**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

**КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**КУРСОВА РОБОТА**

на тему:

«Розподілений телефонний довідник»

з предмету об’єктно орієнтовне програмування С#

студента 2 курсу групи ПД-22

кафедри інженерії програмного забезпечення

Сковпень Михайла Анатолійовича

Викладач

аспірант, старший викладач кафедри

інженерії програмного забезпечення,

Гребенюк Віктор Вікторович

оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_

-Київ 2023-

Зміст

[1.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 4](#_Toc137340233)

[1.1 Історія створення та використання РТД 4](#_Toc137340234)

[1.2 Приклади використання РТД 6](#_Toc137340235)

[2 ВИБІР МЕТОДІВ ДЛЯ АНАЛІЗУ ПРЕДМЕТНОЇ ОБЛАСТІ 8](#_Toc137340236)

[2.1 Базові поняття розподіленого телефонного довідника 8](#_Toc137340237)

[2.2 Переваги та недоліки використання телефонного довідника 10](#_Toc137340238)

[2.3 Сучасні проблеми використання 12](#_Toc137340239)

[3. ОБГРУНТУВАННЯ ВИБОРУ СЕРЕДОВИЩА 14](#_Toc137340240)

[3.1 Вибір мови програмування 14](#_Toc137340241)

[3.1.1 Історія мови програмування 15](#_Toc137340242)

[3.1.2 Особливості мови 17](#_Toc137340243)

[3.1.3 Порівняння з конкурентами 20](#_Toc137340244)

[3.1.4 Версії мови C# 21](#_Toc137340245)

[3.2 Вибір середовища розробки 23](#_Toc137340246)

[3.2.1 Історія середовища розробки C# 25](#_Toc137340247)

[3.2.2 Розподіл IDE за типами 26](#_Toc137340248)

[4 ПРЕДСТАВЛЕННЯ РЕЗУЛЬТАТІВ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ 28](#_Toc137340249)

[4.1 Використані технології 28](#_Toc137340250)

[4.2 Кодова база 30](#_Toc137340251)

[4.3 Представлення результату роботи 38](#_Toc137340252)

[4.4 Перспективи проекту 43](#_Toc137340253)

[Висновки 44](#_Toc137340254)

[Перелік джерел 45](#_Toc137340255)

**Вступ**

Телефонний довідник є невід'ємною частиною нашого повсякденного життя, допомагаючий нам організовувати та зберігати контактну інформацію про наших друзів, сім'ю, колег та ділових партнерів. У сучасному цифровому світі, де інформація швидко змінюється та переміщається, ефективне управління та доступ до контактів стає все більш важливим.

Метою даної курсової є розробка телефонного довідника мовою програмування C#. Ми будемо створювати додаток, який дозволить користувачам додавати, переглядати та видаляти контакти, а також зберігати їх у файловій системі для подальшого використання.

У процесі розробки ми використовуватимемо принципи об'єктно-орієнтованого програмування, структури даних, обробку винятків, а також роботу з файлами для збереження та завантаження контактів. Ми також розглянемо питання безпеки, пов'язані зі збереженням особистої інформації користувачів.

В результаті виконання даної роботи, будуть опанувані основні навички розробки додатків мовою C#. Телефонний довідник буде корисним інструментом для зберігання та управління контактною інформацією, а також дозволить застосувати отримані знання в різних проектах.

Розробка відбувалася у Visual Studio Community 22, мовою програмування C# та платформою .NET 6.0

# 1.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Історія створення та використання РТД

Розподілена телефонна книга — це система, яка зберігає та надає інформацію про номери телефонів і контактну інформацію користувачів. Його історія налічує кілька етапів розвитку, від ручного ведення документації до сучасних цифрових рішень.

Етап 1: ручний запис

На початку існування телефонної системи ручний запис телефонних номерів був поширеним явищем. Кожен телефонний оператор або власник телефону несе самостійну відповідальність за збереження інформації про свої контакти. Система неорганізована та неефективна, особливо в умовах збільшення кількості абонентів.

Етап 2: централізована телефонна книга

У 19 столітті почалося створення централізованих телефонних книг, де інформація про абонентів зберігалася в одній системі. Це друковані книги, які містять списки телефонів і адреси абонентів. Власники телефонів можуть зв’язатися зі своїм оператором телефонної мережі, щоб ввести або змінити свої контактні дані. Ця система значно полегшує отримання інформації про користувачів, але все ще існують обмеження щодо відносної своєчасності та швидкості оновлення.

Етап 3: Електронна телефонна книга

З появою електроніки та комп’ютерних мереж телефонні книги стали оцифрованими. Користувачі можуть зберігати свої контакти на персональних комп’ютерах або інших електронних пристроях. Необхідно створити систему, яка дозволить обмінюватися цими даними між користувачами.

Етап 4: розподілена телефонна книга

З розвитком Інтернету та мережевих технологій стало можливим створення розподіленої телефонної книги. Розподілені системи дозволяють зберігати та обмінюватися контактною інформацією безпосередньо на серверах, доступних через Інтернет. Користувачі можуть реєструватися в такій системі, додавати свої контакти та встановлювати для них права доступу.

Сьогодні розподілені телефонні книги широко використовуються в особистих і ділових цілях. Вони забезпечують зручний спосіб зберігання, оновлення та пошуку контактів. Крім того, вони можуть містити інші функції, такі як фотографії користувачів, категорії контактів, коментарі та оцінки. Розподілені телефонні книги також широко використовуються в бізнес-середовищі для зберігання контактних даних клієнтів, партнерів і співробітників.

## 1.2 Приклади використання РТД

Розподілені телефонні книги використовуються у всіх сферах життя та бізнесу. Ось кілька прикладів їх використання:

* Особисте використання: розподілена телефонна книга дозволяє зберігати контактні дані родичів, друзів, колег та інших. Користувачі можуть оновлювати та керувати своїми контактами, додаючи фотографії, нотатки та іншу додаткову інформацію. Такі довідники забезпечують легкий доступ до контактів з будь-якого пристрою, підключеного до Інтернету.
* Ділові контакти: компанії використовують розподілені телефонні книги для зберігання та організації контактів клієнтів, партнерів і постачальників. Це дозволяє вам ефективно керувати контактною інформацією, шукати та оновлювати контактну інформацію, спілкуватися з клієнтами та керувати взаємодією з партнерами.
* Оператори зв’язку: телефонні компанії використовують розподілені телефонні книги для зберігання та надання інформації про своїх абонентів. Користувачі можуть використовувати такі довідники для пошуку номерів телефонів і контактних даних інших абонентів.
* Професійні мережі: Розповсюджені телефонні книги використовуються в професійних мережах, таких як університети чи компанії, де потрібно зберігати та мати доступ до контактної інформації студентів, викладачів та інших членів спільноти.
* Довідники послуг і бізнесу: розподілені телефонні довідники можна використовувати для створення довідників послуг і бізнесу, де можна знайти контактну інформацію для різних організацій, магазинів, ресторанів та інших підприємств.

Завдяки цифровим технологіям і доступу до Інтернету такі довідники стали необхідними інструментами для зберігання та організації контактів у сучасному світі.

# 2 ВИБІР МЕТОДІВ ДЛЯ АНАЛІЗУ ПРЕДМЕТНОЇ ОБЛАСТІ

## 2.1 Базові поняття розподіленого телефонного довідника

Користувач: це особа або організація, яка має доступ до розподіленого каталогу. Користувачі можуть додавати, редагувати та видаляти контакти, керувати правами доступу та використовувати функції каталогу.

Контакт: це запис, який містить інформацію про абонента або організацію. Контакти можуть містити імена, номери телефонів, адреси електронної пошти, адреси, фотографії та іншу додаткову інформацію, яка ідентифікує особу чи організацію.

Розподілені сервери: ці сервери зберігають інформацію про контакти користувачів і доступність. Ці сервери можуть бути розташовані на різних фізичних машинах або в різних регіонах, щоб забезпечити доступність і надійність.

Автентифікація та авторизація: це процеси перевірки та підтвердження особи користувача (автентифікація) і надання доступу до відповідних функцій каталогу (авторизація). Це забезпечує конфіденційність і безпеку інформації в каталозі.

Пошук і фільтр: Розповсюджена телефонна книга може мати функції пошуку та фільтрації, що дозволяє користувачам швидко знаходити потрібні контакти на основі різних умов, таких як ім’я, номер телефону та категорія.

Синхронізація: розподілені каталоги можуть підтримувати синхронізацію даних між різними пристроями або програмами. Це дозволяє користувачам оновлювати та отримувати доступ до своїх контактів з будь-якого пристрою, підключеного до мережі.

Права доступу: розподілені каталоги можуть підтримувати рівні доступу, дозволяючи встановлювати, хто має дозвіл переглядати, редагувати або видаляти певні контакти. Це корисно в командному чи організаційному середовищі, де різні люди мають різні ролі та обов’язки.

Ці концепції є фундаментальними для розуміння розподіленого каталогу для організації, керування та доступу до контактної інформації.

## 2.2 Переваги та недоліки використання телефонного довідника

Переваги:

* Зручність: телефонна книга дозволяє зберігати всю вашу контактну інформацію в одному місці та забезпечує швидкий доступ. Користувачі можуть швидко знаходити потрібні їм контакти та контакти, не запам’ятовуючи номери телефонів чи іншу контактну інформацію.
* Організація: телефонна книга дозволяє створювати й упорядковувати контакти за категоріями, групами чи іншими параметрами. Це полегшує пошук контактів і керування ними, особливо коли кількість контактів збільшується.
* Синхронізація: багато телефонних книг підтримують синхронізацію даних між різними пристроями та платформами. Це дозволяє підтримувати вашу інформацію в актуальному стані та отримувати доступ до неї з різних пристроїв, таких як смартфони, планшети чи комп’ютери.
* Додаткові функції: деякі телефонні книги можуть мати додаткові функції, такі як додавання фотографій контактів, резервне копіювання даних, нагадування про дні народження та події, синхронізація соціальних мереж тощо.

Недоліки:

* Конфіденційність і безпека: використання телефонної книги може викликати проблеми з конфіденційністю та безпекою. Зберігання контактної інформації на серверах або хмарних службах може поставити під загрозу конфіденційність даних, якщо не вжити належних заходів безпеки.
* Покладення на технології: для використання телефонної книги потрібен доступ до таких технологій, як смартфон або комп’ютер, і підключення до Інтернету. Відсутність доступу до цих ресурсів може обмежити можливість використання посібника.
* Ризик втрати даних: якщо резервне копіювання та регулярне оновлення даних не виконується, існує ризик втрати контактів у разі збою обладнання або технічних проблем каталогу.
* Спам і небажані контакти. Зберігання контактів у вашій телефонній книзі може призвести до отримання небажаних повідомлень або дзвінків від спамерів або незнайомців.

## 2.3 Сучасні проблеми використання

У сучасному світі використання телефонного довідника також пов'язане з деякими проблемами, серед яких можна виділити наступні:

* Збільшення обсягу контактів: Зараз люди мають все більше контактів, включаючи особисті, професійні, соціальні мережі та інші. Управління великим обсягом контактів може бути викликом, і деякі контакти можуть бути загублені або недостатньо оновлені.
* Надмірна залежність від технології: Залежність від смартфонів та комп'ютерів може мати негативні наслідки. Втрата або поломка пристрою, проблеми з підключенням до Інтернету або хмарних сервісів можуть призвести до недоступності контактної інформації.
* Приватність та безпека даних: Зберігання контактної інформації в розподіленому телефонному довіднику може піддавати ризику приватність та безпеку даних. Вразливості в системі можуть призвести до несанкціонованого доступу до особистих даних або їх витоку.
* Надмірна залежність від пошуку: Завдяки збільшенню обсягу контактів, люди стають більш залежними від пошуку контактів, замість того, щоб пам'ятати їх. Це може призводити до втрати навичок пам'яті та залежності від технології.
* Спам та небажані контакти: З популярністю телефонного довідника зростає ймовірність отримання спаму та небажаних контактів. Спамери можуть зловживати контактною інформацією, відправляти небажані повідомлення або нав'язливі дзвінки.

Ці проблеми стають актуальними в контексті сучасного використання телефонного довідника і потребують уваги при розробці та використанні відповідних рішень.

# 3. ОБГРУНТУВАННЯ ВИБОРУ СЕРЕДОВИЩА

## 3.1 Вибір мови програмування

Використання мови програмування C# для створення телефонних книг має кілька переваг:

* Об'єктно-орієнтований підхід: C# - це об'єктно-орієнтована мова програмування. Це означає, що код телефонної книги можна розбити на класи, об'єкти та методи і логічно організувати, що полегшує розробку, управління та розширення програми.
* Незалежність від платформи C# - це мова програмування, яка працює на платформах .NET Framework або .NET Core. Це означає, що додатки, написані на C#, можуть працювати на різних платформах, таких як Windows, macOS та Linux.
* Багата екосистема: C# має розгалужену та популярну екосистему, яка включає широкий спектр бібліотек, фреймворків та інструментів. Це спрощує розробку телефонних книг, оскільки для реалізації різних функцій можна використовувати готові компоненти та рішення.
* Легкість у вивченні C# - це мова програмування з чіткою синтаксичною структурою, що робить її дуже простою у вивченні, особливо для тих, хто вже знайомий з іншими мовами, такими як C, C++ та Java.
* Підтримка розподіленого програмування C# має потужні інструменти для розробки розподілених додатків, таких як телефонні книги, з використанням таких технологій, як WCF (Windows Communication Foundation) і ASP.NET. Це дозволяє створювати додатки, які можуть взаємодіяти з різними пристроями та мережами.

Загалом, використання мови програмування C# для створення телефонної книги є найкращим вибором для такого типу додатків, оскільки вона пропонує широкий спектр можливостей, гнучкість і простоту розробки. Однак вибір мови програмування також залежить від індивідуальних вимог і знань розробника.

## 3.1.1 Історія мови програмування

Мова програмування C# була розроблена в 1999 році командою під керівництвом Андерса Хейлсберга в корпорації Microsoft. Мова була створена в рамках ініціативи Microsoft .NET Framework для розробки програмного забезпечення для платформи Windows.

Історія розвитку мови C#

- У 1997 році корпорація Microsoft ініціювала проект зі створення нової мови програмування під назвою "Cool" (C-like Object Oriented Language) для вирішення проблем, пов'язаних з розробкою програмного забезпечення для Windows і веб-платформ.

- Пізніше проект було перейменовано на C# (англ. C sharp), щоб підкреслити схожість мови з C/C++.

- У 2000 році була випущена перша публічна версія C# як частина платформи .NET Framework 1.0.

- З того часу C# набула значної популярності серед розробників, зокрема в середовищі розробки Microsoft Visual Studio.

- У 2005 році вийшла версія C# 2.0, яка додала нові можливості, такі як узагальнення, індексатори та анонімні методи.

- У 2007 році вийшла версія C# 3.0, в якій з'явилися такі важливі функції, як лямбда-вирази, розширення методів та LINQ (Language-Integrated Query - інтегровані в мову запити).

- У 2010 році вийшла версія C# 4.0 з новими можливостями, такими як динамічне програмування, іменовані та необов'язкові аргументи та вбудовані COM-об'єкти.

- У 2012 році було випущено C# 5.0 з новими можливостями, такими як асинхронне програмування з використанням ключових слів `async` та `await`.

- У 2015 році Microsoft анонсувала C# 6.0 і включила нові можливості, такі як автоматичні властивості з ініціалізаторами, вбудована інтерполяція рядків та управління нульовими значеннями.

- У 2017 році було випущено C# 7.0 з новими можливостями, такими як кортежі, локальні функції та зіставлення шаблонів.

- У 2018 році було представлено C# 8.0 з новими можливостями, такими як занулювані типи посилань, асинхронні потоки та методи інтерфейсу за замовчуванням.

- У 2021 році вийшла версія C# 9.0 з новими можливостями, такими як типи записів, множинні ініціалізатори властивостей, поліморфізм і шаблони відповідей.

- У вересні 2021 року Microsoft анонсувала платформу .NET 6.0, яка надає нові функції та вдосконалення C# є основною мовою програмування для .NET 6.0, і ця версія включає новий компілятор, покращення продуктивності та крос-платформну підтримку Були додані покращення.

В даний час C# є однією з найпопулярніших мов програмування, яка використовується для розробки програмного забезпечення, особливо для платформи Windows, а також для створення широкого спектру додатків, веб-сайтів і сервісів C# продовжує стрімко розвиватися і мова C# і платформа . Нові версії платформи впроваджують нові можливості та функціональність для розробників.

## 3.1.2 Особливості мови

Мова програмування C# має кілька особливостей, які роблять її привабливою для розробки телефонних книг:

Сувора типізація: C# використовує сувору типізацію, яка дозволяє визначати типи даних змінних та об'єктів. Це дозволяє легше контролювати дані та зменшує ймовірність помилок під час виконання програми.

Обробка подій: C# підтримує події та делегати, що дозволяє ефективно маніпулювати змінними та станом програми. Це корисно при реалізації таких функцій, як додавання, видалення або оновлення контактів телефонної книги.

LINQ (Language-Integrated Query): C# включає LINQ, який дозволяє виконувати потужні запити та операції над колекціями даних. Це корисно для фільтрації, сортування та пошуку контактів у телефонній книзі.

Асинхронне програмування: C# має вбудовану підтримку асинхронного програмування за допомогою ключових слів async та await. Це дозволяє зручно виконувати асинхронні процеси, такі як завантаження даних з мережі або багатопотокова обробка, не блокуючи основний потік, що виконується.

Велика стандартна бібліотека C# постачається з великою стандартною бібліотекою, що містить багато корисних класів і функцій. Це дозволяє легко реалізувати різні операції, необхідні для телефонної книги, такі як операції з файлами, мережею та базами даних.

Багатоплатформенна підтримка: .NET Core, реалізація C#, дозволяє створювати багатоплатформні додатки, які працюють на різних операційних системах, таких як Windows, macOS і Linux. Це дозволяє розробляти телефонні книги, які працюють на різних пристроях і платформах.

Ці особливості мови програмування C# роблять її потужним і гнучким інструментом для розробки телефонних книг з різними функціями і продуктивністю.

Наступні можливості мови програмування C# є корисними для розробки телефонних книг:

Класи та об'єкти: класи та об'єкти: C# дозволяє створювати класи, які визначають структуру, властивості та методи телефонної книги. Класи можна використовувати для створення об'єктів, що представляють окремі контакти та їх властивості (ім'я, номер телефону, адреса тощо).

Зберігання даних Мова C# надає різноманітні способи зберігання даних, які можуть бути використані для зберігання контактів телефонної книги. Наприклад, можна використовувати масиви, списки і колекції, а також функції баз даних, такі як SQL Server і SQLite.

Операції пошуку та сортування: C# надає розширені функції для виконання пошуку та сортування даних. Зокрема, LINQ (Language-Integrated Query) дозволяє легко створювати запити до даних телефонної книги та фільтрувати результати.

Обробка вводу/виводу: C# надає інструменти для взаємодії з користувачем, такі як читання і запис даних з консолі та створення зручних інтерфейсів довідників за допомогою графічного інтерфейсу користувача (GUI).

Обробка помилок: C# надає механізм для обробки винятків і помилок та коректної роботи з непередбачуваними ситуаціями, такими як некоректні дані або проблеми зі зберіганням даних.

Можливості розподіленого програмування: використовуючи C# і такі технології, як WCF (Windows Communication Foundation) і ASP.NET, можна створювати розподілені додатки телефонної книги, які можуть взаємодіяти з іншими системами і пристроями через мережу.

Вбудована безпека C# надає механізми безпеки, такі як управління доступом до даних і захист від потенційних загроз, які є важливими для збереження конфіденційності та цілісності даних телефонної книги.

## 3.1.3 Порівняння з конкурентами

Порівняння C# з його конкурентами залежить від ваших конкретних вимог та ситуації розробки. Ось деякі популярні мови програмування, які можна порівняти з C#:

Java: Java: Як і C#, Java є об'єктно-орієнтованою мовою програмування зі схожою синтаксичною структурою. Обидві мови підтримують широкий спектр розподіленого програмування і мають багато спільних бібліотек і фреймворків: вибір між C# і Java залежить від платформи розробки (наприклад, .NET або Java Virtual Machine), екосистеми, доступності бібліотек і ресурсів, особистих особистих уподобань розробника.

Python Python - це інтерпретована мова програмування, тоді як C# потребує компіляції; Python має простий, зрозумілий синтаксис, багаті можливості для роботи з даними та широку підтримку спільноти. C#, з іншого боку, має вищу продуктивність, підтримку типів інтерфейсу та потужну типізацію, що робить її популярною для великомасштабних та промислових додатків.

JavaScript: JavaScript - це мова програмування, яка використовується для розробки клієнтської частини веб-додатків, тоді як C# в основному використовується для розробки на платформі .NET, хоча JavaScript і C# мають різний синтаксис і екосистеми, Обидві мають функції для розробки веб-додатків. Залежно від потреб проекту, може бути доцільним використання комбінації обох мов для реалізації клієнтської та серверної частин програми.

Go: Go - це мова програмування, призначена для створення надійних та ефективних системних додатків. Це мова зі спрощеним синтаксисом, спрямованим на простоту використання для розробників; у порівнянні з C#, Go має менше можливостей і обмежену підтримку деяких рішень, таких як графічний інтерфейс. Однак Go відома своєю швидкістю, простотою розгортання та масштабованістю, що робить її привабливою в деяких сценаріях розробки.

Обираючи мову програмування для телефонних довідників, важливо враховувати потреби проекту, команди розробників та наявність ресурсів і досвіду. Кожна з цих мов має свої переваги та недоліки, і вибір залежить від конкретних вимог та пріоритетів проекту.

## 3.1.4 Версії мови C#

Мова програмування C# має кілька версій, кожна з яких має певні розширення та покращення. Ось деякі з найважливіших версій C#:

* C# 1.0: Перша версія C#, випущена у 2002 році з платформою .NET Framework 1.0. Вона містить основні конструкції, такі як класи, методи, змінні та потоки.
* C# 2.0: Версія 2.0 була випущена в 2005 році з .NET Framework 2.0 і включала нові можливості, такі як інтерфейси, делегати, анонімні типи, типи, що обнуляються, ітератори та інші вдосконалення.
* C# 3.0: Версія 3.0, випущена в 2007 році з .NET Framework 3.5, включає такі важливі зміни, як лямбда-вирази, LINQ (запити мовної інтеграції), автоматична ініціалізація властивостей і розширення методів.
* C# 4.0: Випущена у 2010 році з .NET Framework 4.0, версія 4.0 включає нові можливості, такі як динамічні типи, іменовані та необов'язкові параметри, варіанти типів та покращені узагальнення.
* C# 5.0: Версія 5.0, випущена у 2012 році з .NET Framework 4.5, включає нові можливості, такі як асинхронне програмування з ключовими словами async та await, покращена обробка помилок (наприклад, ключове слово await у блоках try-catch) та C# 6.0: 2015.
* C# 6.0: Версія 6.0, випущена у 2015 році з Visual Studio 2015, включає ініціалізацію властивостей за допомогою автоматичних властивостей, інтерполяцію рядків, індексатори з ініціалізацією, нульові оператори (? , ??) та індексні властивості, серед інших нових можливостей.
* C# 7.0: Версія 7.0, випущена з Visual Studio 2017 у 2017 році, включає нові можливості, такі як кортежі, експресивні ініціалізатори об'єктів, оголошення параметрів методу у зовнішніх змінних, локальні функції, модифікатори доступу для аксесорів властивостей get/set тощо. були включені.
* C# 8.0: Версія 8.0, випущена з .NET Core 3.0 у 2019 році, включає нові можливості, такі як розширені шаблони шаблонів, індексатори з діапазонами, обнуляючі посилання та асинхронні безпечні регіони.
* C# 9.0: Версія 9.0, випущена разом з .NET 5 у 2020 році, включає нові можливості, такі як записи, ініціалізатори шаблонів властивостей, часткові методи з реалізацією, додаткові покращення для Nullable посилань та нові оператори.

Кожна нова версія C# забезпечує вдосконалення мови, нові функції та покращення, які дозволяють розробникам писати більш ефективний, безпечний та простий у використанні код. Рекомендується використовувати останню стабільну версію C#, яка підтримується на вашій платформі розробки.

## 3.2 Вибір середовища розробки

Вибір мови програмування C# та платформи .NET 6.0 для розробки телефонної книги має декілька умов:

* мова C#: Мова C#: C# - це потужна об'єктно-орієнтована мова програмування, яка широко підтримується в середовищі .NET. Завдяки розширеним можливостям типізації, потужній системі типів, зручному синтаксису та багатьом іншим функціональним можливостям, C# дозволяє розробникам писати чистий, структурований та ефективний код.
* платформа .NET 6.0: .NET 6.0 - це остання стабільна версія платформи .NET, що надає широкий спектр бібліотек, фреймворків та інструментів для розробки додатків. NET 6.0 включає в себе підвищення продуктивності, покращену багатоплатформну підтримку, інтеграцію з хмарними сервісами та багато інших інноваційних функцій.
* розширені можливості: за допомогою C# і .NET 6.0 ви можете використовувати широкий спектр інструментів і бібліотек для розробки телефонних книг: на стороні веб-сервера - ASP.NET, для настільної версії - WPF і WinForms, а для мобільної розробки - Xamarin і Blazor, Створювати крос-платформні додатки.
* єдина мова і платформа: мова C# і платформа .NET дозволяють писати код однією мовою і використовувати його на різних платформах, включаючи Windows, macOS, Linux і мобільні платформи. Це спрощує розробку і підтримку додатків, оскільки немає необхідності вивчати і використовувати різні мови і фреймворки.

Це полегшує розробку та підтримку додатків.

Отже, якщо ви розробляєте телефонну книгу, використовуючи мову програмування C# разом з платформою .NET 6.0, у вас є потужність, продуктивність і широкий спектр інструментів для створення веб-, десктопних або мобільних додатків.

## 3.2.1 Історія середовища розробки C#

Історія середовища розробки C# включає кілька важливих етапів, які сприяли розвитку та популяризації цієї мови програмування. Нижче наведено основні етапи історії середовища розробки C#:

Visual Studio .NET (2002): Visual Studio .NET (2002): середовище розробки Visual Studio .NET було випущено в 2002 році разом з першою версією мови C# Visual Studio .NET - це редактор коду, компілятор, відладчик, дизайнер форм та інші потужні інструменти для створення додатків на основі .NET Framework.

Visual Studio 2005: випущена у 2005 році, Visual Studio 2005 надавала розширену функціональність для розробки додатків на C#. Зокрема, вона підтримувала .NET Framework 2.0 і включала нові можливості, такі як автоматичне завершення коду, покращений відладчик, інструменти для рефакторингу коду та інші корисні функції.

Visual Studio 2008: випущена у 2008 році, Visual Studio 2008 підтримує .NET Framework 3.5 і має нові можливості для розробки C# додатків. Сюди входить покращений LINQ, WCF (Windows Communication Foundation), WPF (Windows Presentation Foundation) та інші покращення.

Visual Studio 2010: випущена у 2010 році, Visual Studio 2010 принесла покращення у розробку C#. Було додано багато нових функцій, включаючи розширені можливості паралельного програмування, покращений відладчик і підтримку .NET Framework 4.0.

Visual Studio 2012: випущена у 2012 році, Visual Studio 2012 принесла нові можливості для розробки додатків на C#, включаючи підтримку платформи Windows 8 та покращену мобільну версію. Вона включала в себе поліпшення для розробки мобільних додатків з використанням технології Windows Phone.

Visual Studio 2013: випущена у 2013 році, Visual Studio 2013 містить нові функції та покращення для розробки додатків на C#. Це підтримка .NET Framework 4.5.1, покращений відладчик, інструменти для розробки хмарних додатків та багато іншого.

Visual Studio 2015: випущена в 2015 році, Visual Studio 2015 представила нові можливості для розробки на C#. Вона включає підтримку .NET Framework 4.6, розширення для Universal Windows Platform (UWP), покращені інструменти для розробки веб-додатків та багато інших можливостей.

Visual Studio 2017 та пізніші версії: випуски Visual Studio 2017 та пізніші версії продовжують приносити покращення та вдосконалення для розробки на C#. Кожна нова версія включає редактор коду, відладчик, підтримку нових версій .NET Framework та інші вдосконалення.

Історія середовища розробки C# відображає безперервний розвиток і вдосконалення інструментів для C# розробників; кожна нова версія Visual Studio надає нові можливості і функціональність, які допомагають підвищити продуктивність, ефективність і якість C# розробки.

## 3.2.2 Розподіл IDE за типами

Середовища розробки (IDE) для мови програмування C# надають розробникам потужні інструменти для розробки додатків цією мовою. Залежно від ваших потреб та вподобань, ви можете обрати IDE, яка підійде саме вам. Ось деякі популярні IDE для C# за типами:

Visual Studio: Visual Studio - найпопулярніше, повнофункціональне середовище розробки для C#. Воно має широкий набір інструментів, включаючи потужний редактор коду, налагоджувач, дизайнер форм, засоби тестування та підтримку контролю версій. Visual Studio також надає розширення для розробки веб-додатків, мобільних додатків, ігор та інших типів додатків на базі .NET.

Visual Studio Code: Visual Studio Code (VS Code) - це легке інтегроване середовище розробки, яке надає потужні можливості для розробки на C#. Він також підтримує ряд розширень, які розширюють функціональність для різних типів проектів; VS Code має редактор коду з підказками, відладчик, інструменти управління проектами та підтримку Git.

JetBrains Rider: Rider - це інтегроване середовище розробки, спеціально розроблене для роботи з платформами C# та .NET. Він пропонує розширені функції автозавершення, відладчик, рефакторинг, вбудовану підтримку аналізу коду та інші інструменти для полегшення процесу розробки.

MonoDevelop: MonoDevelop - це безкоштовне середовище розробки з відкритим вихідним кодом, спеціально призначене для розробки додатків на основі Mono Framework з підтримкою C#. Воно надає редактор коду з підсвічуванням синтаксису, відладчик, інструменти управління проектами та підтримку Git.

Це лише кілька прикладів середовищ розробки для мови програмування C#. Вибір конкретного середовища залежить від ваших потреб, вподобань та типу платформи і додатку, який ви плануєте розробляти.

# 4 ПРЕДСТАВЛЕННЯ РЕЗУЛЬТАТІВ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

## 4.1 Використані технології

У цьому коді використовуються наступні технології та бібліотеки:

System: Це простір імен в стандартній бібліотеці .NET, який надає основні класи та функції для роботи зі стрічками, списками, файлами тощо.

System.Collections.Generic: Цей простір імен містить реалізації загальних колекцій, таких як List<T>. У цьому коді використовується List<Contact> для збереження контактів.

System.IO: Цей простір імен надає класи для роботи з файлами та потоками. У цьому коді використовується File.Exists, File.ReadAllLines та File.WriteAllLines для завантаження та збереження контактів у текстовому файлі.

Console: Цей клас надає функції для взаємодії зі стандартним введенням/виведенням консолі. Використовується для виводу повідомлень та отримання введених даних від користувача.

В загальному, код використовує базові технології та бібліотеки, які постачаються разом з платформою .NET, для забезпечення основного функціоналу телефонного довідника, такого як завантаження та збереження контактів, додавання нових контактів, видалення контактів та відображення списку всіх контактів.

Крім технологій та бібліотек, зазначених вище, в цьому коді також використовуються наступні особливості мови програмування C#:

Властивості (Properties): У класах Contact та Phonebook використовуються властивості (Name, PhoneNumber), які надають зручний доступ до приватних полів цих класів. Властивості дозволяють контролювати доступ до даних та виконувати певні дії при читанні або записі значень.

Лямбда-вирази (Lambda Expressions): У методі RemoveContact використовується лямбда-вираз для знаходження контакту, який потрібно видалити. Лямбда-вирази дозволяють створювати анонімні функції для більш зручної та компактної роботи з колекціями.

Конструктори (Constructors): Клас Phonebook має конструктор, який приймає шлях до файлу телефонного довідника і використовує його для завантаження контактів. Конструктори дозволяють ініціалізувати об'єкти класу при їх створенні.

Умовні оператори (Conditional Statements): У коді використовуються умовні оператори if та switch для вибору відповідних дій залежно від обраного користувачем варіанту.

Цикли (Loops): Використовуються цикли foreach та while для ітерації по контактам та взаємодії з користувачем у безкінечному циклі меню.

## 4.2 Кодова база

Ця кодова база представляє програму для телефонної книги, яка дозволяє додавати, видаляти та відображати контакти. Давайте розглянемо кожну частину коду:

1. Клас `Contact` представляє контакт з полями `Name` (ім'я) та `PhoneNumber` (номер телефону).

Зображення, що містить текст, Шрифт, знімок екрана, ряд

Автоматично згенерований опис

2. Клас `Phonebook` містить логіку для роботи з телефонною книгою.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

- Конструктор `Phonebook` приймає шлях до файлу `filePath` і завантажує контакти з цього файлу за допомогою методу `LoadContacts`.

- Метод `LoadContacts` завантажує контакти з файлу, розділені комами, та створює об'єкти класу `Contact`, додаючи їх до списку `loadedContacts`.

Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис

- Метод `SaveContacts` зберігає контакти у файл, записуючи кожен контакт у форматі `ім'я,номер` у список `lines`, а потім записує цей список у файл за допомогою `File.WriteAllLines`.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

- Метод `AddContact` додає новий контакт до списку `contacts` та викликає метод `SaveContacts` для збереження змін у файлі. Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, програмне забезпечення, Мультимедійне програмне забезпечення, знімок екрана

Автоматично згенерований опис - Метод `RemoveContact` видаляє контакт зі списку `contacts`, шукаючи його за співпаданням імені та номера телефону, і також викликає метод `SaveContacts` для оновлення файлу.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис - Метод `GetAllContacts` повертає список всіх контактів.

3. Клас `Program` містить метод `Main`, який представляє виконавчу точку програми.

- В цьому методі створюється екземпляр класу `Phonebook`, передаючи шлях до файлу "phonebook.txt" у конструктор.

- У циклі `while (true)` виводиться меню з варіантами дій для користувача.

- Користувач вводить номер обраної дії, і за допомогою умовного оператора `switch` виконується відповідна функція:

- `AddContact` запускає процес додавання нового контакту до телефонної книги.

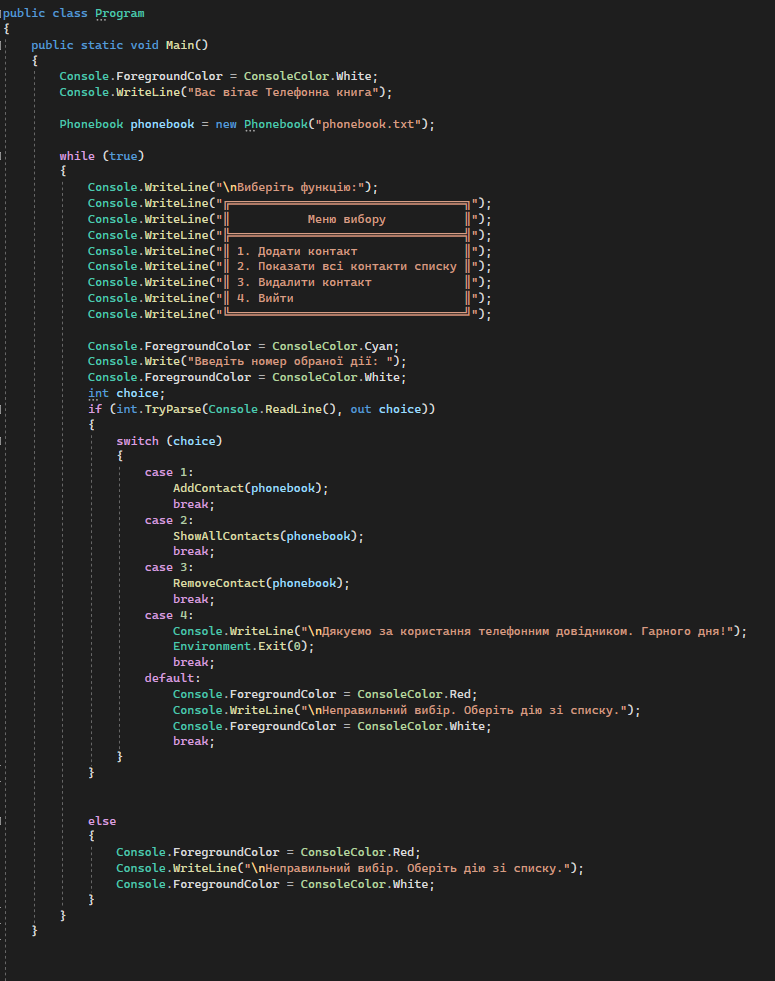
- `ShowAllContacts` виводить список всіх контактів, взятих з телефонної книги.

- `RemoveContact` запускає процес ви

далення контакту з телефонної книги.

- Якщо користувач вибирає варіант 4, програма виводить повідомлення про завершення роботи та викликає `Environment.Exit(0)` для виходу з програми.

- У разі неправильного вводу видається відповідне повідомлення про помилку.



4. Методи `AddContact`, `ShowAllContacts` та `RemoveContact` виконують взаємодію з користувачем, виводячи повідомлення та отримуючи введення даних для відповідних операцій над контактами.

- `AddContact` пропонує користувачу ввести ім'я та номер телефону нового контакту, створює об'єкт `Contact` і передає його методу `AddContact` класу `Phonebook`.

Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис

- `ShowAllContacts` виводить список всіх контактів, взятих з класу `Phonebook`.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис

- `RemoveContact` пропонує користувачу ввести ім'я та номер телефону контакту, який він хоче видалити, створює об'єкт `Contact` і передає його методу `RemoveContact` класу `Phonebook`.

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Ця кодова база реалізує просту телефонну книгу, яка дозволяє додавати, видаляти та відображати контакти. Вона використовує файл для збереження даних телефонної книги та надає зручний інтерфейс для взаємодії з користувачем.

## 4.3 Представлення результату роботи

Код реалізує просту телефонну книгу, що дозволяє користувачам додавати, видаляти та переглядати контакти.

При запуску програми відображається меню, що містить опції. Користувач може вибрати одну з наступних опцій

* Додати контакт: користувач вводить ім'я та номер телефону нового контакту, і контакт додається до телефонної книги.
* Показати всі контакти в списку: всі контакти, що зберігаються в телефонній книзі, відображаються на екрані.
* Видалити контакт: користувач може ввести ім'я та номер телефону контакту, який він бажає видалити. Якщо контакт буде знайдено, його буде видалено з телефонної книги.
* Вихід: вихід з програми.
* Якщо користувач введе неправильний номер дії, з'явиться повідомлення про помилку.

Телефонна книга зберігає дані у файлі з назвою "phonebook.txt"; контакти завантажуються з цього файлу при створенні об'єкта класу Phonebook. Кожен контакт зберігається у форматі "ім'я, номер".

При додаванні або видаленні контакту дані знову зберігаються у файлі.

Стартове вікно:

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Вибір дії 1 (Додавання контакту в розподілений телефонний довідник):



Процес додавання контакту:

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Результат успішного додавання контакту:

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис  
 Вибір дії 2 (Показування всіх контактів, доданих до розподіленого телефонного довідника):

Зображення, що містить Шрифт, знімок екрана, Графіка, типографія

Автоматично згенерований опис  
 Процес виконаної дії 2 та результат її виконання:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Вибір дії 3 (Видалення контакту із розподіленого телефонного довідника):

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

  
 Процес виконання дії 3 та результат її виконання(успішний):

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Процес виконання дії 3 та результат її виконання(не успішний):

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Виконана дія 3(контакт Тестовий 1 прибрався зі списку контактів):

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

Вибір дії 4 (Завершення роботи з розподіленим телефонним довідником):

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис  
 Виконана дія 4 (на екран виводиться надпис, котрий свідчить про вихід з розподіленого телефонного довідника):

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

## 4.4 Перспективи проекту

Проект телефонного довідника має потенціал для розвитку та впровадження широкого спектру функцій та вдосконалень. Ось деякі перспективи подальшого розвитку вашого проекту:

* Розширити функціональність: ви можете додати такі функції, як редагування контактів, пошук за іменем чи номером телефону або групування контактів. Це зробить телефонну книгу більш потужною та зручною для користувача.
* Графічний інтерфейс: наразі програма використовує текстовий інтерфейс командного рядка. Однак можна розробити більш інтуїтивно зрозумілий і зручний графічний інтерфейс; графічний інтерфейс може включати кнопки, поля введення, списки контактів тощо.
* Збереження даних у базі даних: замість збереження контактів у текстовому файлі, ви можете переключитися на використання бази даних. Це дозволяє ефективніше керувати контактами, швидше шукати, сортувати та фільтрувати їх, а також є більш стійким до помилок.
* Синхронізація з іншими пристроями: однією з перспективних можливостей є розробка функції синхронізації, щоб користувачі могли мати доступ до своїх контактів на різних пристроях. Це можна зробити за допомогою хмарного сховища або інших механізмів синхронізації даних.
* Мобільна версія: розробка мобільного додатку для телефонної книги дозволяє користувачам зручно керувати своїми контактами з мобільного пристрою.

Загалом, проект може розвиватися та вдосконалюватися поступово відповідно до потреб та вимог користувачів.

# Висновки

У курсовій роботі було розглянуто реалізацію простої телефонної книги за допомогою мови програмування C#.

Були пояснені основні поняття мови програмування C#, такі як класи, властивості, колекції та маніпуляції з файлами. Телефонна книга зберігає дані контактів у текстових файлах і дозволяє завантажувати та зберігати дані.

Проект має потенціал для подальшого розвитку та вдосконалення. Ідеї для вдосконалення включають редагування, пошук, сортування та групування контактів, графічні інтерфейси та інтеграцію з іншими пристроями.

На закінчення, цей проект демонструє базові знання мови програмування C# та вміння створювати прості додатки з використанням класів та колекцій. Цей код може бути використаний як основа для подальшої розробки більш складних телефонних книг та інших подібних додатків.

Крім того, в процесі розробки студенти набули навичок роботи зі стандартними інструментами C#, такими як читання та запис файлів, маніпулювання колекціями та розробка консольних інтерфейсів користувача.

Загалом, курсова робота демонструє вміння створювати прості додатки на C# та набуття базових знань і навичок роботи з колекціями даних, файлами та класами.

# Перелік джерел

* [C# Офіційний сайт(MSDN)](https://dotnet.microsoft.com/en-us/languages/csharp)
* [C# Programming Guide (MSDN)](https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/?redirectedfrom=MSDN)
* ["Mitigation: New 64-bit JIT Compiler"](https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/mitigation-new-64-bit-jit-compiler).
* [C# Corner](https://www.c-sharpcorner.com/) - Статті, приклади коду та поради з розробки програм на C#
* [Tutorialspoint](https://www.tutorialspoint.com/csharp/) - Навчальний ресурс з курсами та прикладами
* [Microsoft Docs](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/file-system/how-to-read-a-text-file) - Документація про класи List<T> і Dictionary<TKey, TValue>:
* [C# History](https://www.javatpoint.com/csharp-history)
* [C# course](https://itproger.com/ua/course/csharp)
* [Stack Overflow](https://stackoverflow.com/)
* [Hamilton"The A-Z of Programming Languages: C#"](https://www2.computerworld.com.au/article/261958/a-z_programming_languages_c_/?pp)

Додаток (посилання)

<https://github.com/MishaSkovpen/cs-course-project.git>