

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

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

ВІДДІЛ КАДРІВ

Керівник, проф.

Бондарєв В.М.

Студент гр. ПЗПІ-22-1

Тимощук Д.О.

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 студента

Тимощука Дениса Олеговича

1 Тема проекту: *Відділ кадрів*

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.

Студент

Тимошук Денис Олегович

Керівник

Бондарєв Володимир Михайлович

«31» травня 2023 року

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 47 с., 36 рис., 1 табл., 2 джерела.

СКОРОЧЕННЯ ШТАТІВ, СПИСОК ПРАЦІВНИКІВ, ЗВІЛЬНЕННЯ, ПРАЦІВНИК, МОВА ПРОГРАМУВАННЯ C#, ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПЛАТФОРМА .NET, WINDOWS FORMS.

Метою роботи є розробка програми «Відділ кадрів», яка буде надавати користувачу можливість отримувати інформацію про працівників, можливість її редагування та збереження, прийняття на роботу та звільнення працівників, а також скорочення штатів.

В результаті отримана програма, що дозволяє керувати списком працівників. Є можливості додавання працівників за певними ознаками, такими як: прізвище, ім'я, вік, паспортні данні, освіта, спеціальність, підрозділ, посада, оклад, дата надходження у фірму, дата призначення. Є можливість пошуку працівників за довільним шаблоном, перегляду інформації про кожного працівника, а також її редагування. Також, розроблено можливість скорочення вибіркового штатів зважаючи на вік працівника.

В процесі розробки використано середовища Microsoft Visual Studio 2022, фреймворк Windows Forms, платформи .NET Framework 4.8, мова програмування C#.

ЗМІСТ

ВСТУП	5
1 ОПИС ВИМОГ	6
1.1 Сценарії використання	6
1.2 Функціональні вимоги	13
2 ПРОЕКТУВАННЯ ПРОГРАМИ	24
3 ІНСТРУКЦІЯ КОРИСТУВАЧА	26
3.1 Встановлення програми	26
3.2 Видалення програми	26
3.3 Робота з програмою	26
ВИСНОВКИ	38
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	39
ДОДАТКИ	40

ВСТУП

Метою даної роботи є розробка програми з використанням фреймворку Windows Forms для обліку інформації про працівників. Призначеною функцією цього програмного продукту є зручне управління персоналом на фірмі або будь-якому виробництві, з обліком даних про працівників.

Основна мета полягає в тому, щоб забезпечити користувача зручним інструментом для збереження контактної інформації кожного працівника, такої як ім'я, паспортні дані, заробітна плата та інші деталі.

Програма "Відділ кадрів" призначена для широкого кола користувачів, починаючи від школярів і закінчуючи літніми людьми, які вміють користуватися гаджетами. Вона надає можливість додавання, редагування, видалення і перегляду даних про працівників, а також функцію пошуку за будь-яким шаблоном, скорочення штатів та перевірку правильності введених даних.

1 ОПИС ВИМОГ

1.1 Сценарії використання

Сценарій 1. Перегляд інформації про працівників

Передумова

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

Основний сценарій

1. Користувач натискає кнопку “Інформація про співробітників”.
2. Програма відкриває форму з списком усіх працівників.
3. Користувач натискає на поле пошуку.
4. Користувач у полі пошуку вводить данні необхідного йому працівника.
5. Програма оновлює список та користувач отримує усю інформацію про працівника.

Додатковий сценарій

1. Користувач натискає кнопку “Інформація про працівників”.
2. Програма відкриває форму з списком усіх працівників.
3. Користувач переглядає власноруч список працівників та знаходить необхідну людину.

4. Користувач отримує усю інформацію про працівника.

Додатковий сценарій

1. Користувач натискає кнопку “Інформація про працівників”.
2. Програма відкриває форму з списком усіх працівників.
3. Користувач натискає на поле пошуку.
4. Користувач у полі пошуку вводить часткові данні працівника.

5. Програма оновлює список та користувач отримує усю інформацію про працівника.

Сценарій 2. Редагування інформації про працівника

Передумова

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

Основний сценарій

1. Користувач обирає працівника зі списку, дані якої хоче змінити, та натискає кнопку “Редагувати”.
2. Програма відкриває форму для редагування інформації.
3. Користувач редагує необхідні дані. Натискає кнопку “Зберегти”.
4. Виводиться діалогове вікно і запрошує підтвердження редагування даних.
5. Користувач підтверджує редагування.
6. Програма редагує дані, оновлює список працівників.

Додатковий сценарій

1. Користувач обирає працівника зі списку, дані якої хоче змінити, та натискає кнопку “Редагувати”.
2. Програма відкриває форму для редагування інформації.
3. Користувач редагує необхідні дані. Натискає кнопку “Зберегти”.
4. Програма перевіряє введену користувачем інформацію, знаходить помилки і виводить користувачу повідомлення про них.

Додатковий сценарій

1. Користувач обирає працівника зі списку, дані якої хоче змінити, та натискає кнопку “Редагувати”.

2. Програма відкриває форму для редагування інформації.
3. Користувач редагує необхідні дані. Натискає кнопку “Зберегти”.
4. Виводиться діалогове вікно і запрошує підтвердження для збереження змін.
5. Користувач у діалоговому вікні натискає “НІ”.
6. Програма закриває діалогове вікно та не виконує ніяких змін.

Сценарій 3. Звільнення працівника

Передумова

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

Основний сценарій

1. Користувач натискає кнопку “Підготувати наказ”
2. Програма завантажує форму із варіантами наказів.
3. Користувач натискає кнопку “Звільнення”.
4. Програма завантажує форму зі списком усіх працівників.
5. Користувач натискає у поле пошуку та вводить інформацію про працівника якого необхідно звільнити.
6. Програма виконує пошук та оновлює список.
7. Користувач обирає зі списку працівника якого необхідно звільнити. Натискає кнопку “Звільнити”.
8. Програма виводить діалогове вікно із підтвердженням звільнення працівника.
9. Користувач підтверджує звільнення працівника натискаючи кнопку “Так”.
10. Програма видаляє усі данні працівника та оновлює список працівників.

Додатковий сценарій

1. Користувач натискає кнопку “Підготувати наказ”
2. Програма завантажує форму із варіантами наказів.
3. Користувач натискає кнопку “Звільнення”.
4. Програма завантажує форму зі списком усіх працівників.
5. Користувач відразу обирає зі списку працівника якого необхідно звільнити. Натискає кнопку “Звільнити”.
6. Програма виводить діалогове вікно із підтвердженням звільнення працівника.
7. Користувач підтверджує звільнення працівника натискаючи кнопку “Так”.
8. Програма видаляє усі данні працівника та оновлює список працівників.

Додатковий сценарій

1. Користувач натискає кнопку “Підготувати наказ”
2. Програма завантажує форму із варіантами наказів.
3. Користувач натискає кнопку “Звільнення”.
4. Програма завантажує форму зі списком усіх працівників.
5. Користувач натискає у поле пошуку та вводить інформацію про працівника якого необхідно звільнити.
6. Програма виконує пошук та оновлює список.
7. Користувач обирає зі списку працівника якого необхідно звільнити. Натискає кнопку “Звільнити”.
8. Програма виводить діалогове вікно із підтвердженням звільнення працівника.

9. Користувач не підтверджує звільнення працівника натискаючи кнопку “Ні”.

10. Програма повертає форму зі списком працівників.

Сценарій 4. Прийом працівника на роботу

Передумова

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

Основний сценарій

1. Користувач натискає кнопку “Підготувати наказ”
2. Програма завантажує форму із варіантами наказів.
3. Користувач натискає кнопку “Найм”.
4. Програма завантажує форму зі полями для заповнення інформації про працівника.
5. Користувач натискає кнопку “Зберегти”.
6. Програма виводить діалогове вікно із підтвердженням найму працівника.
7. Користувач підтверджує дію натискаючи кнопку “Так”.
8. Програма виводить діалогове вікно із підтвердженням звільнення працівника.
9. Користувач підтверджує звільнення працівника натискаючи кнопку “Так”.

10. Програма видалляє усі данні працівника та оновлює список працівників.

Додатковий сценарій

1. Користувач натискає кнопку “Підготувати наказ”
2. Програма завантажує форму із варіантами наказів.
3. Користувач натискає кнопку “Найм”.

4. Програма завантажує форму зі списком усіх працівників.
5. Користувач відразу із завантаженого списку обирає працівника якого необхідно звільнити. Натискає кнопку “Зберегти”.
6. Програма виводить діалогове вікно із підтвердженням звільнення працівника.
7. Користувач підтверджує звільнення працівника натискаючи кнопку “Так”.
8. Програма видаляє усі данні працівника та оновлює список працівників.

Додатковий сценарій

1. Користувач натискає кнопку “Підготувати наказ”
2. Програма завантажує форму із варіантами наказів.
3. Користувач натискає кнопку “Найм”.
4. Програма завантажує форму зі полями для заповнення інформації про працівника.
5. Користувач натискає кнопку “Зберегти”.
6. Програма виводить діалогове вікно із підтвердженням найму працівника.
7. Користувач підтверджує дію натискаючи кнопку “Так”.
8. Програма виводить діалогове вікно із підтвердженням звільнення працівника.
9. Користувач не підтверджує звільнення працівника натискаючи кнопку “Ні”.
10. Програма повертає форму зі введеними користувачем даними.

Сценарій 5. Скорочення штатів пенсійного та передпенсійного віку

Передумова

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

Основний сценарій

1. Користувач натискає кнопку “Підготувати наказ”
2. Програма завантажує форму із варіантами наказів.
3. Користувач натискає кнопку “Скорочення штатів пенсійного та передпенсійного віку”.
4. Програма завантажує список працівників на форму.
5. Користувач у полі віку вводить максимальний вік працівників.
6. Програма оновлює список та виводить на екран усіх працівників що старші заданого віку.
7. Користувач обирає декілька або одного працівника .Натискає кнопку “Скоротити”.
8. Програма виводить діалогове вікно із підтвердженням скорочення працівників.
9. Користувач підтверджує скорочення працівників натискаючи кнопку “Так”.
10. Програма видаляє усі данні працівників та оновлює список.

Додатковий сценарій

1. Користувач натискає кнопку “Підготувати наказ”
2. Програма завантажує форму із варіантами наказів.
3. Користувач натискає кнопку “Скорочення штатів пенсійного та передпенсійного віку”.
4. Програма завантажує список працівників на форму.
5. Користувач у полі віку вводить максимальний вік працівників.

6. Програма оновлює список та виводить на екран усіх працівників що старші заданого віку.

7. Користувач обирає декілька або одного працівника. Натискає кнопку “Скоротити”.

8. Програма виводить діалогове вікно із підтвердженням скорочення працівників.

9. Користувач не підтверджує скорочення працівників натискаючи кнопку “Ні”.

10. Програма повертає форму зі списком працівників.

1.2 Функціональні вимоги

Функція 1. Перегляд інформації про працівників

Головне вікно програми із якого відбувається навігація (рис. 1.1).

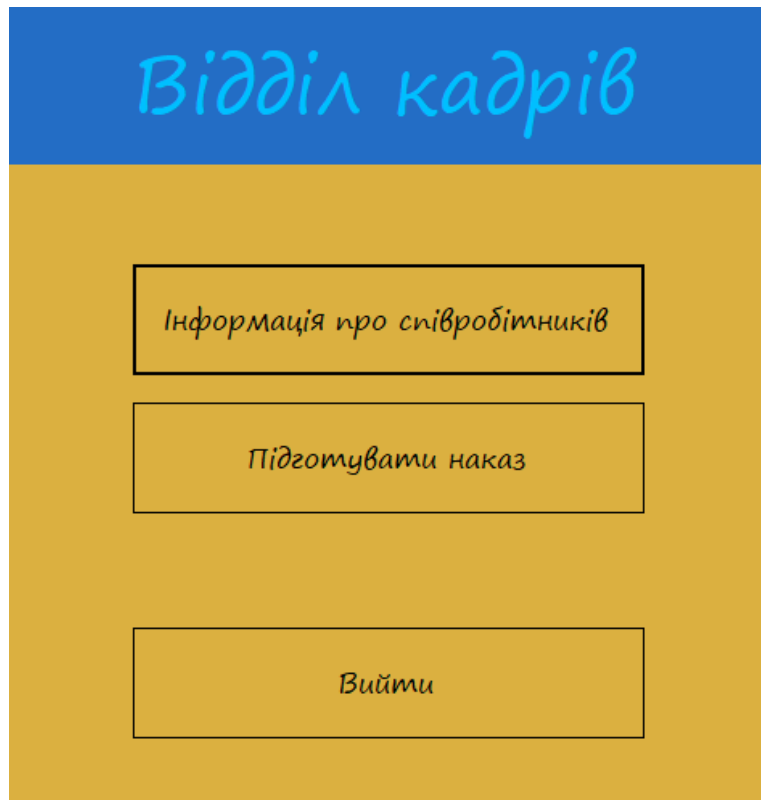


Рисунок 1.1 – Головне вікно програми

Функція перегляду здійснюється з допоміжного вікна програми (рис. 1.2).

Відділ кадрів

Прізвище:	Ім'я:	Вік:	Номер паспорту:	Освіта:	Спеціальність	Оклад:	Посада:	Входження у фірму:	Дата призначення
Петров	Іван	30	12345	Середня	Розробк...	5000	Розробн...	01.05.2015	15.01.2022
Сміт	Жанна	28	67890	Середня	Маркетинг	4000	Маркети...	10.08.2016	03.12.2021
Джонсон	Давид	35	54321	Середня	Біохімія	6000	Науковець	15.03.2014	20.05.2023
Браун	Емілія	32	98765	Середня	Графічні...	4500	Графічні...	05.10.2017	08.09.2022
Вільсон	Михайло	40	13579	Бакалав...	Фінанси	5500	Фінансо...	01.12.2010	25.07.2021
Тейлор	Олівія	27	24680	Бакалав...	Управлін...	3800	Асистент...	12.04.2019	18.02.2023
Кларк	Вільям	33	86420	Професор	Продажі	4200	Предста...	20.07.2013	10.11.2022
Андерсон	Софія	29	11111	Магістра...	Аналіз да...	4800	Аналітик ...	08.02.2016	27.03.2023
Томас	Яків	31	99999	Середня	Управлін...	5200	Менедже...	18.09.2014	05.08.2022
Робертс	Ава	26	77777	Магістра...	Публічні ...	4000	Спеціалі...	10.01.2020	30.04.2023

Пошук

Редагувати

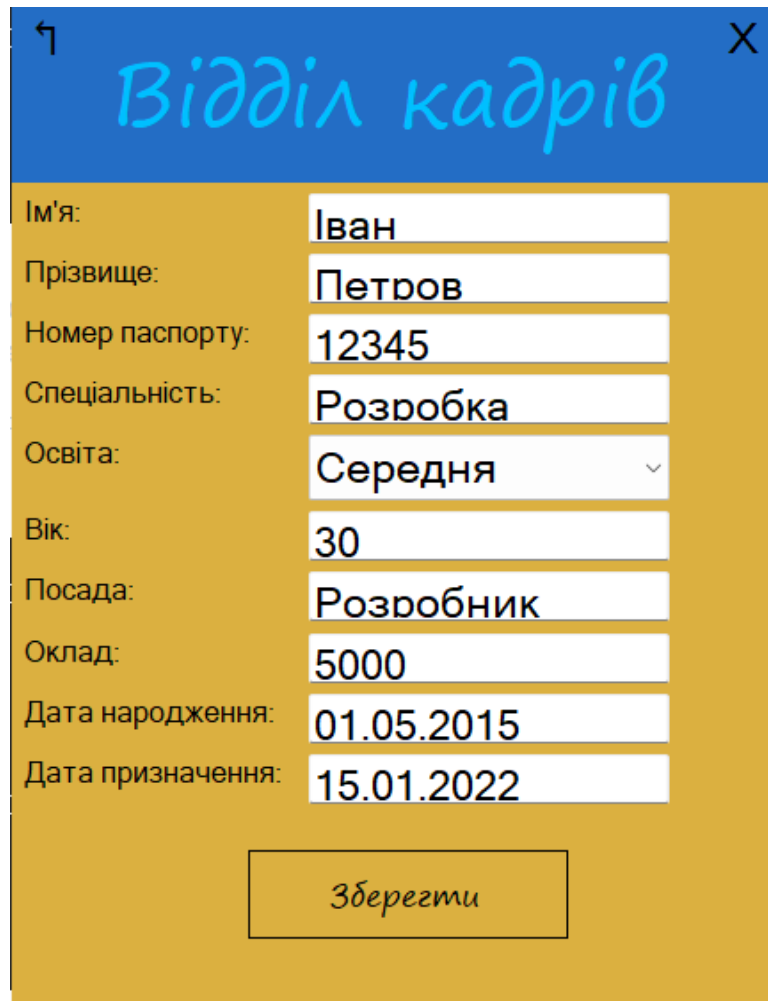
Рисунок 1.2 – Допоміжне вікно програми

Інформація про працівників безпосередньо оновлюється під час роботи програми.

Функція 2. Редагування інформації про працівника

Функція редагування здійснюється з допоміжного вікна програми (рис. 1.2). У відкритому допоміжному вікні користувач обирає працівника якого бажає редагувати та натискає кнопку “Редагувати”.

Програма відкриває форму з полями, в які передаються дані про обрану людину, з можливістю зміни даних (рис. 1.3).



Ім'я:	Іван
Прізвище:	Петров
Номер паспорту:	12345
Спеціальність:	Розробка
Освіта:	Середня
Вік:	30
Посада:	Розробник
Оклад:	5000
Дата народження:	01.05.2015
Дата призначення:	15.01.2022

Зберегти

Рисунок 1.3 – Вікно редагування працівника

Щоб редагувати дані про людину, користувач може змінювати будь-яке з 10 полів:

1. Ім'я
2. Прізвище,
3. Номер паспорту,
4. Вік,
5. Освіту,
6. Спеціальність,
7. Посаду,
8. Оклад,

9. Дата надходження у фірму,

10. Дата призначення у фірму.

Усі поля, окрім номера паспорту, освіти, віку, окладу, дати надходження у фірму та дати призначення у фірму, текстові. У полях прізвища, ім'я 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, також дату можна увести через крапку: DD.MM.YYYY. Усі поля є обов'язковими.

Користувач натискає кнопку “Зберегти”. Виводиться діалогове вікно із підтвердженням збереження змін (рис. 1.4).

Якщо користувач робить некоректний ввід даних, то програма про це сповіщає і виводить текст помилки (рис. 1.5).

Після чого програма автоматично оновлює увесь список та повертає допоміжне вікно (рис. 1.2).

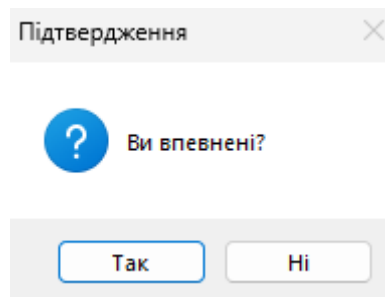


Рисунок 1.4 – Вікно підтвердження

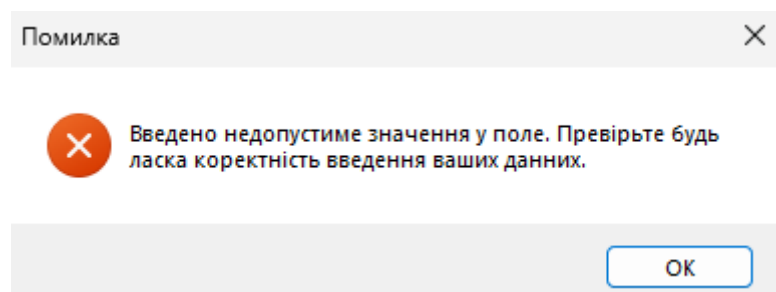


Рисунок 1.5 – Повідомлення про помилку

Функція 3. Звільнення працівника

Функція звільнення здійснюється з вікна звільнення (рис. 1.7). Користувачу із головного меню(рис. 1.1) необхідно натиснути кнопку “Підготувати наказ”. Після чого буде відкрито вікно підготовки наказів (рис. 1.6). Далі користувач вибирає кнопку “Звільнення та переходить у нове вікно ”

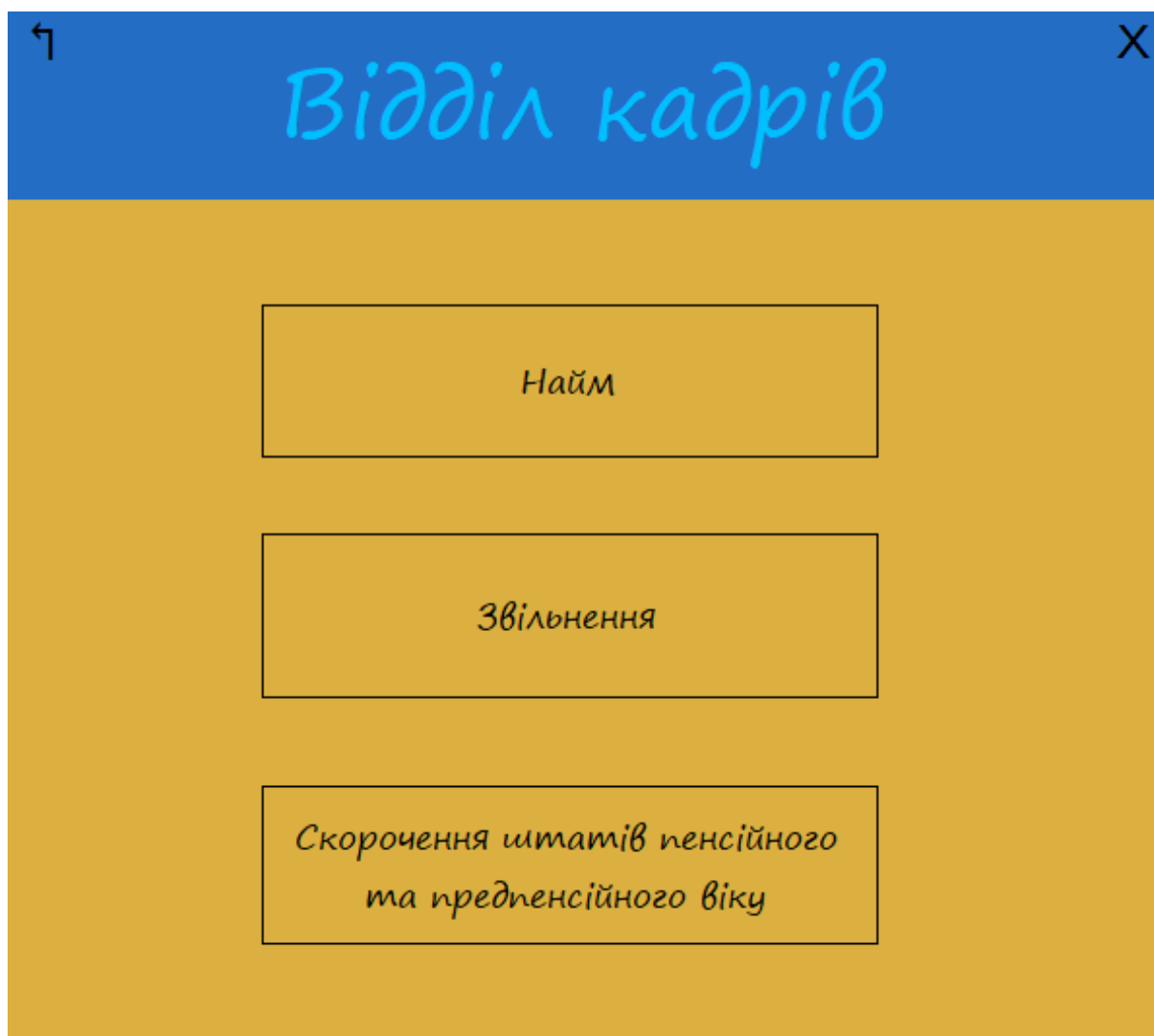
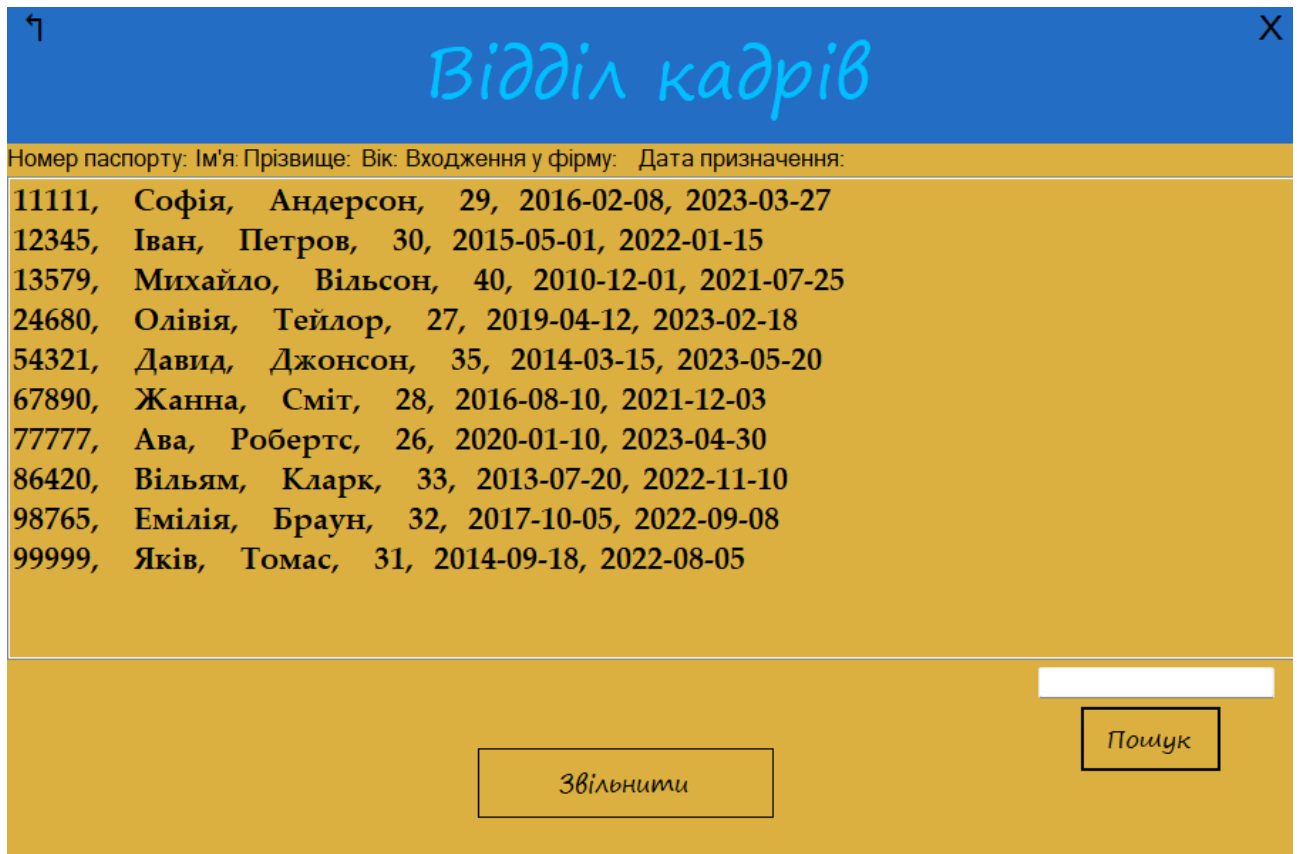


Рисунок 1.6 – Вікно підготовки наказів



Відділ кадрів

Номер паспорту: Ім'я Прізвище: Вік: Вхідження у фірму: Дата призначення:

11111,	Софія,	Андерсон,	29,	2016-02-08,	2023-03-27
12345,	Іван,	Петров,	30,	2015-05-01,	2022-01-15
13579,	Михайло,	Вільсон,	40,	2010-12-01,	2021-07-25
24680,	Олівія,	Тейлор,	27,	2019-04-12,	2023-02-18
54321,	Давид,	Джонсон,	35,	2014-03-15,	2023-05-20
67890,	Жанна,	Сміт,	28,	2016-08-10,	2021-12-03
77777,	Ава,	Робертс,	26,	2020-01-10,	2023-04-30
86420,	Вільям,	Кларк,	33,	2013-07-20,	2022-11-10
98765,	Емілія,	Браун,	32,	2017-10-05,	2022-09-08
99999,	Яків,	Томас,	31,	2014-09-18,	2022-08-05

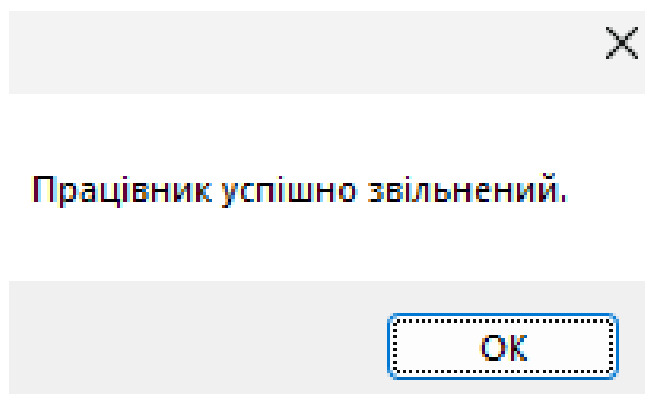
Звільнити

Пошук

Рисунок 1.7 – Вікно звільнення працівників

Програма запрошує підтвердження звільнення (рис. 1.4).

Після підтвердження буде виведене вікно у якому буде підтверджено звільнення працівника(рис. 1.8)



Працівник успішно звільнений.

OK

Рисунок 1.8 – Вікно підтвердження звільнення працівника

Після чого користувач залишається у вікні звільнення програми де може продовжувати звільнення інших працівників.

Функція 4. Прийом працівника на роботу

Функція звільнення здійснюється з вікна найму(рис. 1.9). Користувачу із головного меню(рис. 1.1) необхідно натиснути кнопку “Підготувати наказ”. Після чого буде відкрито вікно підготовки наказів (рис. 1.6). Далі користувач вибирає кнопку “Найм” та переходить у вікно редагування(рис. 1.9), при цьому усі форми не будуть заповненні жодним текстом.

Відділ кадрів

Ім'я:

Прізвище:

Номер паспорту:

Спеціальність:

Освіта:

Вік:

Посада:

Оклад:

Дата народження:

Дата призначення:

Зберегти

Рисунок 1.9 – Вікно найму працівника

Таким чином програма витрачає у двічі менше ресурсів виконуючи два накази у єдиній формі.

Щоб найняти працівника потрібно надати дані про людину в усі 10 полів:

1. Ім'я
2. Прізвище,
3. Номер паспорту,
4. Вік,
5. Освіту,
6. Спеціальність,
7. Посаду,
8. Оклад,
9. Дата надходження у фірму,
10. Дата призначення у фірму.

Усі поля, окрім номера паспорту, освіти, віку, окладу, дати надходження у фірму та дати призначення у фірму, текстові. У полях прізвища, ім'я 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, також дату можна увести через крапку: DD.MM.YYYY. Усі поля є обов'язковими.

Програма запрошує підтвердження найму працівника (рис. 1.4).

Після чого програма додасть нового працівника у список та залишиться у відкритому вікні для можливості додавання більшої кількості працівників.

Функція 5. Скорочення штатів пенсійного та передпенсійного віку

Функція скорочення штатів здійснюється з вікна скорочення(рис. 1.10). Користувачу із головного меню(рис. 1.1) необхідно натиснути кнопку “Підготувати наказ”. Після чого буде відкрито вікно підготовки наказів (рис. 1.6). Далі користувач вибирає кнопку “Скорочення штатів пенсійного та передпенсійного віку” та переходить у вікно скорочення(рис. 1.10).

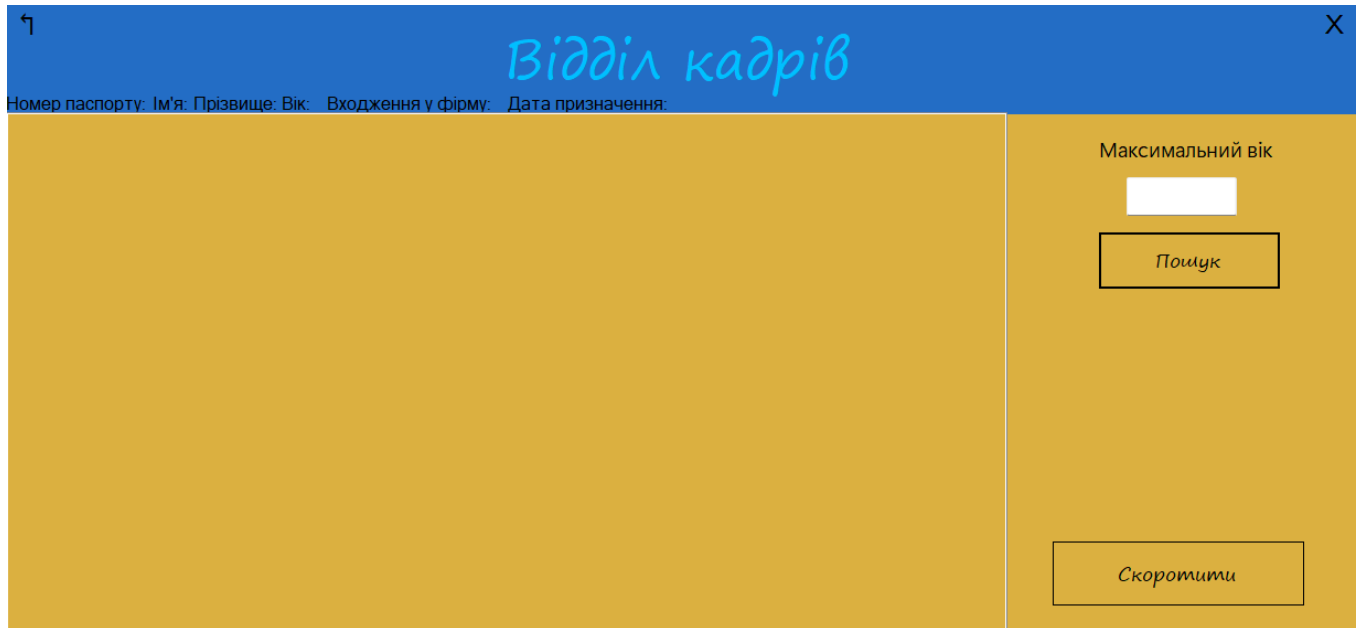


Рисунок 1.10 – Вікно скорочення працівників

Користувач вводить у поле пошуку максимальний вік працівника та натискає пошук, після чого буде виведено список усіх працівників старше заданого віку.

У разі некоректного введення віку у поле пошуку програма виведе діалогове вікно із помилкою коректності вводу даних(рис. 1.11)

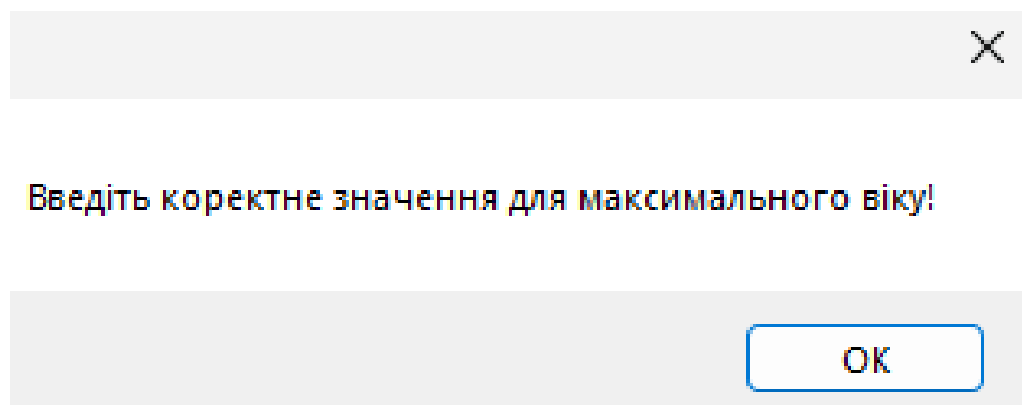


Рисунок 1.11 – Вікно помилки поля пошуку

Після обрання усіх працівників користувач натискає кнопку “Скоротити” після чого буде виведене діалогове вікно підтвердження(рис. 1.4).

Після вдалого скорочення штатів буде виведене вікно із успішним видаленням працівників(рис. 1.12).

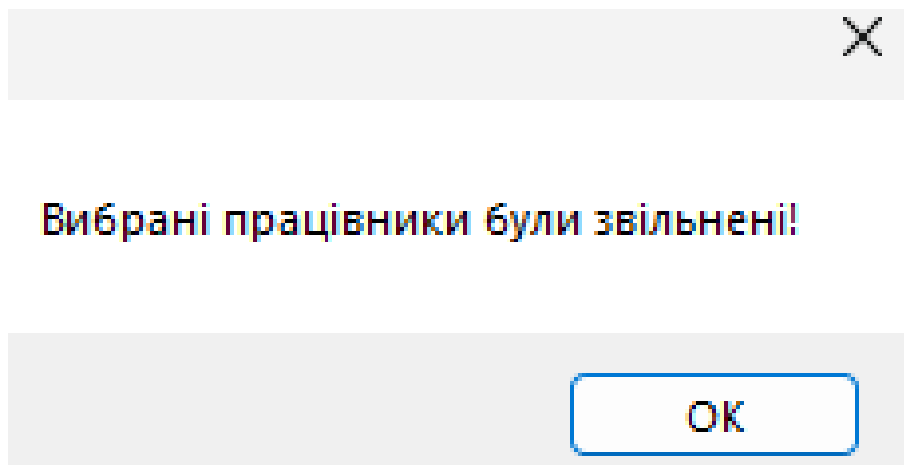


Рисунок 1.12 – Вікно підтвердження скорочення штатів

Після чого програма оновить список та залишиться у вікні скорочення задля можливості подальшого скорочення персоналу.

Функція 5. Пошук працівників

Панель пошуку (рис. 1.11) знаходиться на у вікнах звільнення(рис. 1.7)., скорочення (рис. 1.10). та допоміжному вікні програми(рис. 1.2). У кожному вікні поле пошуку виконує окрему функцію. У вікні звільнення буде відбуватись пошук працівників за усіма полями працівників, у полі скорочення пошук буде виводити на екран у вигляді списку працівників тільки заданої певної вікової групи.

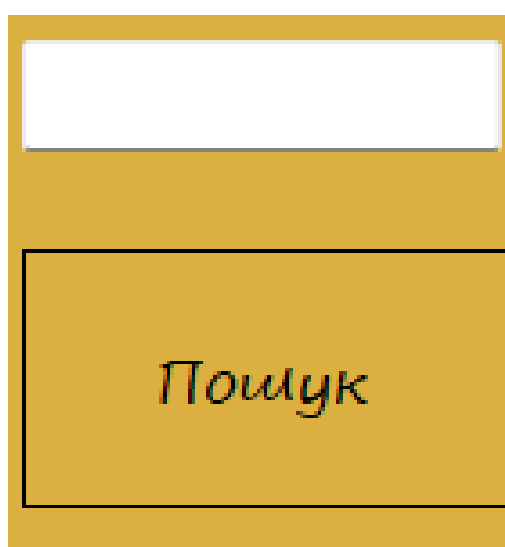


Рисунок 1.13 – Панель пошуку працівників

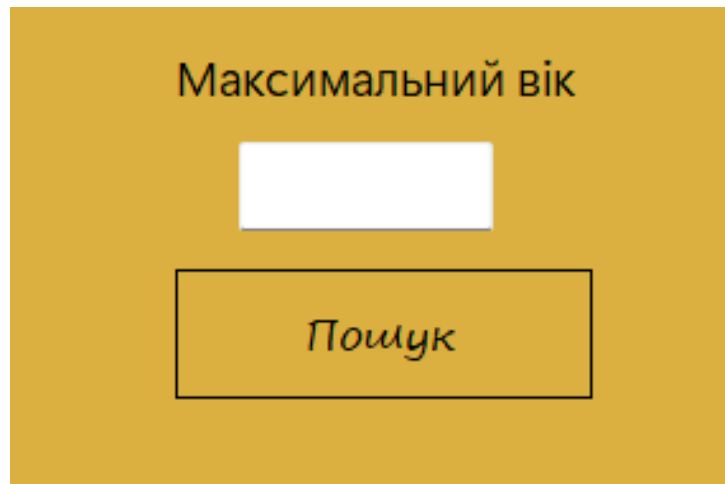


Рисунок 1.14 – Панель пошуку працівників(за віком)

Панель пошуку(рис. 1.10) – текстове поле, в якому пошук здійснюється за такими критеріями:

1. Ім'я
2. Прізвище,
3. Номер паспорту,
4. Вік,
5. Освіту,
6. Спеціальність,
7. Посаду,
8. Оклад.

Задля розпочатку пошуку користувачу потрібно увести дані у форму та натиснути кнопку “Пошук”.

Функція 6. Збереження, завантаження працівників

Данна функція працює автономно у коді програми. Після кожного переходу між формами а також взаємодії із будь-якими кнопками не помітно для користувача, данні автоматично оновлюються задля забезпечення цілісності файлів. Запис працівників відбувається як списком так і поодинокими персонами, дане рішення створено прихованим від користувача оскільки при першому

використанні користувач може припуститись помилок при роботі із файлами та порушити цілісність інших файлів.

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).

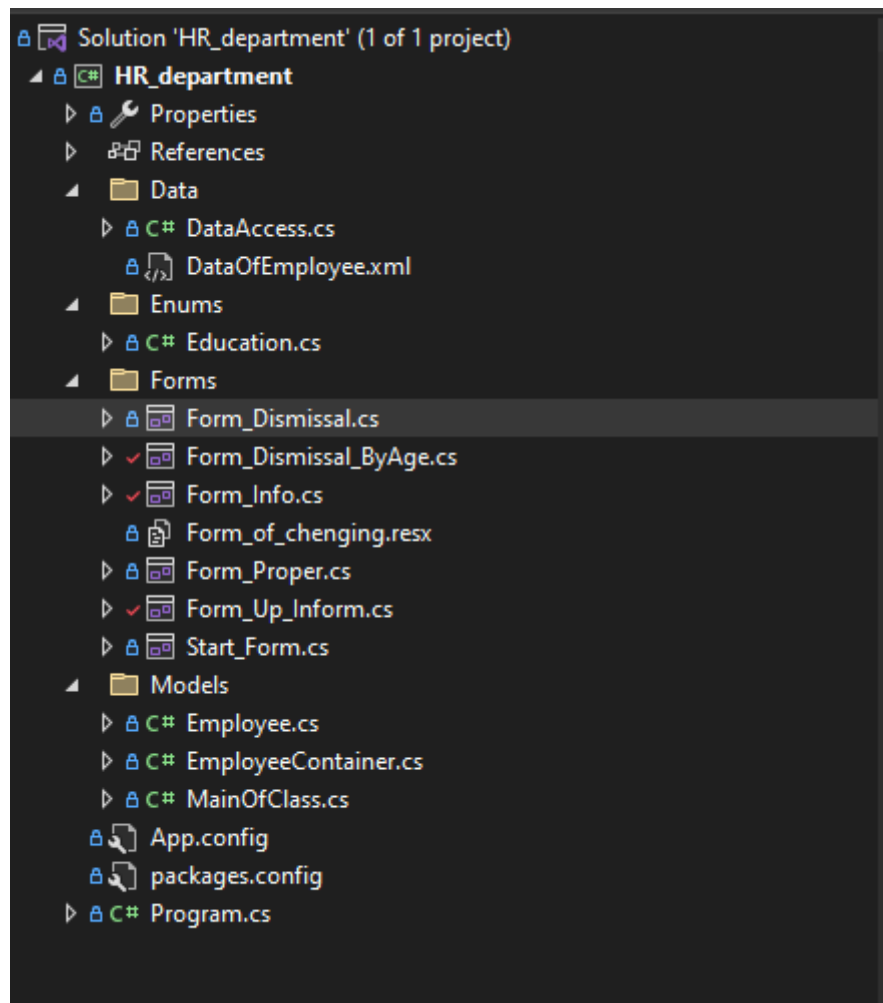


Рисунок 2.1 – Структура файлів проекту

Усі данні будуть зберігатись у форматі XML. Таке рішення обумовлене серіалізацією приватних полів а також більшою ефективністю стиснення. Також при пошкодженні файлів відновлення пошкодженої ділянки коду можна найкращим способом відновити данні (мова йде про переваги структурованого запису об'єктів).

У проєкті є такі класи:

1. Клас "Start Form":

Це головне вікно програми, яке з'являється при запуску. З нього відбувається уся навігація по програмі.

2. Клас "Form info":

Це вікно відкривається з головної сторінки. Використовується для показу інформації про працівників.

3. Клас "Form up Inform":

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

4. Клас "Form Dismissal":

Це вікно відкривається з вікна подачі наказів. Використовується для звільнення працівників.

5. Клас "Employee":

Представляє об'єкт, що описує працівника в програмі.

6. Клас "Employee Container":

Клас, що містить список працівників а також останнього використаного працівника. Використовується для серіалізації даних.

7. Клас "Form Dismissal By Age":

Це вікно відкривається з вікна подачі наказів. Використовується для скорочення працівників за віком

3 ІНСТРУКЦІЯ КОРИСТУВАЧА

3.1 Встановлення програми

Процедура встановлення програми протікає таким чином:

- 1) Завантажити .zip файл.
- 2) Розархівувати його.
- 3) Далі шукаємо .exe у папках. HR_department > obj > Debug > notebook.exe.

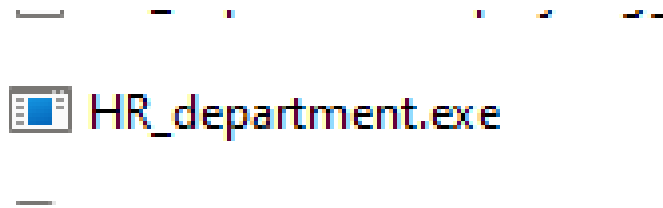


Рисунок 3.1 – Вигляд у папці

- 4) Запустити застосунок

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. 3.2), з якого проводяться усі переходи до наступних вікон.

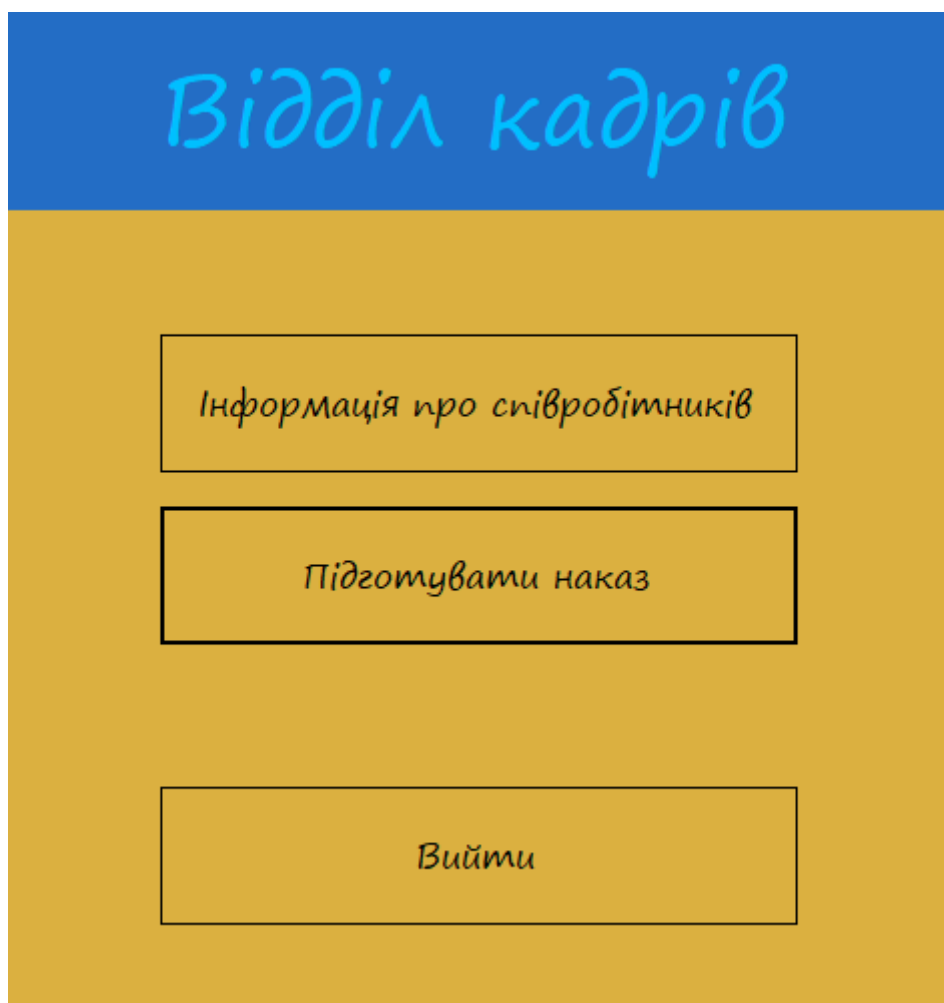


Рисунок 3.2 – Головне вікно програми

На головному екрані знаходяться три кнопки. Перша кнопка “Інформація про співробітників” безпосередньо відкриває нам вікно із усією інформацією про працівників. Друга кнопка “Підготувати наказ ” відкриє нам нову форму у якій ми можемо працювати із поданням наказів.

Розглянемо роботу кнопки “Інформація про співробітників”:

При натисненні нам відкривається вікно із інформацією про працівників, поле пошуку та кнопка редагування(рис. 3.3):

Відділ кадрів									
Прізвище:	Ім'я:	Вік:	Номер паспорту:	Освіта:	Спеціальність	Оклад:	Посада:	Входження у фірму:	Дата призначення
Петров	Іван	30	12345	Середня	Розробк...	5000	Розробн...	01.05.2015	15.01.2022
Сміт	Жанна	28	67890	Середня	Маркетинг	4000	Маркети...	10.08.2016	03.12.2021
Джонсон	Давид	35	54321	Середня	Біохімія	6000	Науковець	15.03.2014	20.05.2023
Браун	Емілія	32	98765	Середня	Графічні...	4500	Графічні...	05.10.2017	08.09.2022
Вільсон	Михайло	40	13579	Бакалав...	Фінанси	5500	Фінансо...	01.12.2010	25.07.2021
Тейлор	Олівія	27	24680	Бакалав...	Управлін...	3800	Асистент...	12.04.2019	18.02.2023
Кларк	Вільям	33	86420	Професор	Продажі	4200	Предста...	20.07.2013	10.11.2022
Андерсон	Софія	29	11111	Магістра...	Аналіз да...	4800	Аналітик ...	08.02.2016	27.03.2023
Томас	Яків	31	99999	Середня	Управлін...	5200	Менедже...	18.09.2014	05.08.2022
Робертс	Ава	26	77777	Магістра...	Публічні ...	4000	Спеціалі...	10.01.2020	30.04.2023

Рисунок 3.3 – Приклад роботи вікна “Інформація про співробітників”

Тут відразу у вікні розташовано панель пошуку(рис. 3.4):

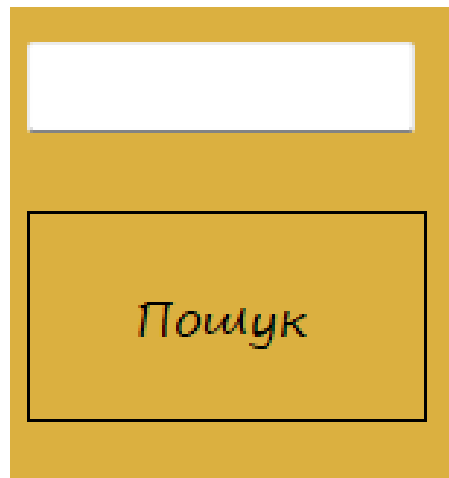


Рисунок 3.4 – Панель пошуку у вікні Інформація про співробітників”

Після обирання працівника зі списку та натискання кнопки “Редагувати” користувач потрапляє у форму зі змінною даних працівника (рис. 3.5):

Ім'я:	Яків
Прізвище:	Томас
Номер паспорту:	99999
Спеціальність:	Управління
Освіта:	Середня
Вік:	31
Посада:	Менеджер з
Оклад:	5200
Дата народження:	18.09.2014
Дата призначення:	05.08.2022

Зберегти

Рисунок 3.5 – Приклад редагування працівника

Після редагування користувач натискає кнопку “Зберегти” вас питають чи дійсно ви хочете зберегти нового працівника. (рис. 3.6):

Підтвердження

Ви впевнені?

Так Ні

Рисунок 3.6 – Діалогове вікно підтвердження

У разі введення не коректних даних, виведеться вікно із помилкою(рис. 3.7):

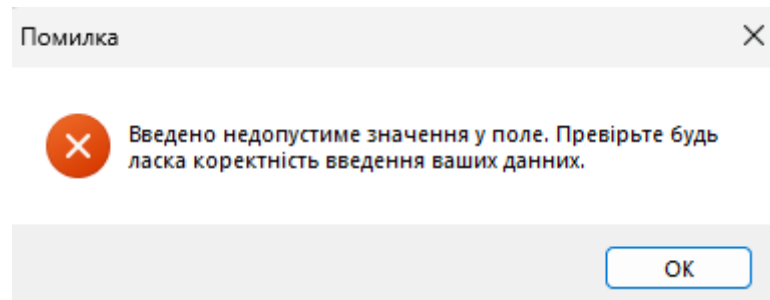


Рисунок 3.7 – Помилка при введенні даних у форму

Натиснувши кнопку “Зберегти”, програма перевірить коректність введених Вами даних, якщо буде помилка, то програма сповістить Вас (рис. 3.7). Якщо ж помилки немає, то з’явиться діалогове вікно для підтвердження операції (рис. 3.6). Якщо не підтверджуємо, то ми повертаємося на попереднє вікно програми.

Якщо ми підтверджуємо редагування працівника тоді програма оновить список (рис. 3.8.):

Відділ кадрів										
Прізвище:	Ім'я:	Вік:	Номер паспорту:	Освіта:	Спеціальність	Оклад:	Посада:	Входження у фірму:	Дата призначення	
Петров	Іван	30	12345	Середня	Розробк...	5000	Розробн...	01.05.2015	15.01.2022	
Сміт	Жанна	28	67890	Середня	Маркетинг	4000	Маркети...	10.08.2016	03.12.2021	
Джонсон	Давид	35	54321	Середня	Біохімія	6000	Науковець	15.03.2014	20.05.2023	
Браун	Емілія	32	98765	Середня	Графічні...	4500	Графічні...	05.10.2017	08.09.2022	
Вільсон	Михайло	40	13579	Бакалав...	Фінанси	5500	Фінансо...	01.12.2010	25.07.2021	
Тейлор	Олівія	27	24680	Бакалав...	Управлін...	3800	Асистент...	12.04.2019	18.02.2023	
Кларк	Вільям	33	86420	Професор	Продажі	4200	Предста...	20.07.2013	10.11.2022	
Андерсон	Софія	29	11111	Магістра...	Аналіз да...	4800	Аналітик ...	08.02.2016	27.03.2023	
Томас	Яків	31	99999	Середня	Управлін...	5200	Менедже...	19.06.2000	05.08.2022	
Робертс	Ава	26	77777	Магістра...	Публічні ...	4000	Спеціалі...	10.01.2020	19.06.2000 23	

Рисунок 3.8 – Результат операції

Далі користувач може повернутись у головне вікно за допомогою кнопки у лівому верхньому куті(рис. 3.9.):

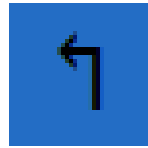


Рисунок 3.9 – Кнопка повернення на попереднє вікно

Після натискання ми повертаємось у головне меню(рис. 3.2.)

Натиснення у головному меню кнопки “Підготувати наказ” відкриє нам нову форму із трьома кнопками для навігації(рис. 3.10.):

Рисунок 3.10 – Вікно для вибору подання наказу

Із даного вікна користувач може подати 3 види наказів:

- 1) “Найм”
- 2) “Звільнення”
- 3) “Скорочення штатів пенсійного та передпенсійного віку”

Розгляньмо усі можливості подання наказів:

При натисненні кнопки “Найм” нам відкривається вікно із формами які необхідно заповнити для додавання нового працівника (рис. 3.11.):

Відділ кадрів

Ім'я:

Прізвище:

Номер паспорту:

Спеціальність:

Освіта:

Вік:

Посада:

Оклад:

Дата народження:

Дата призначення:

Зберегти

Рисунок 3.11 – Вікно для найму нового працівника

Користувач вводить данні та натискає кнопку “Зберегти”, після чого йому буде виведено на екран діалогове вікно із підтвердженням (рис. 3.6). Якщо не

підтверджуємо, то ми повертаємося на попереднє вікно програми. При не заповненні якогось поля на екран буде виведене повідомлення (рис. 3.12).

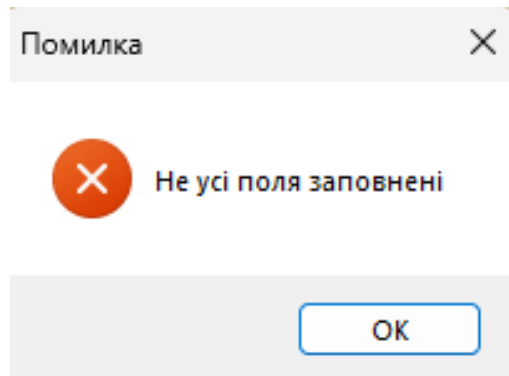


Рисунок 3.12 – Вікно помилки не заповнення поля

Також у разі запису у числове поле буквених символів буде виведене наступне повідомлення (рис. 3.13):

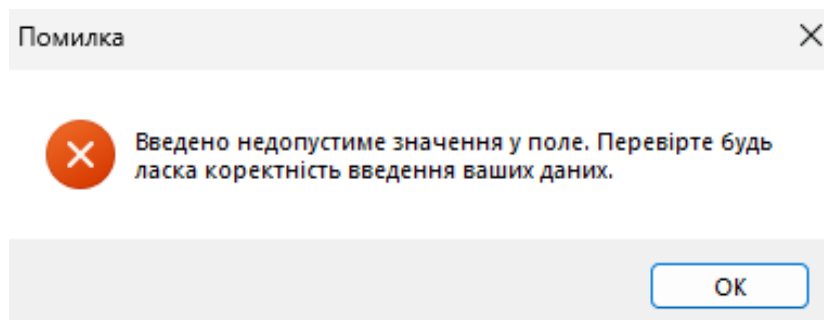


Рисунок 3.13 – Вікно помилки заповнення полів

Далі натиснувши кнопку повернення (рис. 3.9.) ми повертаємось у вікно для вибору подання наказу (рис. 3.10)

При натисненні кнопки “Звільнення ” нам відкривається вікно зі списком, формою пошуку та самою кнопкою “Звільнити”(рис. 3.14.):

The screenshot shows a window titled 'Відділ кадрів' (HR Department) with a blue header and a yellow body. The header contains a back arrow icon and a close 'X' icon. Below the header, there is a search form with labels: 'Номер паспорту:', 'Ім'я:', 'Прізвище:', 'Вік:', 'Входження у фірму:', and 'Дата призначення:'. The form area is currently empty. At the bottom, there are two buttons: 'Звільнити' (Dismiss) and 'Пошук' (Search). A small white input field is visible next to the 'Пошук' button.

Рисунок 3.14 – Вікно звільнення працівника

Користувач вводить у форму пошуку данні та натискає кнопку пошук, після чого список оновиться та на екран нам буде показано шуканого працівника(рис. 3.15.):

The screenshot shows the same 'Відділ кадрів' (HR Department) window, but now it displays search results. The search form at the top now contains the following data: '86420, Вільям, Кларк, 33, 2013-07-20, 2022-11-10'. Below the form, the 'Пошук' (Search) button is highlighted. The 'Звільнити' (Dismiss) button remains at the bottom. A small white input field next to the 'Пошук' button now contains the text 'Вільям'.

Рисунок 3.15 – Приклад пошуку працівника

Після вибрання працівника та натиснення кнопки “Звільнити” буде відкрито діалогове вікно з підтвердженням (рис. 3.6.) Якщо не підтверджуємо, то ми повертаємося на попереднє вікно програми.

При підтвердженні користувачу буде виведено вікно у якому сказано що працівника було видалено (рис. 3.16.):

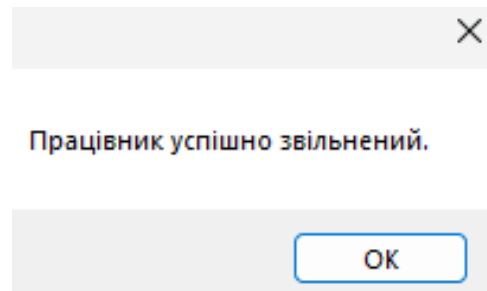


Рисунок 3.16 – Повідомлення про звільнення працівника

Після чого користувач залишиться у тому самому вікні та список оновиться (рис. 3.17.):

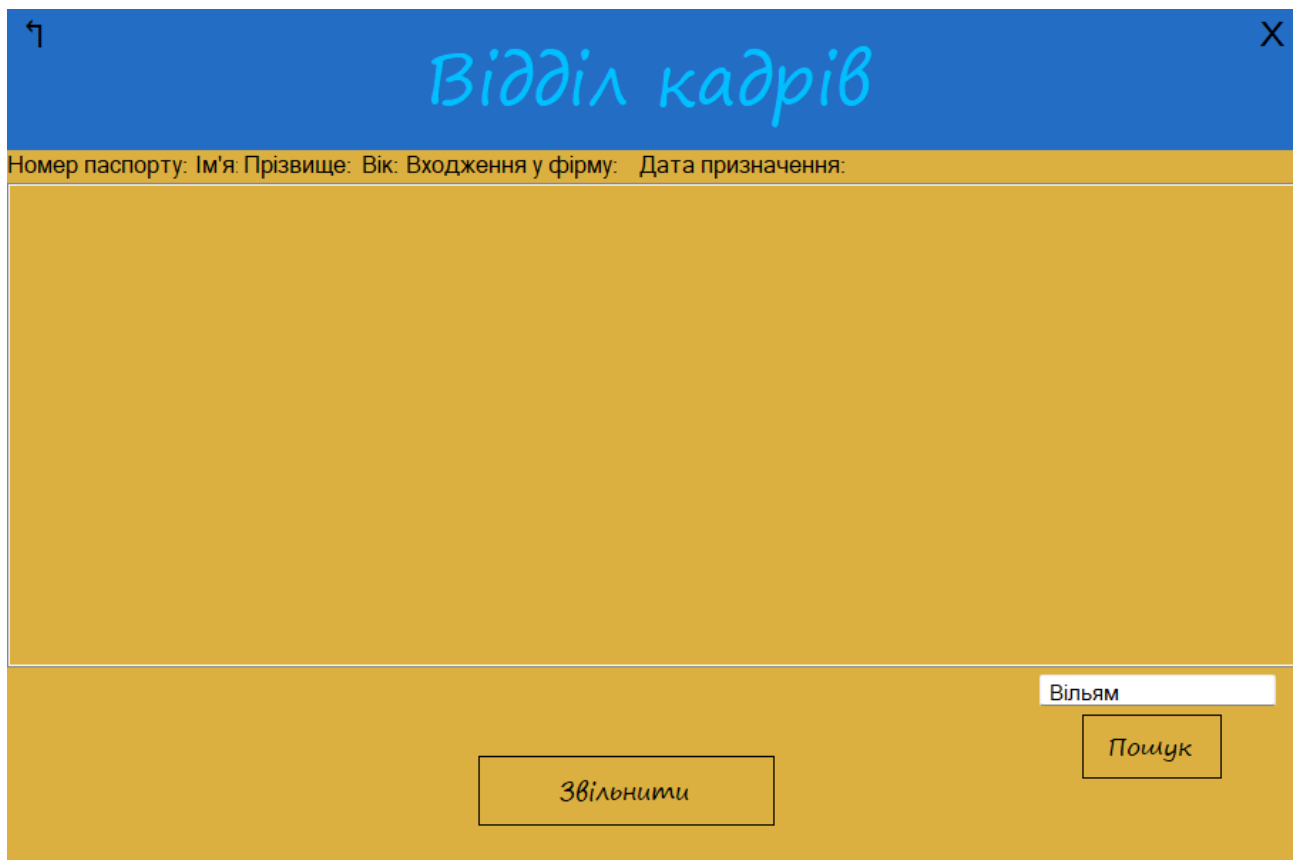


Рисунок 3.17 – Результат після звільнення працівника

Знову повертаємось на попереднє вікно за допомогою кнопки у лівому верхньому куті (рис. 3.9.)

При натисненні кнопки “Скорочення штатів пенсійного та передпенсійного віку” нам відкривається вікно зі списком, формою пошуку та самою кнопкою “Скоротити”(рис. 3.18.):

Рисунок 3.18 – Вікно скорочення працівників за віком

Далі користувач вводить у вікно пошуку максимальний вік працівника та натискає кнопку пошук. При введенні інших символів окрім цифр буде виведене наступне повідомлення (рис. 3.19.):

Рисунок 3.19 – Повідомлення про некоректне введення даних

У разі коректного введення даних список на формі оновиться та вивиде усіх працівників заданого віку та вище (рис. 3.20.):

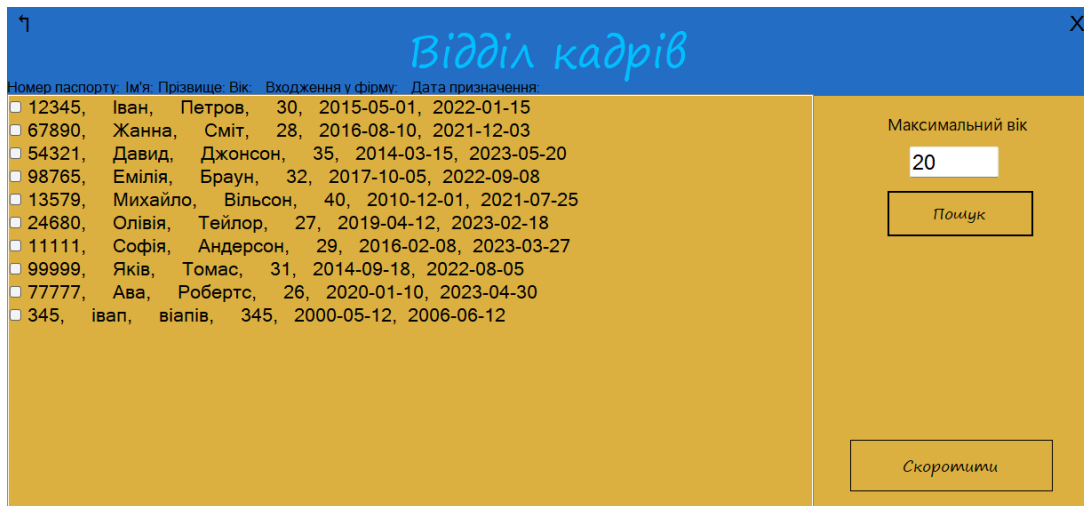


Рисунок 3.20 – Результат пошуку працівників за віком

Далі користувач вибирає кого необхідно скоротити та натискає кнопку “Скоротити” далі буде відкрито діалогове вікно з підтвердженням (рис. 3.6.)

Якщо не підтверджуємо, то ми повертаємося на попереднє вікно програми.

У разі підтвердження виводиться повідомлення про звільнення працівників(рис. 3.21.): та відбувається оновлення списку(рис. 3.22.):

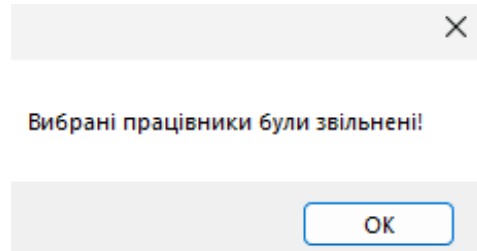


Рисунок 3.21 – Повідомлення про скорочення працівників

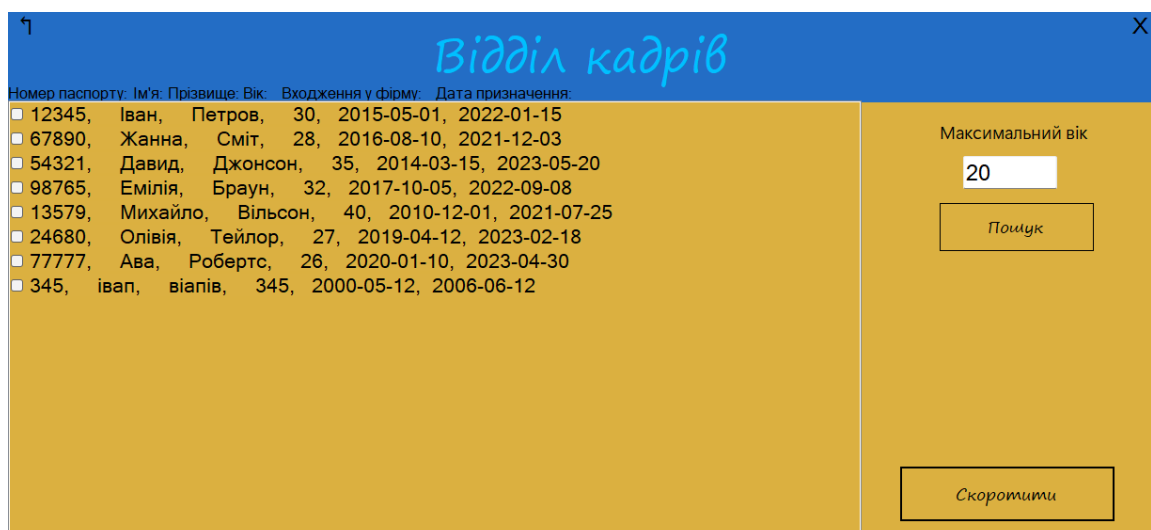


Рисунок 3.21 – Результат скорочення працівників

ВИСНОВКИ

Розроблена програма успішно відповідає всім початковим вимогам і надає широкі можливості користувачам. За допомогою інтуїтивно зрозумілих діалогових вікон, вона дозволяє здійснювати зручне отримання, редагування та зберігання інформації про працівників. Програма також включає функціонал прийняття на роботу та звільнення працівників, а навіть дозволяє здійснювати скорочення штату, якщо така потреба виникає.

Під час розробки застосунку, я вклав багато зусиль у покращення своїх навичок програмування мовою C#. Використання Windows Forms дало мені можливість створити зручний і привабливий інтерфейс для користувачів. За допомогою формату XML, я організував зберігання даних про працівників, забезпечуючи надійність та доступність цієї інформації.

У результаті мого наполегливого старання, розроблена програма стала потужним інструментом для керування персоналом. Вона спрощує рутинні задачі, зменшує можливість помилок та забезпечує збереження даних в безпеці. Надійна, зручна та з легкістю розширювана, ця програма стане незамінним помічником у роботі з персоналом для будь-якого підприємства.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бондарєв В.М. Електронний учбовий посібник з дисципліни «Основи програмування» URL: <http://tss.co.ua:5555/>(дата звернення: 31.05.2023)

2. Windows Forms documentation. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-7.0> (дата звернення: 31.05.2023)

ДОДАТОК А

Код програми

```

public partial class Start_Form : Form
{
    public Start_Form()
    {
        InitializeComponent();
    }
    private void infoButton_Click(object sender, EventArgs e)
    {
        Form_Info _form_Info = new Form_Info();
        _form_Info.FormClosed += (eventSender, EventArgs) => Show();
        _form_Info.Show();
        Hide();
    }
    private void button4_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
    private void button3_Click(object sender, EventArgs e)
    {
        Form_Proper _form_Proper = new Form_Proper();
        _form_Proper.FormClosed += (eventSender, EventArgs) => Show();
        _form_Proper.Show();
        Hide();
    }

    private Point lastPoint;
    private void panell1_MouseMove(object sender, MouseEventArgs e)
    {
        if (e.Button == MouseButtons.Left)
        {
            Left += e.X - lastPoint.X;
            Top += e.Y - lastPoint.Y;
        }
    }

    private void panell1_MouseDown(object sender, MouseEventArgs e)
    {
        lastPoint = new Point(e.X, e.Y);
    }
}

public partial class Form_Up_Inform : Form
{
    private Point lastPoint;
    private readonly Employee ChengetEmployee;
    public Form_Up_Inform()
    {
        InitializeComponent();
        ChengetEmployee = new Employee();
    }

    public Form_Up_Inform(Employee employeeInfo)
    {
        InitializeComponent();
        ChengetEmployee = employeeInfo;
        textBox1.Text = employeeInfo.Name.ToString();
        textBox2.Text = employeeInfo.Surname.ToString();
        textBox3.Text = employeeInfo.PassNum.ToString();
        comboBox1.Text = employeeInfo.Education.ToString();
    }
}

```

```

        textBox4.Text = employeeInfo.Specialization.ToString();
        textBox5.Text = employeeInfo.Age.ToString();
        textBox6.Text = employeeInfo.Position.ToString();
        textBox7.Text = employeeInfo.Salary.ToString();
        textBox8.Text = employeeInfo.EntryIntoCompany.ToString();
        textBox9.Text = employeeInfo.LastAppointment.ToString();
    }
    private void ClouseButton_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Education enumValue;
        if (Enum.TryParse(comboBox1.Text, out enumValue))
        {
            DialogResult result = MessageBox.Show("Ви впевнені?",
"Підтвердження", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (result == DialogResult.Yes)
            {
                if (string.IsNullOrEmpty(textBox1.Text) ||
                    string.IsNullOrEmpty(textBox2.Text) ||
                    string.IsNullOrEmpty(textBox3.Text) ||
                    string.IsNullOrEmpty(textBox5.Text) ||
                    string.IsNullOrEmpty(textBox4.Text) ||
                    string.IsNullOrEmpty(textBox6.Text) ||
                    string.IsNullOrEmpty(textBox7.Text) ||
                    string.IsNullOrEmpty(textBox8.Text) ||
                    string.IsNullOrEmpty(textBox9.Text))
                {
                    MessageBox.Show("Не усі поля заповнені", "Помилка",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
                    return;
                }

                try
                {
                    ChengetEmployee.Name = textBox1.Text;
                    ChengetEmployee.Surname = textBox2.Text;
                    ChengetEmployee.PassNum = int.Parse(textBox3.Text);
                    ChengetEmployee.Education = enumValue;
                    ChengetEmployee.Age = int.Parse(textBox5.Text);
                    ChengetEmployee.Specialization = textBox4.Text;
                    ChengetEmployee.Position = textBox6.Text;
                    ChengetEmployee.Salary = int.Parse(textBox7.Text);
                    ChengetEmployee.EntryIntoCompany =
                    DateTime.Parse(textBox8.Text);
                    ChengetEmployee.LastAppointment =
                    DateTime.Parse(textBox9.Text);

                    Serialization.SerializeObject(ChengetEmployee);
                }
                catch (FormatException)
                {
                    MessageBox.Show("Введено недопустиме значення у поле.
Перевірте будь ласка коректність введення ваших даних.", "Помилка",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Сталася помилка: {ex.Message}", "Помилка",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else
            {
                return;
            }
        }
    }

```

```

    }
    else
    {
        MessageBox.Show("Недопустиме значення в полі освіта.", "Помилка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
private void panell1_MouseDown(object sender, MouseEventArgs e)
{
    lastPoint = new Point(e.X, e.Y);
}

private void panell1_MouseMove(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        Left += e.X - lastPoint.X;
        Top += e.Y - lastPoint.Y;
    }
}

private void button4_Click(object sender, EventArgs e)
{
    Close();
}

public partial class Form_Proper : Form
{
    private Point lastPoint;

    public Form_Proper()
    {
        InitializeComponent();
    }

    private void ClouseButton_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void buttonLogin_Click(object sender, EventArgs e)
    {
        Form_Up_Inform _form_Up_Inform = new Form_Up_Inform();
        _form_Up_Inform.FormClosed += (eventSender, EventArgs) => Show();
        _form_Up_Inform.Show();
        Hide();
    }

    private void panell1_MouseDown(object sender, MouseEventArgs e)
    {
        lastPoint = new Point(e.X, e.Y);
    }

    private void panell1_MouseMove_1(object sender, MouseEventArgs e)
    {
        if (e.Button == MouseButtons.Left)
        {
            Left += e.X - lastPoint.X;
            Top += e.Y - lastPoint.Y;
        }
    }

    private void Dismissal_Click(object sender, EventArgs e)
    {
        Form_Dismissal _form_Dismissal = new Form_Dismissal();
        _form_Dismissal.FormClosed += (eventSender, EventArgs) => Show();
        _form_Dismissal.Show();
        Hide();
    }
}

```

```

private void DismissalOld_Click(object sender, EventArgs e)
{
    Form_Dismissal_ByAge _form_Dismissal_ByAge = new Form_Dismissal_ByAge();
    _form_Dismissal_ByAge.FormClosed += (eventSender, EventArgs) => Show();
    _form_Dismissal_ByAge.Show();
    Hide();
}

private void button5_Click(object sender, EventArgs e)
{
    Close();
}

}

public partial class Form_Info : Form
{
    private Point lastPoint;
    private List<Employee> employees;
    public Form_Info()
    {
        InitializeComponent();
        employees = Serialization.DeserializeObjects();
        dataGridView1.DataSource = employees;
        Font font = new Font("Arial", 12, FontStyle.Bold);

        foreach (DataGridViewColumn column in dataGridView1.Columns)
        {
            column.DefaultCellStyle.Font = font;
        }
        DataGridViewCellStyle cellStyle = new
DataGridViewCellStyle(dataGridView1.DefaultCellStyle);
        cellStyle.Font = new Font("Microsoft Sans Serif", 10,
GraphicsUnit.Pixel);

        // Застосовуємо новий стиль до dataGridView1
        dataGridView1.DefaultCellStyle = cellStyle;
    }
    private void ClouseButton_Click(object sender, EventArgs e)
    {
        System.Windows.Forms.Application.Exit();
    }

    private void buttonLogin_Click(object sender, EventArgs e)
    {
        if (dataGridView1.SelectedRows.Count > 0)
        {
            DataGridViewRow selectedRow = dataGridView1.SelectedRows[0];
            Employee selectedEmployee = (Employee)selectedRow.DataBoundItem;
            Form_Up_Inform _form_Up_Inform = new
Form_Up_Inform(selectedEmployee);
            _form_Up_Inform.FormClosed += (eventSender, EventArgs) => Show();
            _form_Up_Inform.Show();
            Hide();
        }
        else
        {
            MessageBox.Show("Виберіть користувача для редагування.");
        }
    }

    private void panell1_MouseDown_1(object sender, MouseEventArgs e)
    {
        lastPoint = new Point(e.X, e.Y);
    }

    private void panell1_MouseMove_1(object sender, MouseEventArgs e)
    {

```

```

        if (e.Button == MouseButton.Left)
        {
            Left += e.X - lastPoint.X;
            Top += e.Y - lastPoint.Y;
        }
    }

    private void button4_Click_1(object sender, EventArgs e)
    {
        Close();
    }

    private void ListOfEmployers_SelectedIndexChanged(object sender, EventArgs
e)
    {
    }

    private void button1_Click(object sender, EventArgs e)
    {
        List<Employee> listForDismissal;
        listForDismissal = MainOfClass.SearchEmployee(textBox1.Text, employees);
        dataGridView1.DataSource = listForDismissal;
    }

    public partial class Form_Dismissal_ByAge : Form
    {
        private Point lastPoint;
        private readonly List<Employee> employees;
        public Form_Dismissal_ByAge()
        {
            InitializeComponent();
            employees = Serialization.DeserializeObjects();
        }

        private void ClouseButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void panell1_MouseDown(object sender, MouseEventArgs e)
        {
            lastPoint = new Point(e.X, e.Y);
        }

        private void panell1_MouseMove(object sender, MouseEventArgs e)
        {
            if (e.Button == MouseButton.Left)
            {
                Left += e.X - lastPoint.X;
                Top += e.Y - lastPoint.Y;
            }
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void checkedListBox1_SelectedIndexChanged(object sender, EventArgs
e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (int.TryParse(textBox1.Text, out int maxAge)) // Отримуємо
максимальний вік з текстового поля
            {

```

```

        Updatethis(maxAge);
    }
    else
    {
        _ = MessageBox.Show("Введіть коректне значення для максимального віку!");
    }
}
private void Updatethis(int maxAge)
{
    checkedListBox1.Items.Clear(); // Очищаємо список перед пошуком

    foreach (Employee employee in employees)
    {
        if (employee.Age >= maxAge)
        {
            _ = checkedListBox1.Items.Add(employee.ToString()); // Додаємо працівника до списку
        }
    }
}

private void buttonLogin_Click(object sender, EventArgs e)
{
    List<Employee> employeesToFire = new List<Employee>();

    DialogResult result = MessageBox.Show("Ви впевнені?", "Підтвердження",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (result == DialogResult.Yes)
    {
    }
    else
    {
        return;
    }

    foreach (string selectedEmployee in checkedListBox1.CheckedItems)
    {
        Employee employee = employees.FirstOrDefault(emp => emp.ToString()
    == selectedEmployee);
        if (employee != null)
        {
            employeesToFire.Add(employee);
        }
    }

    if (employeesToFire.Count > 0)
    {
        foreach (Employee employee in employeesToFire)
        {
            _ = employees.Remove(employee);
        }

        Serialization.SerializeObjects(employees);
        _ = MessageBox.Show("Вибрані працівники були звільнені!");
        Updatethis(int.Parse(textBox1.Text));
    }
    else
    {
        _ = MessageBox.Show("Виберіть працівників для звільнення!");
    }
}
}

public partial class Form_Dismissal : Form
{
    private Point lastPoint;
    private List<Employee> Employers = Serialization.DeserializeObjects();
    public Form_Dismissal()

```

```

{
    InitializeComponent();
}

private void button4_Click_1(object sender, EventArgs e)
{
    Close();
}

private void ClouseButton_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void button1_Click(object sender, EventArgs e)
{
    List<Employee> listForDismissal;
    listForDismissal = MainOfClass.SearchEmploye(textBox1.Text, Employers);
    ListOfEmployers.Items.Clear();
    foreach (Employee employee in listForDismissal)
    {
        _ = ListOfEmployers.Items.Add(employee.ToString());
    }
}

private void button3_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show("Ви впевнені?", "Підтвердження",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (result == DialogResult.Yes)
    {
    }
    else
    {
        return;
    }
    if (ListOfEmployers.SelectedItem != null)
    {
        string selectedEmployee = ListOfEmployers.SelectedItem.ToString();

        Employee employeeToRemove = Employers.FirstOrDefault(emp =>
emp.ToString() == selectedEmployee);
        if (employeeToRemove != null)
        {
            _ = Employers.Remove(employeeToRemove);
            Serialization.SerializeObjects(Employers);

            ListOfEmployers.Items.Remove(selectedEmployee);

            _ = MessageBox.Show("Працівник успішно звільнений.");

            // Оновлення списку Employers після видалення працівника
            Employers = Serialization.DeserializeObjects();
        }
    }
}

private void panell1_MouseDown(object sender, MouseEventArgs e)
{
    lastPoint = new Point(e.X, e.Y);
}

private void panell1_MouseMove(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        Left += e.X - lastPoint.X;
        Top += e.Y - lastPoint.Y;
    }
}
}

```