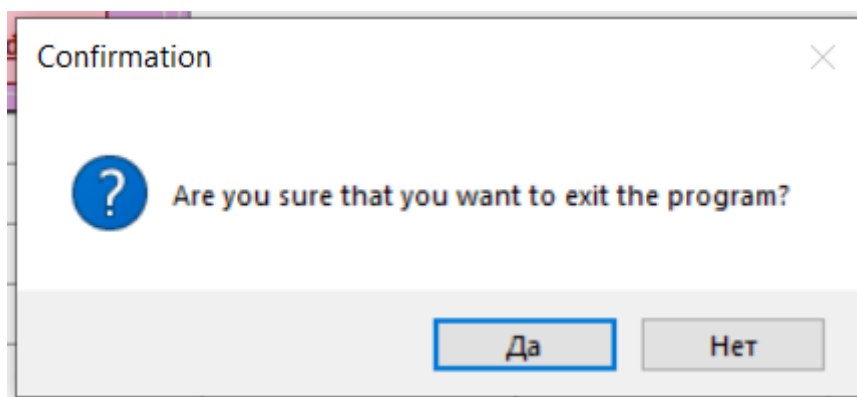
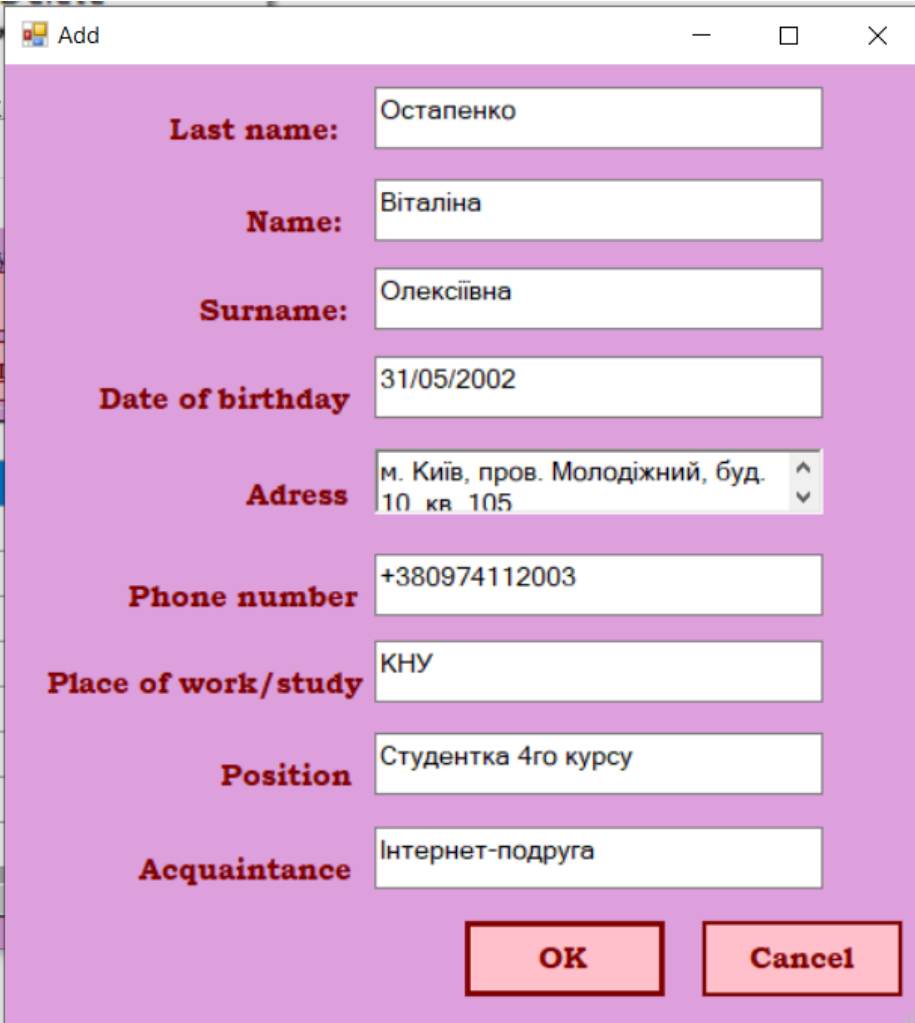


Рисунок 3.3.10 – Діалогове вікно при виході з програми

Розглянемо можливості елемента “Friend” (рис. 3.3.3).

Перший елемент меню – “Add”. За допомогою цього елемента меню ми можемо додати людину до списку друзів.

Натиснувши на дану кнопку, відкривається форма (рис. 3.3.11), яку потрібно заповнити.



Last name:	Остапенко
Name:	Віталіна
Surname:	Олексіївна
Date of birthday	31/05/2002
Adress	м. Київ, пров. Молодіжний, буд. 10 кв. 105
Phone number	+380974112003
Place of work/study	КНУ
Position	Студентка 4го курсу
Acquaintance	Інтернет-подруга

OK Cancel

Рисунок 3.3.11 – Приклад заповнення форми додавання друга

Натиснувши кнопку “OK”, програма перевірить коректність введених Вами даних, якщо буде помилка, то програма сповістить Вас (рис. 3.3.12). Якщо ж помилки немає, то з’явиться діалогове вікно для підтвердження операції (рис. 3.3.13). Якщо натиснути кнопку “Cancel”, програма запитає, чи дійсно Ви хочете покинути додавання людини до списку (рис. 3.3.14).

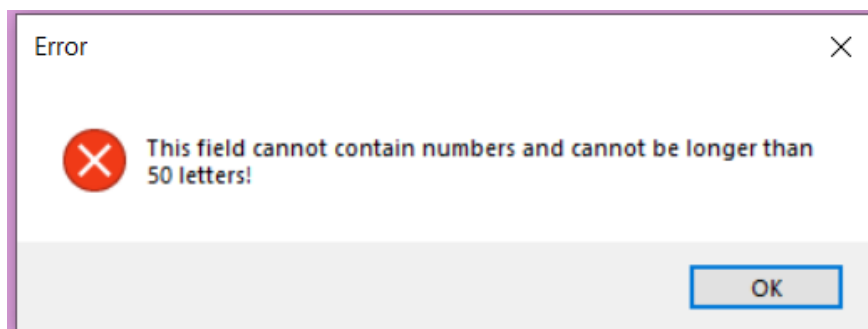


Рисунок 3.3.12 – Приклад можливої помилки

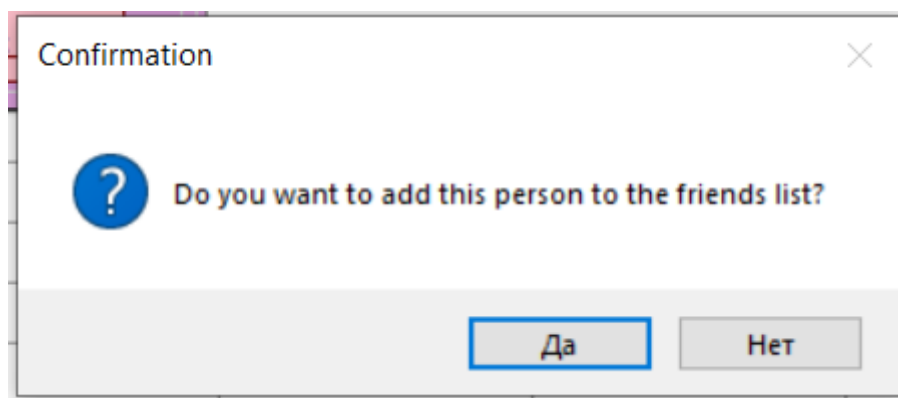


Рисунок 3.3.13 – Діалогове вікно

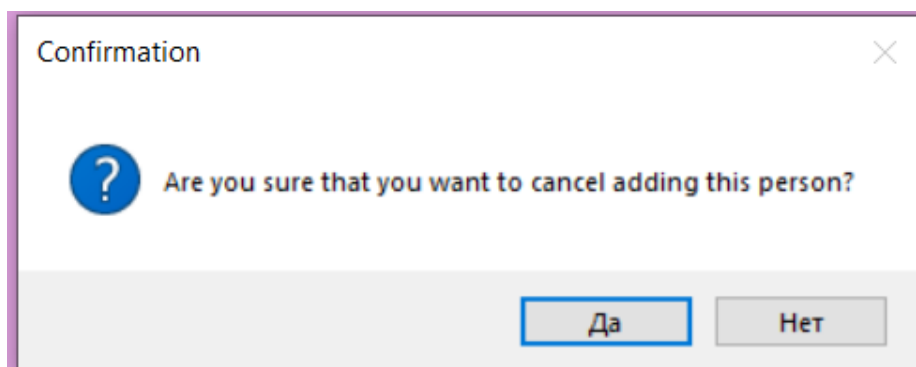


Рисунок 3.3.14 – Діалогове вікно

Якщо ми підтверджуємо додавання людини до списку (рис. 3.3.13), з'явиться повідомлення про успішне додавання. Програма оновить список (рис. 3.3.15).

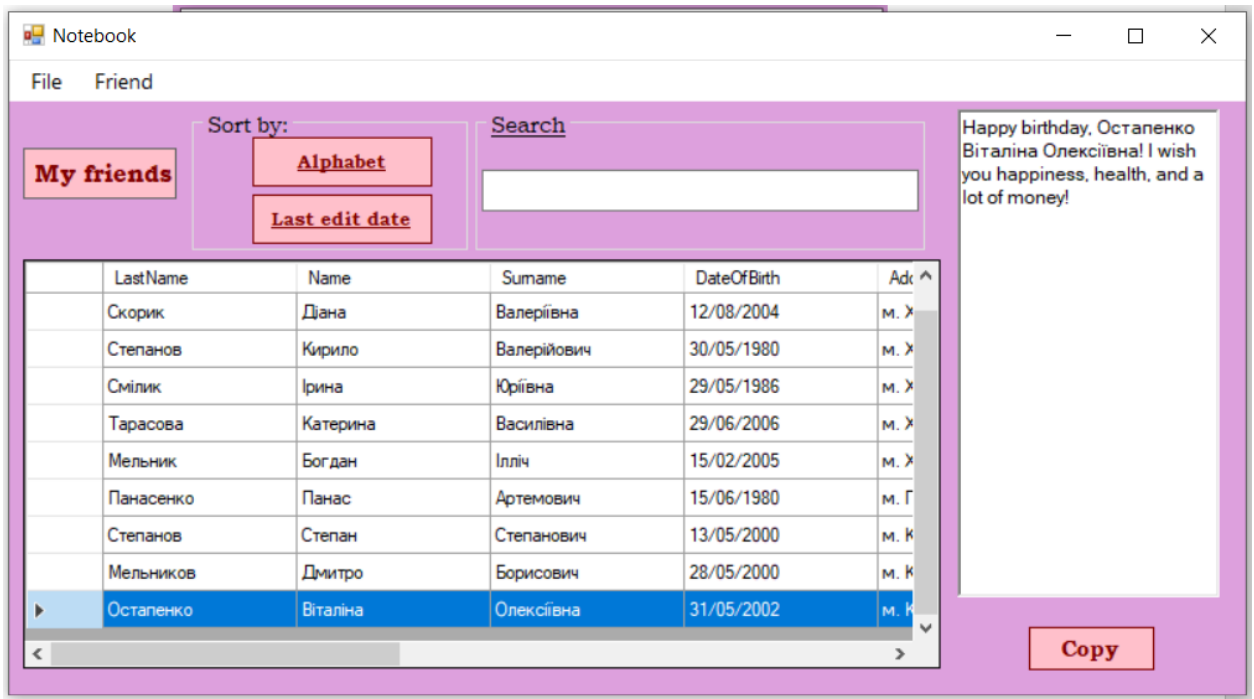


Рисунок 3.3.15 – Результат операції

До речі, програма фіксує час додавання людини до списку.

Наступним елементом меню є кнопка “Edit”. За допомогою цього елемента меню ми можемо відредагувати дані про будь-яку людину зі списку, якщо список не порожній.

Спочатку обираємо людину зі списку. Потім натискаємо на дану кнопку, відкривається форма (рис. 3.3.16), на якій ми можемо редагувати інформацію.

Edit

Last name: Остапенко

Name: Віталіна

Surname: Олексіївна

Date of birthday: 31/05/2002

Adress: м. Київ, пров. Молодіжний, буд. 10 кв. 105

Phone number: +380974112003

Place of work/study: КНУ

Position: Студентка 4го курсу

Acquaintance: Інтернет-подруга

OK **Cancel**

Рисунок 3.3.16 – Вигляд форми редагування даних про друга

Після внесення необхідних змін, натискаємо кнопку “OK”. Далі, у вікні, яке з’явилося (рис. 3.17), підтвердити бажання змінити дані.

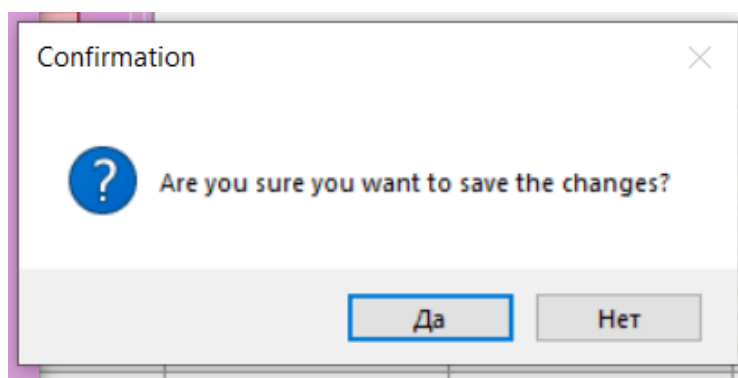


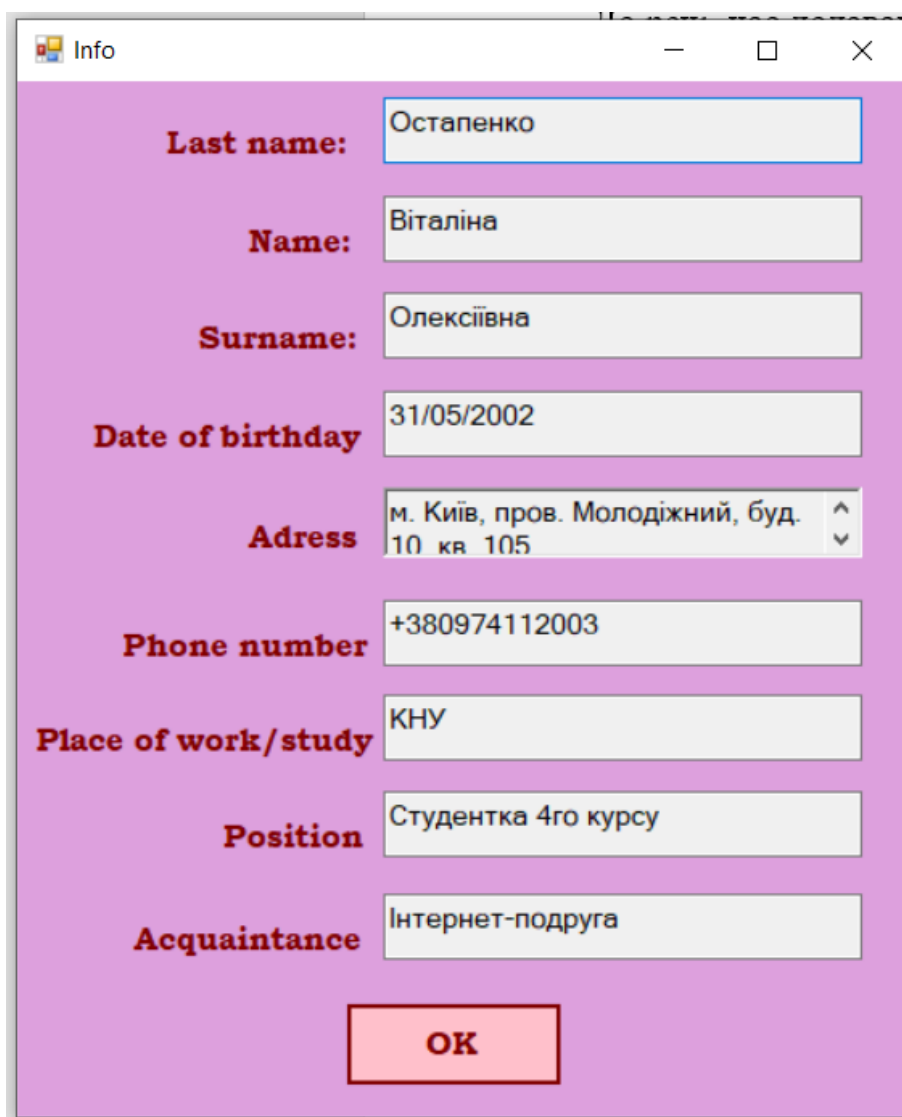
Рисунок 3.3.17 – Діалогове вікно

Якщо підтверджуємо редагування, то програма оновлює інформацію. До речі, час додавання людини теж змінюється на поточний.

Якщо натиснемо кнопку “Cancel”, програма питає, чи дійсно Ви хочете відмінити редагування даних.

Наступним елементом меню є кнопка “View info”. За допомогою цього елемента меню ми можемо переглянути дані про будь-яку особу зі списку, якщо список не пустий, у зручному вигляді.

Спочатку ми обираємо людину зі списку. Потім натискаємо на дану кнопку, відкривається вікно для перегляду (рис. 3.3.18), воно виглядає так само як і форма для редагування, але на цьому вікні ми не можемо вносити зміни.



The image shows a Windows-style dialog box titled "Info". It has a light purple background and contains several text input fields, each preceded by a label in bold red text. The labels and their corresponding values are: "Last name:" with "Остапенко", "Name:" with "Віталіна", "Surname:" with "Олексіївна", "Date of birthday" with "31/05/2002", "Adress" (misspelled) with "м. Київ, пров. Молодіжний, буд. 10 кв 105", "Phone number" with "+380974112003", "Place of work/study" with "КНУ", "Position" with "Студентка 4го курсу", and "Acquaintance" with "Інтернет-подруга". At the bottom center of the dialog is a red rectangular button with the text "OK" in black.

Field Label	Value
Last name:	Остапенко
Name:	Віталіна
Surname:	Олексіївна
Date of birthday	31/05/2002
Adress	м. Київ, пров. Молодіжний, буд. 10 кв 105
Phone number	+380974112003
Place of work/study	КНУ
Position	Студентка 4го курсу
Acquaintance	Інтернет-подруга

Рисунок 3.3.18 – Вигляд вікна для перегляду інформації

Натиснувши на кнопку “OK”, знову опиняємося на головному вікні.

Останнім елементом меню є кнопка “Delete”. За допомогою цього елемента меню ми можемо видалити будь-яку людину зі списку, якщо список не порожній.

Спочатку обираємо людину зі списку. Потім натискаємо на дану кнопку, з’являється діалогове вікно (рис. 3.3.19), на якому ми маємо підтвердити наше бажання видалити людину зі списку.

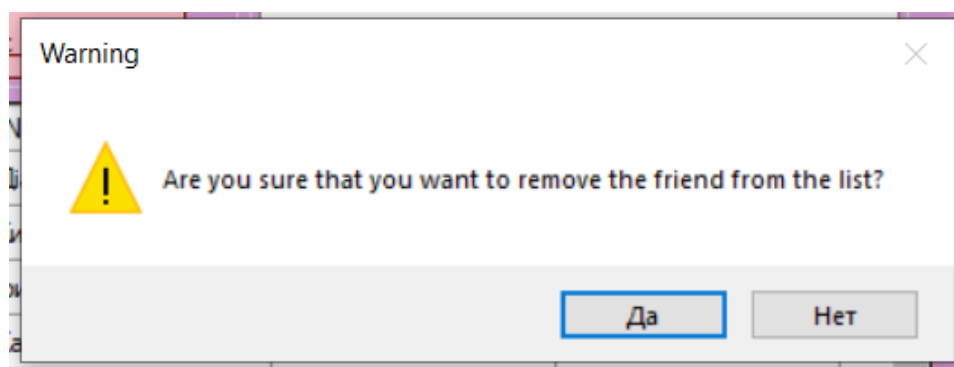


Рисунок 3.3.19 – Діалогове вікно

Якщо ми підтверджуємо видалення, програма видаляє обрану людину зі списку, з'являється повідомлення про успішне видалення і список оновлюється. Якщо не підтверджуємо, то ми повертаємося на головне вікно програми.

На головному вікні є панель "Sort by:", на якій присутні дві кнопки "Alphabet" і "Last edit date" (рис. 3.3.20). Вони потрібні нам, щоб відсортувати людей за алфавітом і за датою останнього коригування.

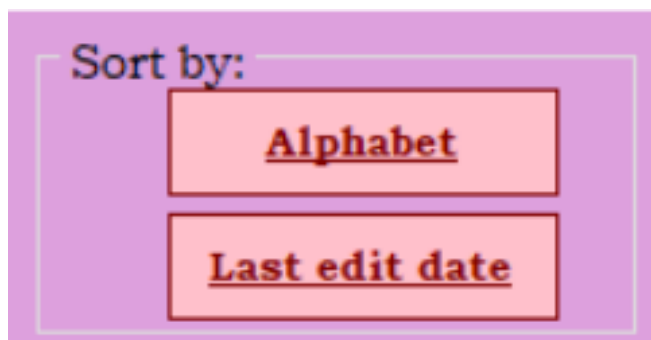


Рисунок 3.3.20 – Панель для сортування

Якщо хочемо відсортувати список за алфавітом, натискаємо верхню кнопку (рис. 3.3.20).

	LastName	Name	Surname	DateOfBirth	Adx ^
	Коваленко	Олександра	Макарівна	31/05/1978	м. Г
	Мельник	Богдан	Ілліч	15/02/2005	м. Х
	Мельников	Дмитро	Борисович	28/05/2000	м. К
	Панасенко	Панас	Артемович	15/06/1980	м. Г
	Скорик	Діана	Валеріївна	12/08/2004	м. Х
	Смілик	Ірина	Юріївна	29/05/1986	м. Х
	Степанов	Кирило	Валерійович	30/05/1980	м. Х
	Степанов	Степан	Степанович	13/05/2000	м. К
►	Тарасова	Катерина	Василівна	29/06/2006	м. Х
<	...	-	-	-----	>

Рисунок 3.3.21 – Результат сортування за алфавітом

Сортування за алфавітом відбувається по прізвищам. Якщо хочемо відсортувати список за датою, натискаємо нижню кнопку (рис. 3.3.20).

	LastName	Name	Surname	DateOfBirth	Adx ^	Date ^
	Коваленко	Олександра	Макарівна	31/05/1978	м. Г	31.05.2023 00:14
	Шевченко	Егор	Сергійович	30/05/2002	м. Х	30.05.2023 17:39
	Скорик	Діана	Валеріївна	12/08/2004	м. Х	30.05.2023 12:15
	Степанов	Кирило	Валерійович	30/05/1980	м. Х	30.05.2023 12:13
	Смілик	Ірина	Юріївна	29/05/1986	м. Х	29.05.2023 17:44
	Тарасова	Катерина	Василівна	29/06/2006	м. Х	29.05.2023 14:50
	Мельник	Богдан	Ілліч	15/02/2005	м. Х	28.05.2023 19:55
	Панасенко	Панас	Артемович	15/06/1980	м. Г	28.05.2023 19:32
►	Степанов	Степан	Степанович	13/05/2000	м. К	28.05.2023 19:13
<	...	-	-	-----	>	>

Рисунок 3.3.21 – Результат сортування за датою

На головному вікні також міститься панель для пошуку за довільним шаблоном (рис. 3.3.22). Шукати людей у списку ми можемо за такими критеріями:

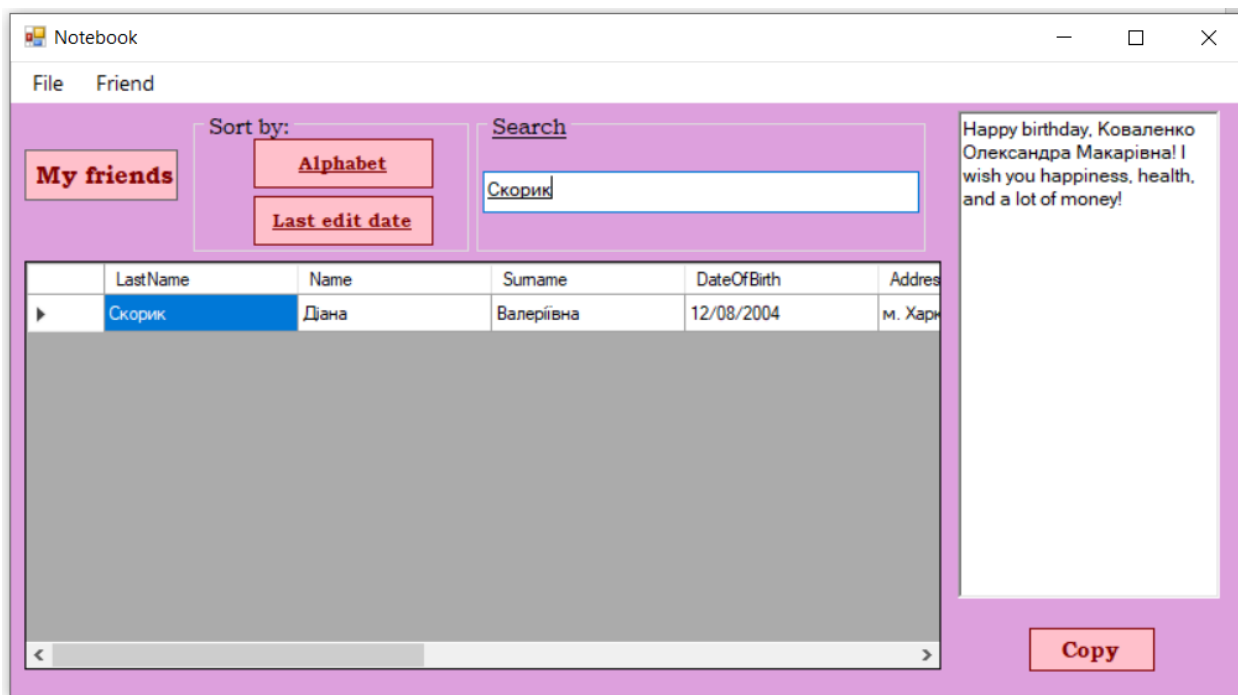
1. Last name,
2. Name,

3. Surname,
4. Place of work/study,
5. Position,
6. Acquaintance.



Рисунок 3.3.22 – Панель пошуку

Після введення параметрів пошуку, ми одразу побачимо список людей, які задовольняють умови пошуку (рис. 3.3.23), бо програма здійснює пошук автоматично.



LastName	Name	Surname	DateOfBirth	Address
Скорик	Діана	Валеріївна	12/08/2004	м. Харків

Рисунок 3.3.23 – Результат пошуку

Щоб повернутися назад до списку, нам слід прибрати текст з панелі пошуку.

І остання функція, розроблена у програмі – автоматичне формування привітання з днем народження (рис. 3.3.24).

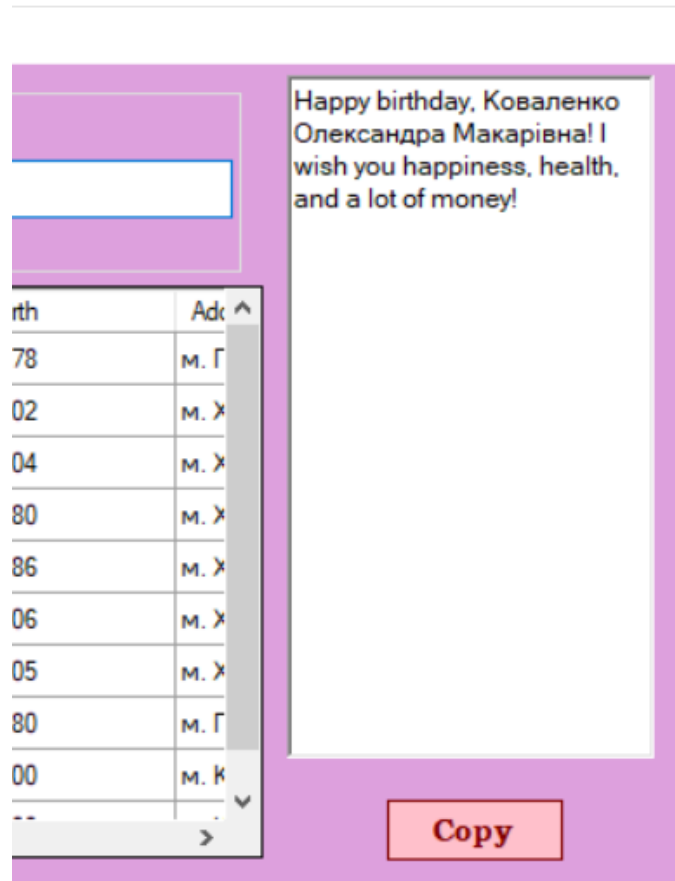


Рисунок 3.3.24 – Вигляд автоматичного привітання з днем народження

Після будь-яких змін у списку, дане поле оновлюватиметься. Оновивши дані про дні народження деяких осіб зі списку, ми одразу бачимо зміни у полі.

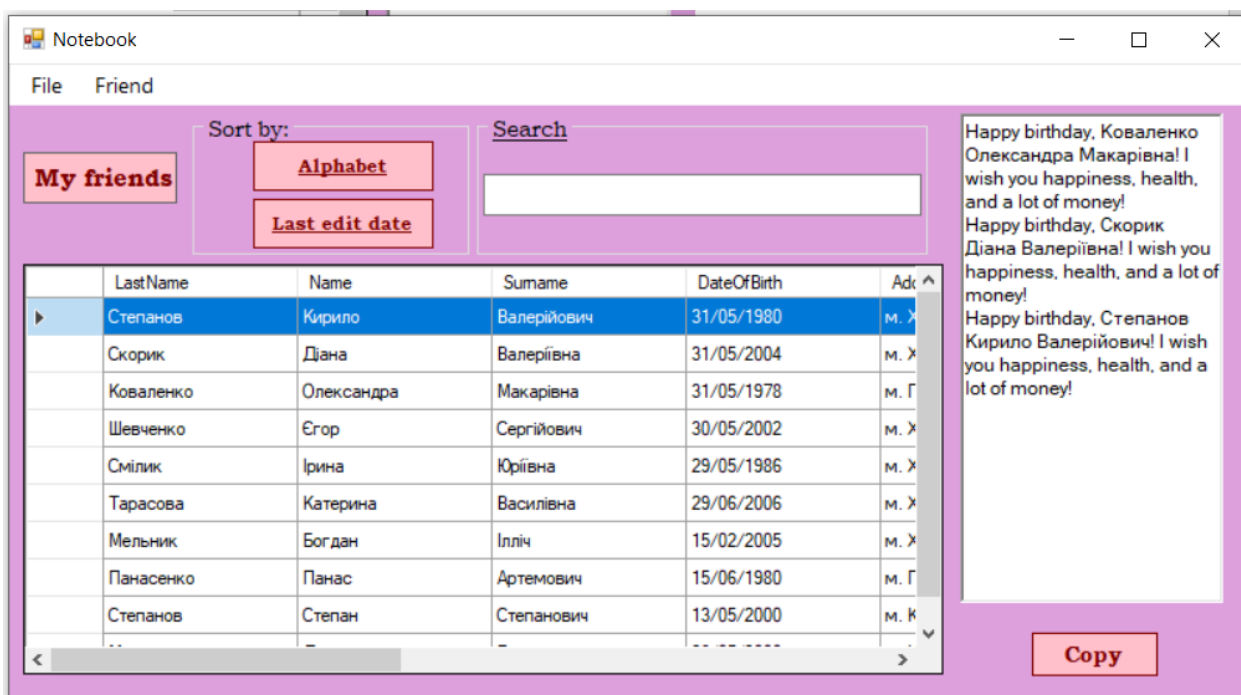


Рисунок 3.3.24 – Результат оновлення

За потреби, ми можемо натиснути кнопку “Сору” і тоді текст з поля буде скопійований до буфера обміну, з’явиться повідомлення (рис. 3.3.25).

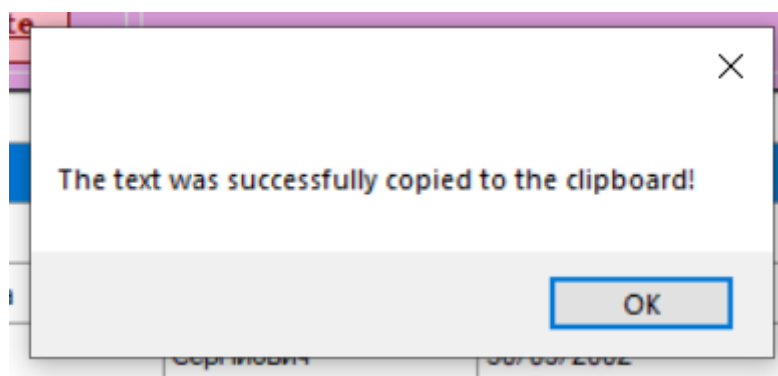


Рисунок 3.3.25 – Повідомлення про копіювання тексту привітань

Happy birthday, Степанов Кирило Валерійович! I wish you happiness, health, and a lot of money!
 Happy birthday, Скорик Діана Валеріївна! I wish you happiness, health, and a lot of money!
 Happy birthday, Коваленко Олександра Макарівна! I wish you happiness, health, and a lot of money!

Рисунок 3.3.26 – Результат копіювання

ВИСНОВКИ

Виконана програма відповідає початковим умовам. Розроблені можливості додавання, редагування, видалення, перегляд даних. Є пошук за довільним шаблоном, сортування за алфавітом і за датою останнього коригування. Також розроблено автоматичне привітання з днем народження.

Програма взаємодіє з користувачем за допомогою діалогових вікнах, які з'являються, коли користувач робить будь-які маніпуляції зі списком. Це зроблено для того, щоб не дані не губилися і не зникали безслідно, без відома користувача.

Під час розробки застосунку покращила знання C#, навчилася працювати з Windows Forms, а також ознайомила з форматом JSON, який було використано для збереження даних про друзів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бондарєв В.М. Електронний учбовий посібник з дисципліни «Основи програмування» URL: <http://tss.co.ua:5555/> (дата звернення: 30.05.2023)
2. Windows Forms documentation. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-7.0> (дата звернення: 30.05.2023)

ДОДАТОК А

Код програми

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using notebook.Models;
using notebook.Data;
using notebook.Forms;
using System.Globalization;

namespace notebook
{
    public partial class MainForm : Form
    {
        ListOfFriends listoffriends;

        public MainForm()
        {
            InitializeComponent();

            listoffriends = new ListOfFriends();
            filteredFriends = new List<Person>();

            personBindingSource.DataSource = listoffriends.Persons;

            Shown += MainForm_Shown;
        }

        private void saveToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DialogResult result = MessageBox.Show("Do you want to save the data?", "Confirmation",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (result == DialogResult.Yes)
            {
                DataAccess.Save(listoffriends);
                MessageBox.Show("Data saved successfully.", "Success", MessageBoxButtons.OK);
            }
        }

        private void loadToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DialogResult result = MessageBox.Show("Do you want to load previous data? This will overwrite any unsaved
            changes.", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (result == DialogResult.Yes)
            {
                DataAccess.Load(listoffriends);
                personBindingSource.ResetBindings(true);
                personBindingSource.DataSource = listoffriends.Persons;

                UpdateBirthdayMessage();
            }
        }
    }
}

```

```

    }

    private void clearToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Are you sure that you want to clear your friends list?", "Warning",
            MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
        {
            listoffriends.Persons.Clear();

            personBindingSource.Clear();

            personBindingSource.ResetBindings(false);

            UpdateBirthdayMessage();
        }
    }

    private void addToolStripMenuItem_Click(object sender, EventArgs e)
    {
        AddFriend form = new AddFriend(listoffriends);
        var result = form.ShowDialog();

        if (result == DialogResult.OK)
        {
            listoffriends.Persons.Add(form.Person);
            personBindingSource.ResetBindings(true);
            listoffriends.IsDirty = true;

            UpdateBirthdayMessage();
        }
    }

    private bool shouldExit;

    private void exitToolStripMenuItem_Click(object sender, EventArgs e)
    {
        DialogResult result = MessageBox.Show("Are you sure that you want to exit the program?", "Confirmation",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question);

        if (result == DialogResult.Yes)
        {
            shouldExit = true;
            Close();
        }
    }

    private void deleteToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Are you sure that you want to remove the friend from the list?", "Warning",
            MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
        {
            var selectedPerson = personBindingSource.Current as Person;
            if (selectedPerson != null)
            {
                filteredFriends.Remove(selectedPerson);
                listoffriends.Persons.Remove(selectedPerson);
                personBindingSource.ResetBindings(false);

                listoffriends.IsDirty = true;

                UpdateBirthdayMessage();

                MessageBox.Show("Friend successfully removed from the list.", "Success", MessageBoxButtons.OK);
            }
        }
    }
}

```



```

private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
    var selectedRow = friendslist.CurrentRow;

    if (selectedRow == null)
    {
        return;
    }

    var selectedPerson = selectedRow.DataBoundItem as Person;

    if (selectedPerson == null)
    {
        return;
    }

    EditFriend form = new EditFriend(selectedPerson, listoffriends);

    if (form.ShowDialog() == DialogResult.OK)
    {
        personBindingSource.ResetBindings(true);
        listoffriends.IsDirty = true;

        UpdateBirthdayMessage();
    }
}

private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
{
    if (listoffriends.IsDirty)
    {
        var saveChangesResult = MessageBox.Show("Do you want to save changes?", "Confirmation",
        MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
        switch (saveChangesResult)
        {
            case DialogResult.Yes:
                DataAccess.Save(listoffriends);
                break;
            case DialogResult.Cancel:
                e.Cancel = true;
                return;
        }
    }

    if (!shouldExit)
    {
        DialogResult exitResult = MessageBox.Show("Are you sure that you want to exit the program?",
        "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

        if (exitResult == DialogResult.No)
        {
            e.Cancel = true;
        }
    }
}

private List<Person> filteredFriends;

private void searchBox_TextChanged(object sender, EventArgs e)
{
    filteredFriends = listoffriends.SearchFriends(searchBox.Text.ToLower());
    personBindingSource.DataSource = filteredFriends;
    personBindingSource.ResetBindings(false);
}

```

```

private void MainForm_Load(object sender, EventArgs e)
{
    DataAccess.Load(listoffriends);
    personBindingSource.ResetBindings(true);
    personBindingSource.DataSource = listoffriends.Persons;

    friendslist.ClearSelection();
    friendslist.AllowUserToAddRows = false;

    birthdayTextBox.ReadOnly = true;

    UpdateBirthdayMessage();
}

public static string CheckBirthdays(ListOfFriends listOfFriends)
{
    DateTime today = DateTime.Today;
    StringBuilder sb = new StringBuilder();
    bool hasBirthdays = false;

    foreach (Person person in listOfFriends.Persons)
    {
        DateTime dateOfBirth;
        if (DateTime.TryParseExact(person.DateOfBirth, "dd/MM/yyyy", CultureInfo.InvariantCulture,
DateTimeStyles.None, out dateOfBirth))
        {
            if (dateOfBirth.Month == today.Month && dateOfBirth.Day == today.Day)
            {
                sb.AppendLine($"Happy birthday, {person.LastName} {person.Name} {person.Surname}! I wish you
happiness, health, and a lot of money!");
                hasBirthdays = true;
            }
        }
    }

    if (listOfFriends.Persons.Count == 0 || !hasBirthdays)
    {
        sb.AppendLine("Today is not anyone's birthday. Have a great day!");
    }

    return sb.ToString();
}

private void MainForm_Shown(object sender, EventArgs e)
{
    string birthdayMessage = CheckBirthdays(listoffriends);
    if (!string.IsNullOrEmpty(birthdayMessage))
    {
        Clipboard.SetText(birthdayMessage);
        birthdayTextBox.Text = birthdayMessage;
    }
}

private void copyButton_Click(object sender, EventArgs e)
{
    string birthdayMessage = CheckBirthdays(listoffriends);
    if (!string.IsNullOrEmpty(birthdayMessage))
    {
        Clipboard.SetText(birthdayMessage);
        MessageBox.Show("The text was successfully copied to the clipboard!");
    }
}

private void UpdateBirthdayMessage()
{
    string message = CheckBirthdays(listoffriends);

```

```

        birthdayTextBox.Text = message;
    }

    private void viewInfoToolStripMenuItem_Click(object sender, EventArgs e)
    {
        var selectedRow = friendslist.CurrentRow;

        if (selectedRow == null)
        {
            return;
        }

        var selectedPerson = selectedRow.DataBoundItem as Person;

        if (selectedPerson == null)
        {
            return;
        }

        InfoForm form = new InfoForm(selectedPerson, listoffriends);
        form.Show();
    }

    private void alphabetSortButton_Click(object sender, EventArgs e)
    {
        var sortedList = listoffriends.Persons.OrderBy(p => p.LastName).ToList();
        listoffriends.Persons = sortedList;
        personBindingSource.DataSource = sortedList;
        personBindingSource.ResetBindings(false);
    }

    private void dateSortButton_Click(object sender, EventArgs e)
    {
        var sortedList = listoffriends.Persons.OrderByDescending(p => p.Date).ToList();
        listoffriends.Persons = sortedList;
        personBindingSource.DataSource = sortedList;
        personBindingSource.ResetBindings(false);
    }
}

```