

## ЛАБОРАТОРНА РОБОТА

**Тема:** Використання LINQ та PLINQ.

**Мета:** Ознайомитись з синтаксисом LINQ та PLINQ. Застосування LINQ та PLINQ в процесі розробки міжплатформного програмного забезпечення на платформі .NET.

### УВАГА!

*Для перевірки наведених прикладів необхідно створити консольне застосування (Console App) C# та вказати цільову платформу (Target Framework) .NET 5. Після чого додати в проект відповідні файли з кодом, що наведені в додатках.*

*Код окремих прикладів необхідно вносити в метод Run() класу ExecuteProgram*

### Теоретичні відомості

Під час виконання програми можуть використовуватись різні дані, які потрібно опрацювати певним чином. До опрацювання можна віднести операції:

1. Керування даними (додавання, видалення, сортування, групування, переміщення);
2. Пошук даних;
3. Модифікація даних (оновлення, зміна типу даних або вмісту, форматування);
4. Отримання метаданих (кількість даних, максимальне/мінімальне значення, перший/останній елементи і т.д.);

Опрацювання даних відбувається за допомогою одних й тих самих конструкцій мови програмування (функції, цикли, умови). Для скорочення написання одноманітного програмного коду, можна застосовувати **автоматизовані засоби опрацювання даних**. Одним з найчастіше використовуваних засобів автоматизації опрацювання даних є **мова запитів**.

**Мова запитів (query language)** — система символів (*алфавіт*) та команд, призначених для опрацювання даних з різних джерел. До мов запитів відносяться:

1. **SQL (Structured Query Language)** — мова запитів до реляційних баз даних. Складається з DDL (Data Definition Language — створення даних), DML (Data Modification Language — модифікація даних), DQL (Data Query Language — отримання даних);
2. **LINQ/PLINQ (Language Integrated Query/Parallel Language Integrated Query)** — сукупність механізмів платформи .NET для маніпулювання даними з різних джерел (колекції та масиви (*LINQ to Objects*), бази даних (*LINQ to DataSet*, *LINQ to SQL*, *LINQ to Entities*), файли (*LINQ to XML*)). Реалізований у вигляді методів розширення (*extension method*). Використання LINQ в кожній окремій .NET-сумісній мові відрізняється. PLINQ — версія LINQ для опрацювання даних в паралельному режимі;
3. **XQuery, XPath** — мови запитів для управління даними з різних джерел (веб-служби, бази даних, файли), які використовують XML-модель (доступ до елементів, структура). XPath, на даний момент є частиною XQuery;
4. Інші.

В даній лабораторній роботі буде розглядатись LINQ to Objects, інші типи запитів будуть розглядатись в наступних роботах.

Основний простір імен для технології LINQ / PLINQ — **System.Linq**. Основний простір імен для LINQ to XML — **System.Xml.Linq**.

В мовах запитів, для управління даними використовуються **вирази (команди)**. Вирази мають синтаксис та логіку опрацювання. Можуть бути строго-типізованими та слабо-типізованими.

Загальна схема використання мови запитів LINQ [2]:

1. Отримати доступ до джерела даних (джерело має реалізовувати інтерфейс IEnumerable або IEnumerable<T>);
2. Сформулювати запит для взаємодії з даними з джерела;

## 3. Виконати запит;

Мова запитів LINQ/PLINQ є **стро́го-типізованою**, тобто параметрами та результатами виконання виразів є дані певного типу даних.

Запит LINQ/PLINQ може виконуватись [1]:

1. **Відкладено (Deferred)** — під час використання результату (зазвичай, колекція IEnumerable). Наприклад, під час використання результату в циклі foreach();
2. **Негайного (Forcing Immediate)** — під час виклику методів, які вимагають отримання даних через запит. Наприклад, ToList<>(), ToArray().

Запити LINQ/PLINQ можна використовувати в методах та властивостях.

Існує 2 підходи до формування запитів:

1. **Методи розширення (extension methods)** — взаємодія з даними відбувається за допомогою послідовних викликів необхідних методів, в які передаються відповідні параметри (зазвичай, лямбда-вирази). Оскільки не всі .NET-сумісні мови підтримують методи розширення, то цей підхід може відрізнитись за синтаксисом (наприклад, в C++/CLI для цього використовуються прямий доступ до методу інтерфейсу IEnumerable, як приклад, Enumerable::FirstOrDefault ). Приклад застосування методів розширення на мові C#:

```
// Вибрати тільки внутрішні документи
// з переліку всіх документів
var internalDocuments =
    _documents.Where((document) => document.Type.Contains($"Внутрішній"));
```

2. **За допомогою синтаксису запитів (query) [3]** — взаємодія з даними відбувається за допомогою послідовності ключових слів (keywords). Цей декларативний підхід побудований на методах розширення і використовується в мові програмування C#. Приклад використання ключових слів:

```
// Вибрати тільки внутрішні документи
// з переліку всіх документів
var internalDocuments = from document in _documents
    where document.Type.Contains($"Внутрішній")
    select document;
```

Перелік ключових слів та аналогічних методів розширення наведено в Таб.1.

Яким з методів формування запитів користуватись, вирішує розробник. Можна комбінувати обидва підходи, але бажано дотримуватись єдиного стилю написання коду.

Використання запитів LINQ/PLINQ може забезпечувати відсутній функціонал, наприклад, для інтерфейсу IEnumerable<> не передбачено властивості Count, але LINQ/PLINQ забезпечує метод розширення Count(), аналогічний одноіменній властивості.

Опис	Ключове слово	Метод розширення
Визначає джерело даних та зміну діапазону (range variable)	<b>from</b> (завжди використовується в парі з ключовим словом <b>in</b> )	—
Фільтрує елементи з джерела даних за вказаною умовою або декількома, перелічених з використанням логічних операцій <b>I (&amp;&amp;)</b> та <b>АБО (  )</b>	<b>where</b>	Where()
Значення яке буде отримано в результаті виконання запиту	<b>select</b>	Select()
Групування елементів згідно певного критерію (зазвичай, властивість об'єкта)	<b>group</b> (використовується в парі з ключовим словом <b>by</b> )	GroupBy() GroupJoin()

Змінна, яка може використовуватись як посилання на результати об'єднання (join), групування (group), чи вибору (select)	<b>Into</b>	—
Відсортувати результат запиту за зростанням або спаданням, використовуючи порівняння за замовчуванням для вказаних даних (зазвичай, Клас Comparer або IComparer, IComparable)	<b>orderby</b> (для сортування за зростанням використовується ключове слово <b>ascending</b> , а за спаданням — <b>descending</b> )	OrderBy() — за зростанням OrderByDescending() — за спаданням
Об'єднання рівних (equality) елементів з двох джерел даних, за вказаним критерієм	<b>join</b> (використовується з ключовими словами <b>in</b> , <b>on</b> , <b>equals</b> )	Join()
Зміна діапазону, для збереження проміжних результатів запитів. Може використовуватись тільки всередині запитів.	<b>let</b>	—

Таблиця 1 — Перелік ключових слів LINQ

Всі методи розширення LINQ/PLINQ умовно можна поділити (детально описані в Таб.2):

1. **Вибірки (фільтрації)** — отримання елементів за заданими умовами;
2. **Зміни порядку (групування, сортування, додавання)** — управління позиціями елементів всередині колекції або декількох колекцій;
3. **Агрегації** — отримання відомостей про елементи колекції;
4. **Управління запитами** — спеціальні методи розширення, які вказують середовищу виконання, як потрібно опрацювати запит.

Для перевірки прикладів коду з Таб.2 необхідно код прикладу розмістити в методі Run() з Додатку 4.

Назва	Опис методу розширення	Результат виконання
<b>CurrentType</b> – тип даних поточного елемента колекції <b>AnotherType</b> — тип даних після перетворень (якщо вони використовуються)		
<b>ВИБІРКИ (ФІЛЬТРАЦІЇ)</b>		
Select [4]	Над кожним елементом колекції можуть бути проведені перетворення (наприклад, в інший тип даних) або отримати конкретну властивість поточного об'єкта.	IEnumerable<CurrentType> (якщо не було ніяких перетворень) або IEnumerable<AnotherType> (якщо було виконано перетворення)
Приклад використання Select	<pre>var selectedElements = _archivedDocuments?.Select(doc =&gt; {     // Якщо документ не архівний, то зробити його архівним     if (!(doc is ArchivedDocument))         return ArchivedDocument.ArchiveDocument(doc?.Title);     else         return doc; });  // Вивести вибрані елементи Out(selectedElements);</pre>	

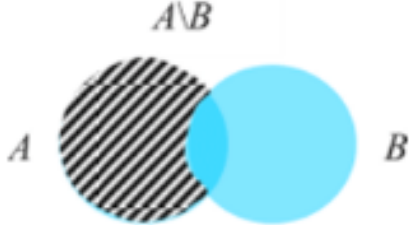
Результат виконання прикладу Select	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Архівний документ №:      0bf34353-7ef4-4e20-b444-cad3c4542ee5 Постанова з міністерства      від      03.04.2022  Архівний документ №:      26031f09-ea64-4142-933f-53ddb265cee2 Про дохід      від      06.04.2022  Архівний документ №:      c19b3b7b-39a1-4f14-88f1-3fd9cf8ebe6c Наказ з міністерства      від      11.04.2022  Архівний документ №:      032a39c6-3c24-407a-bc70-328b9caf9249 Соціальні виплати      від      03.04.2022  Архівний документ №:      f37cea91-1b88-4238-8430-ce5b7df893a0 Внутрішнє розпорядження від      20.04.2022  Архівний документ №:      6169b304-dc75-4fd1-a7e7-ceb04269e587 Оплата контракту      від      09.04.2022 </pre>	
SelectMany [5]	<p>Кожен елемент колекції перетворюється в тип IEnumerable&lt;CurrentType&gt;, і об'єднується в одну велику послідовність типу IEnumerable&lt;IEnumerable&lt;CurrentType&gt;&gt;. Застосовується при роботі з вкладеними в об'єкт колекціями.</p>	<p>IEnumerable&lt;IEnumerable&lt;CurrentType&gt;&gt; (якщо не було ніяких перетворень) або IEnumerable&lt;IEnumerable&lt;AnotherType&gt;&gt; (якщо було виконано перетворення)</p>
Приклад використання SelectMany	<pre> var selectedElements = _archivedDocuments?.SelectMany(doc =&gt; {     // інформація про версію поточного документа     // якщо в документа декілька версій, то вони будуть     // записані у вигляді окремої колекції     var documentVerInfo = doc?.Versions.Select(version =&gt;          \$"{Environment.NewLine}" +         \$"Назва документа:\t{doc?.Title}" +         \$"{Environment.NewLine}" +         \$"ID документа:\t{doc?.Id}" +         \$"{Environment.NewLine}" +         \$"Версія документа: {version}" +         \$"{Environment.NewLine}");      return documentVerInfo;  });  // Вивести вибрані елементи Out(selectedElements); </pre>	

Результат виконання прикладу SelectMany	<pre> 1.) Виконати програму; 2.) Завершити роботу; 1  Назва документа:      Постанова з міністерства ID документа:   30e0b491-050b-463f-9196-82594b836649 Версія документа: v.1.0  Назва документа:      Постанова з міністерства ID документа:   30e0b491-050b-463f-9196-82594b836649 Версія документа: v.2.0  Назва документа:      Про дохід ID документа:   bac180af-b017-49da-bd36-e8f9f65af2da Версія документа: v.1.0  Назва документа:      Про дохід ID документа:   bac180af-b017-49da-bd36-e8f9f65af2da Версія документа: v.2.0  Назва документа:      Наказ з міністерства ID документа:   eff926b2-744d-4c4d-ae19-30b9f0ef48c3 Версія документа: v.1.0  Назва документа:      Наказ з міністерства ID документа:   eff926b2-744d-4c4d-ae19-30b9f0ef48c3 Версія документа: v.2.0  Назва документа:      Соціальні виплати ID документа:   68c7ca3a-09b6-4152-ad76-8cb8fe3e4c54 Версія документа: v.1.0  Назва документа:      Внутрішнє розпорядження ID документа:   2076e362-360a-4916-9b9d-216f3fe39248 Версія документа: v.1.0  Назва документа:      Внутрішнє розпорядження ID документа:   2076e362-360a-4916-9b9d-216f3fe39248 Версія документа: v.2.0  Назва документа:      Оплата контракту ID документа:   e435c36f-ec14-4900-bd45-af069ca73c10 Версія документа: v.1.0 </pre>
Where [6]	<div>Визначається умова відбору елементів з колекції</div> <div>IEnumerable&lt;CurrentType&gt;</div>
Приклад виконання Where	<pre> // Вибрати накази серед архівних документів var allArchivedOrders = _archivedDocuments?.Where(doc =&gt; {     return doc.Title.Contains(\$"наказ",                              StringComparison.CurrentCultureIgnoreCase); }); // Вивести перелік знайдених наказів Out(allArchivedOrders); </pre>

Результат виконання прикладу Where	Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Вхідний документ №: ccf3e71a-179a-49e0-a780-1665a6b8b9a5 Наказ з міністерства від 01.04.2022	
OfType [7]	Пошук елементів відбувається за вказаним типом даних	IEnumerable<CurrentType>
Приклад виконання OfType	<pre>// Отримати документи, які ще не здані до архіву var notArchivedDocuments = _archivedDocuments?.OfType&lt;Document&gt;();  // Вивести перелік документів, // які ще не здано до архіву Out(notArchivedDocuments);</pre>	
Результат виконання прикладу OfType	Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Вхідний документ №: 1f2ce824-8645-4e9c-a99b-c8e22af595e2 Постанова з міністерства від 20.04.2022  Вхідний документ №: a3446355-3de1-40ab-afd6-89a15b3578e9 Наказ з міністерства від 01.04.2022  Внутрішній документ №: a1d63804-c324-4ec9-bd5b-a9758b8c5d68 Внутрішнє розпорядження від 14.04.2022	
Cast [9]	Перетворює кожен елемент колекції у вказаний тип даних	IEnumerable<AnotherType>
Приклад неявного виконання Cast	<pre>// Отримати документ з архіву var documentsFromArchive = _archivedDocuments?.Select(doc =&gt; {     IDocument retrievedDoc = null;      // Якщо документ архівний, то     // зробити його внутрішнім документом     // для подальшого використання     if (doc is ArchivedDocument)     {         // Використовується приведення типів (Cast),         // оскільки в колекції вони зберігаються         // в узагальненому вигляді (інтерфейс IDocument)         retrievedDoc = (Document)(doc as ArchivedDocument);     }      return retrievedDoc; });  // Вивести перелік отриманих документів Out(documentsFromArchive);</pre>	

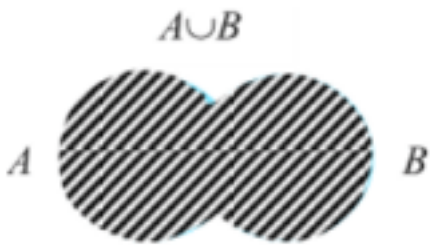


Результат виконання прикладу Cast	Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Внутрішній документ №: 914250e8-f53a-48d2-903b-523c339de77e Про дохід від 16.04.2022 Внутрішній документ №: 9774aaba-7882-43f7-8149-ba33b31a17c2 Соціальні виплати від 26.04.2022 Внутрішній документ №: 3866f9bc-b56e-44a6-91ea-8cb46e9ddeb4 Оплата контракту від 29.04.2022	
Join [10]	Знаходить елементи з обох колекцій, які співпадають за вказаною властивістю. Наприклад, співпадає Id. Аналог Inner Join в SQL.	IEnumerable<CurrentType>
Приклад виконання Join	<pre>// Максимальний час зберігання документа за його типом var documentStoringDeadline = new[] {     new {         // Архівний документ         Type = ArchivedDocument.ArchivedTypeTitle,         // Зберігається 5 років         Duration = 5, },     new {         // Внутрішній документ         Type = Document.InternalDocumentTypeTitle,         // Зберігається 1 рік         Duration = 1, },     new {         // Вхідний документ         Type = Document.ExternalDocumentTypeTitle,         // Зберігається 2 роки         Duration = 2, }, }; // Зпівставлення типу документа та його часу зберігання var documentsStoringDeadline = _newDocuments?.Join(documentStoringDeadline,     // ключі повинні співпадати за типом даних     newDoc =&gt; newDoc?.Type,     // ключі повинні співпадати за типом даних     docType =&gt; docType?.Type,     (newDoc, docType) =&gt;     {         return new {             Title = newDoc?.Title,             DocTypeTitle = newDoc?.Type,             StorageDuration = docType?.Duration, };     }); // Вивести інформація про час зберігання документа foreach(var doc in documentsStoringDeadline) {     Console.WriteLine(\$"-----");     Console.WriteLine(\$"Назва документа:\t{doc?.Title}");     Console.WriteLine(\$"Тип документа:\t{doc?.DocTypeTitle}");     Console.WriteLine(\$"Час зберігання (в роках):\t{doc?.StorageDuration}"); } }</pre>	

Результат виконання прикладу Join	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 ----- Назва документа:      Розпорядження з міністерства Тип документа:   Вхідний Час зберігання (в роках):      2 ----- Назва документа:      Витрати на конференції Тип документа:   Архівний Час зберігання (в роках):      5 ----- Назва документа:      Наказ декана Тип документа:   Внутрішній Час зберігання (в роках):      1 ----- Назва документа:      Річні витрати на зарплатню Тип документа:   Архівний Час зберігання (в роках):      5 ----- Назва документа:      Наказ ректора Тип документа:   Внутрішній Час зберігання (в роках):      1 ----- Назва документа:      Річні витрати на господарську частину Тип документа:   Архівний Час зберігання (в роках):      5 </pre>
Ексерпт [11]	<p>Знаходить елементи які присутні в першій колекції, але відсутні в другій.</p> <p><small>Ексерпт</small></p> <p>Ексерпт можна розглядати як операцію різниці множин (A,B):</p> 
Приклад виконання Ексерпт	<pre> // Згенерувати єдиний ідентифікатор var uniqueId = Guid.NewGuid();  // Задати однаковий ідентифікатор для документів // з різних колекцій, щоб можна було отримати різницю (_newDocuments?.ElementAt(1) as DocumentBase)?.UpdateId(uniqueId); (_newDocuments?.ElementAt(3) as DocumentBase)?.UpdateId(uniqueId); (_newDocuments?.ElementAt(5) as DocumentBase)?.UpdateId(uniqueId); (_archivedDocuments?.ElementAt(4) as DocumentBase)?.UpdateId(uniqueId); (_archivedDocuments?.ElementAt(3) as DocumentBase)?.UpdateId(uniqueId); (_archivedDocuments?.ElementAt(5) as DocumentBase)?.UpdateId(uniqueId);  // Отримати перелік документів, які наявні тільки // в переліку нових документів // Для роботи Ексерпт обов'язково в користувацьких // типах даних (класах) повинні бути перевизначені методи: // bool Equals(object obj) та int GetHashCode() var difference = _newDocuments?.Except(_archivedDocuments);  // Відобразити таблицю з документами </pre>

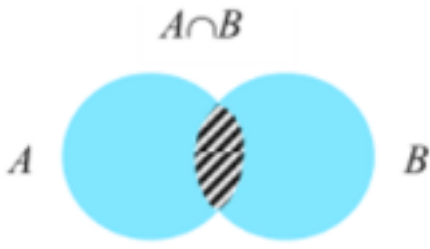


	<pre>ShowAllDocuments();  // Вивести тільки нові документи, які ще ні // разу не були в архіві Out(difference);</pre>	
Результат виконання прикладу Excerpt	<pre>2.) Завершити роботу; 1 ----- Нові документи   Архівні документи ----- Вхідний документ №:   Вхідний документ №: 4ce92c78-ffba-47de-8879-c6360a66b020   377e6d55-de84-49de-acbb-c9a4ff1f1664 Розпорядження з міністерства   Постанова з міністерства від 08.04.2022   від 20.04.2022 ----- Архівний документ №:   Архівний документ №: fa9d96e7-17b2-4461-a9b5-97253e9976d6   9758372c-1621-4f60-96b8-62df0029ff2a Витрати на конференції   Про дохід від 17.04.2022   від 21.04.2022 ----- Внутрішній документ №:   Вхідний документ №: 8b8efc69-f979-4b6e-863c-e5ee475c7e15   02d6e8a9-3535-479e-99e6-0e33a12d613b Наказ декана   Наказ з міністерства від 03.04.2022   від 03.04.2022 ----- Архівний документ №:   Архівний документ №: fa9d96e7-17b2-4461-a9b5-97253e9976d6   fa9d96e7-17b2-4461-a9b5-97253e9976d6 Річні витрати на зарплатню   Соціальні виплати від 27.04.2022   від 02.05.2022 ----- Внутрішній документ №:   Внутрішній документ №: 54ef6e98-94f1-42f0-acdc-2a62da16d23f   fa9d96e7-17b2-4461-a9b5-97253e9976d6 Наказ ректора   Внутрішнє розпорядження від 08.04.2022   від 27.04.2022 ----- Архівний документ №:   Архівний документ №: fa9d96e7-17b2-4461-a9b5-97253e9976d6   fa9d96e7-17b2-4461-a9b5-97253e9976d6 Річні витрати на господарську частину   Оплата контракту від 02.05.2022   від 28.04.2022 ----- Вхідний документ №: 4ce92c78-ffba-47de-8879-c6360a66b020 Розпорядження з міністерства від 08.04.2022 Внутрішній документ №: 8b8efc69-f979-4b6e-863c-e5ee475c7e15 Наказ декана від 03.04.2022 Внутрішній документ №: 54ef6e98-94f1-42f0-acdc-2a62da16d23f Наказ ректора від 08.04.2022</pre>	
Union [12]	Знаходить всі елементи двох однотипних колекцій. Якщо елементи повторюються, то буде включено тільки 1 раз	IEnumerable<CurrentType>
	Union можна розглядати як операцію об'єднання множин (A,B):	

		
Приклад виконання Union	<pre>// Згенерувати єдиний ідентифікатор var uniqueId = Guid.NewGuid();  // Задати однаковий ідентифікатор для документів // з різних колекцій, щоб можна було отримати дублікати (_newDocuments?.ElementAt(1) as DocumentBase)?.UpdateId(uniqueId); (_newDocuments?.ElementAt(3) as DocumentBase)?.UpdateId(uniqueId); (_newDocuments?.ElementAt(5) as DocumentBase)?.UpdateId(uniqueId); (_archivedDocuments?.ElementAt(4) as DocumentBase)?.UpdateId(uniqueId); (_archivedDocuments?.ElementAt(3) as DocumentBase)?.UpdateId(uniqueId); (_archivedDocuments?.ElementAt(5) as DocumentBase)?.UpdateId(uniqueId);  // Отримати перелік унікальних документів, // які наявні в обох переліках документів // Для роботи Union обов'язково в користувацьких // типах даних (класах) повинні бути перевизначені методи: // bool Equals(object obj) та int GetHashCode() var difference = _newDocuments?.Union(_archivedDocuments);  // Відобразити таблицю з документами ShowAllDocuments();  // Вивести всі унікальні документи // з обох колекцій Out(difference);</pre>	

Результат  
виконання  
прикладу Union

Нові документи	Архівні документи
Вхідний документ №: fac3a065-e123-40f1-9963-058e271c70bf Розпорядження з міністерства від 16.04.2022	Вхідний документ №: c996b787-78ed-4ebc-9434-ad87842b65f1 Постанова з міністерства від 06.04.2022
Архівний документ №: 04c76cf7-faeb-4c5a-9003-345d71722636 Витрати на конференції від 01.05.2022	Архівний документ №: 9db97927-5e70-409b-ba2f-1828f0e34d25 Про дохід від 26.04.2022
Внутрішній документ №: 94c58aeb-a399-4b68-a912-64f347acc107 Наказ декана від 03.04.2022	Вхідний документ №: 152e0e72-96f0-449d-a377-a1f7a82fd78f Наказ з міністерства від 03.04.2022
Архівний документ №: 04c76cf7-faeb-4c5a-9003-345d71722636 Річні витрати на зарплатню від 15.04.2022	Архівний документ №: 04c76cf7-faeb-4c5a-9003-345d71722636 Соціальні виплати від 01.05.2022
Внутрішній документ №: 3070853a-0e6b-485c-8aad-2454b4ef6a89 Наказ ректора від 24.04.2022	Внутрішній документ №: 04c76cf7-faeb-4c5a-9003-345d71722636 Внутрішнє розпорядження від 25.04.2022
Архівний документ №: 04c76cf7-faeb-4c5a-9003-345d71722636 Річні витрати на господарську частину від 04.04.2022	Архівний документ №: 04c76cf7-faeb-4c5a-9003-345d71722636 Оплата контракту від 10.04.2022
-----	
Вхідний документ №:	fac3a065-e123-40f1-9963-058e271c70bf
Розпорядження з міністерства	від 16.04.2022
Архівний документ №:	04c76cf7-faeb-4c5a-9003-345d71722636
Витрати на конференції	від 01.05.2022
Внутрішній документ №:	94c58aeb-a399-4b68-a912-64f347acc107
Наказ декана	від 03.04.2022
Внутрішній документ №:	3070853a-0e6b-485c-8aad-2454b4ef6a89
Наказ ректора	від 24.04.2022
Вхідний документ №:	c996b787-78ed-4ebc-9434-ad87842b65f1
Постанова з міністерства	від 06.04.2022
Архівний документ №:	9db97927-5e70-409b-ba2f-1828f0e34d25
Про дохід	від 26.04.2022
Вхідний документ №:	152e0e72-96f0-449d-a377-a1f7a82fd78f
Наказ з міністерства	від 03.04.2022

Intersect [13]	<p>Знаходить однакові елементи двох однотипних колекцій. Дублікати видаляються.</p> <p>Intersect можна розглядати як операцію перетину множин (A,B):</p> 	IEnumerable<CurrentType>
Приклад виконання Intersect	<pre>// Згенерувати єдиний ідентифікатор var uniqueId = Guid.NewGuid();  // Задати однаковий ідентифікатор для документів // з різних колекцій, щоб можна було отримати дублікати (_newDocuments?.ElementAt(1) as DocumentBase)?.UpdateId(uniqueId); (_newDocuments?.ElementAt(3) as DocumentBase)?.UpdateId(uniqueId); (_newDocuments?.ElementAt(5) as DocumentBase)?.UpdateId(uniqueId); (_archivedDocuments?.ElementAt(4) as DocumentBase)?.UpdateId(uniqueId); (_archivedDocuments?.ElementAt(3) as DocumentBase)?.UpdateId(uniqueId); (_archivedDocuments?.ElementAt(5) as DocumentBase)?.UpdateId(uniqueId);  // Отримати перелік однакових документів, // які наявні в обох переліках документів // Для роботи Intersect обов'язково в користувацьких // типах даних (класах) повинні бути перевизначені методи: // bool Equals(object obj) та int GetHashCode() var difference = _newDocuments?.Intersect(_archivedDocuments);  // Відобразити таблицю з документами ShowAllDocuments();  // Вивести всі однакових документи // з обох колекцій Out(difference);</pre>	

Результат виконання прикладу Intersect	<table border="1"> <thead> <tr> <th>Нові документи</th><th>Архівні документи</th></tr> </thead> <tbody> <tr> <td>Вхідний документ №:</td><td>Вхідний документ №:</td></tr> <tr> <td>cad732cc-5e04-4bec-b48b-ca5224b8152e</td><td>451a5ed7-b2a9-4737-a165-f4d6c7dd7dc9</td></tr> <tr> <td>Розпорядження з міністерства</td><td>Постанова з міністерства</td></tr> <tr> <td>від 04.04.2022</td><td>від 16.04.2022</td></tr> <tr> <td>Архівний документ №:</td><td>Архівний документ №:</td></tr> <tr> <td>cb60d035-dce9-43d3-a2d4-a4164edde642</td><td>1e93ac16-877f-4f02-bece-5842852d0c10</td></tr> <tr> <td>Витрати на конференції</td><td>Про дохід</td></tr> <tr> <td>від 23.04.2022</td><td>від 08.04.2022</td></tr> <tr> <td>Внутрішній документ №:</td><td>Вхідний документ №:</td></tr> <tr> <td>c564e3c4-194b-45d7-beaa-6db7e59d5b01</td><td>b548cbac-0d1b-4512-9447-91165ec46512</td></tr> <tr> <td>Наказ декана</td><td>Наказ з міністерства</td></tr> <tr> <td>від 03.04.2022</td><td>від 03.04.2022</td></tr> <tr> <td>Архівний документ №:</td><td>Архівний документ №:</td></tr> <tr> <td>cb60d035-dce9-43d3-a2d4-a4164edde642</td><td>cb60d035-dce9-43d3-a2d4-a4164edde642</td></tr> <tr> <td>Річні витрати на зарплатню</td><td>Соціальні виплати</td></tr> <tr> <td>від 21.04.2022</td><td>від 21.04.2022</td></tr> <tr> <td>Внутрішній документ №:</td><td>Внутрішній документ №:</td></tr> <tr> <td>e08461ba-ba4e-458a-9126-3478701f34dc</td><td>cb60d035-dce9-43d3-a2d4-a4164edde642</td></tr> <tr> <td>Наказ ректора</td><td>Внутрішнє розпорядження</td></tr> <tr> <td>від 21.04.2022</td><td>від 09.04.2022</td></tr> <tr> <td>Архівний документ №:</td><td>Архівний документ №:</td></tr> <tr> <td>cb60d035-dce9-43d3-a2d4-a4164edde642</td><td>cb60d035-dce9-43d3-a2d4-a4164edde642</td></tr> <tr> <td>Річні витрати на господарську частину</td><td>Оплата контракту</td></tr> <tr> <td>від 10.04.2022</td><td>від 25.04.2022</td></tr> <tr> <td>Архівний документ №:</td><td>Архівний документ №:</td></tr> <tr> <td>cb60d035-dce9-43d3-a2d4-a4164edde642</td><td>cb60d035-dce9-43d3-a2d4-a4164edde642</td></tr> <tr> <td>Витрати на конференції</td><td>Витрати на конференції</td></tr> <tr> <td>від 23.04.2022</td><td>від 23.04.2022</td></tr> </tbody> </table>	Нові документи	Архівні документи	Вхідний документ №:	Вхідний документ №:	cad732cc-5e04-4bec-b48b-ca5224b8152e	451a5ed7-b2a9-4737-a165-f4d6c7dd7dc9	Розпорядження з міністерства	Постанова з міністерства	від 04.04.2022	від 16.04.2022	Архівний документ №:	Архівний документ №:	cb60d035-dce9-43d3-a2d4-a4164edde642	1e93ac16-877f-4f02-bece-5842852d0c10	Витрати на конференції	Про дохід	від 23.04.2022	від 08.04.2022	Внутрішній документ №:	Вхідний документ №:	c564e3c4-194b-45d7-beaa-6db7e59d5b01	b548cbac-0d1b-4512-9447-91165ec46512	Наказ декана	Наказ з міністерства	від 03.04.2022	від 03.04.2022	Архівний документ №:	Архівний документ №:	cb60d035-dce9-43d3-a2d4-a4164edde642	cb60d035-dce9-43d3-a2d4-a4164edde642	Річні витрати на зарплатню	Соціальні виплати	від 21.04.2022	від 21.04.2022	Внутрішній документ №:	Внутрішній документ №:	e08461ba-ba4e-458a-9126-3478701f34dc	cb60d035-dce9-43d3-a2d4-a4164edde642	Наказ ректора	Внутрішнє розпорядження	від 21.04.2022	від 09.04.2022	Архівний документ №:	Архівний документ №:	cb60d035-dce9-43d3-a2d4-a4164edde642	cb60d035-dce9-43d3-a2d4-a4164edde642	Річні витрати на господарську частину	Оплата контракту	від 10.04.2022	від 25.04.2022	Архівний документ №:	Архівний документ №:	cb60d035-dce9-43d3-a2d4-a4164edde642	cb60d035-dce9-43d3-a2d4-a4164edde642	Витрати на конференції	Витрати на конференції	від 23.04.2022	від 23.04.2022
Нові документи	Архівні документи																																																										
Вхідний документ №:	Вхідний документ №:																																																										
cad732cc-5e04-4bec-b48b-ca5224b8152e	451a5ed7-b2a9-4737-a165-f4d6c7dd7dc9																																																										
Розпорядження з міністерства	Постанова з міністерства																																																										
від 04.04.2022	від 16.04.2022																																																										
Архівний документ №:	Архівний документ №:																																																										
cb60d035-dce9-43d3-a2d4-a4164edde642	1e93ac16-877f-4f02-bece-5842852d0c10																																																										
Витрати на конференції	Про дохід																																																										
від 23.04.2022	від 08.04.2022																																																										
Внутрішній документ №:	Вхідний документ №:																																																										
c564e3c4-194b-45d7-beaa-6db7e59d5b01	b548cbac-0d1b-4512-9447-91165ec46512																																																										
Наказ декана	Наказ з міністерства																																																										
від 03.04.2022	від 03.04.2022																																																										
Архівний документ №:	Архівний документ №:																																																										
cb60d035-dce9-43d3-a2d4-a4164edde642	cb60d035-dce9-43d3-a2d4-a4164edde642																																																										
Річні витрати на зарплатню	Соціальні виплати																																																										
від 21.04.2022	від 21.04.2022																																																										
Внутрішній документ №:	Внутрішній документ №:																																																										
e08461ba-ba4e-458a-9126-3478701f34dc	cb60d035-dce9-43d3-a2d4-a4164edde642																																																										
Наказ ректора	Внутрішнє розпорядження																																																										
від 21.04.2022	від 09.04.2022																																																										
Архівний документ №:	Архівний документ №:																																																										
cb60d035-dce9-43d3-a2d4-a4164edde642	cb60d035-dce9-43d3-a2d4-a4164edde642																																																										
Річні витрати на господарську частину	Оплата контракту																																																										
від 10.04.2022	від 25.04.2022																																																										
Архівний документ №:	Архівний документ №:																																																										
cb60d035-dce9-43d3-a2d4-a4164edde642	cb60d035-dce9-43d3-a2d4-a4164edde642																																																										
Витрати на конференції	Витрати на конференції																																																										
від 23.04.2022	від 23.04.2022																																																										
Take [14]	Отримати з колекції вказану кількість елементів. Відлік починається з першого елемента колекції.																																																										
Приклад виконання Take	<pre>// Отримати 4 нових документа var documents = _newDocuments?.Take(4);  // Вивести перших 4 документа Out(documents);</pre>																																																										

Результат виконання прикладу Take	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Вхідний документ №:      ebe2af18-25b7-4e65-9829-02f252f88c49 Розпорядження з міністерства від      30.04.2022  Архівний документ №:      6771c4cd-ddf7-4de0-81b3-4694255bb06a Витрати на конференції від      21.04.2022  Внутрішній документ №:      8a337c06-32e3-459d-a16d-355282519779 Наказ декана від      03.04.2022  Архівний документ №:      f263c048-daeef-4632-a6b0-2692aa449b99 Річні витрати на зарплатню від      04.04.2022 </pre>	
TakeWhile [15]	Отримати елементи поки виконується умова відбору елементів. Відлік починається з першого елемента колекції.	IEnumerable<CurrentType>
Приклад виконання TakeWhile	<pre> var date = new DateTime(DateTime.Now.Year,                         6,                         DateTime.Now.Day);  _newDocuments.ElementAt(0).CreationDate = date; _newDocuments.ElementAt(1).CreationDate = date;  // Вибрати всі нові документи створені в // червні поточного року var documents = _newDocuments?.TakeWhile(doc =&gt; {     if (doc?.CreationDate.Month == 6)         return true;     else         return false; });  // Вивести документи Out(documents); </pre>	
Результат виконання прикладу TakeWhile	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Вхідний документ №:      d8856998-c2d7-4729-bf8f-569868839b5b Розпорядження з міністерства від      03.06.2022  Архівний документ №:      2a4c65e5-36a1-449a-96ea-6d82ab34f38a Витрати на конференції від      03.06.2022 </pre>	
Skip [16]	Пропустити вказану кількість елементів колекції. Відлік починається з першого елемента.	IEnumerable<CurrentType>
Приклад виконання Skip	<pre> // Пропустити 4 документи var skippedDocs = _newDocuments.Skip(4);  Out(skippedDocs); </pre>	



Результат виконання прикладу Skip	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Внутрішній документ №: 0758e300-5983-4198-990a-6be8c84a6f83 Наказ ректора від 20.04.2022  Архівний документ №: 2dccdd66-88f8-45eb-99e3-e37e1f029a73 Річні витрати на господарську частину від 29.04.2022 </pre>	
SkipWhile [17]	Пропустити елементи колекції поки не буде знайдено перший елемент, що відповідає умові. Якщо елемент знайдено, повертаються всі елементи, які йдуть після знайденого. Відлік починається з першого елемента.	IEnumerable<CurrentType>
Приклад виконання SkipWhile	<pre> var date = new DateTime(DateTime.Now.Year,                         6,                         DateTime.Now.Day);  _newDocuments.ElementAt(2).CreationDate = date;  // Пропускати документи, поки не буде // знайдено документ створений в червні var documents = _newDocuments?.TakeWhile(doc =&gt; {     if (doc?.CreationDate.Month != 6)         return true;     else         return false; });  // Вивести документи Out(documents); </pre>	
Результат виконання прикладу SkipWhile	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Вхідний документ №: 677afccc-b21d-41b7-bcca-9f260bf1cbc1 Розпорядження з міністерства від 06.04.2022  Архівний документ №: ecd6c10e-86b9-4706-9e57-70ef1852269f Витрати на конференції від 10.04.2022 </pre>	
First [18]	Отримати перший елемент в колекції або перший елемент, який відповідає умові. Якщо послідовність порожня або відсутній елемент — буде згенеровано виключення.	CurrentType
Приклад виконання First	<pre> // Отримати перший елемент колекції var firstDocument = _newDocuments?.First();  // Отримати перший елемент який відповідає умові var reportdDocument = _archivedDocuments?.First(doc =&gt;     doc.Title.Contains("Оплата"), </pre>	

	<pre>StringComparison.CurrentCultureIgnoreCase));  Console.WriteLine(\$"Перший документ колекції"); Console.WriteLine(\$"{firstDocument}");  Console.WriteLine(\$"Документ, який відповідає умові"); Console.WriteLine(\$"{reportdDocument}");</pre>	
Результат виконання прикладу First	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Перший документ колекції Вхідний документ №:      16f6f611-7681-4a6b-9e1c-556328e23aba Розпорядження з міністерства від      30.04.2022  Документ, який відповідає умові Архівний документ №:      88893884-1766-454d-be47-f2c3d88af3c8 Оплата контракту від      18.04.2022</pre>	
FirstOrDefault [19]	Отримати перший елемент послідовності, якщо елемент відсутній повернути значення за замочування (ключове слово default). Якщо колекція порожня – виключення генеруватись <b>не буде</b> . Є безпечною версією First.	CurrentType
Приклад виконання FirstOrDefault	<pre>// Перелік видалених документів var deletedDocs = new List&lt;IDocument&gt;();  // Перший документ зі списку нових документів var first = _newDocuments?.FirstOrDefault();  Console.WriteLine(\$"Перший видалений документ:"); Console.WriteLine(\$"{deletedDocs?.FirstOrDefault()}"); Console.WriteLine(\$"Перший новий документ:"); Console.WriteLine(\$"{first}");</pre>	
Результат виконання прикладу FirstOrDefault	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Перший видалений документ:  Перший новий документ: Вхідний документ №:      bae05f41-3d40-4b60-85c1-2fd27061281b Розпорядження з міністерства від      25.04.2022</pre>	
Single [20]	Вибрати єдиний елемент з колекції, який відповідає умові. Якщо умова не задана, і колекція складається з одного елемента, то Single працює як First. Якщо колекція порожня або елементів, що відповідають	CurrentType

	умові більше одного – буде згенеровано виключення.	
Приклад виконання Single	<pre>// Отримати єдиний документ про річні витрати var salary = _newDocuments?.Single(doc =&gt; doc?.Title ==     "Річні витрати на зарплатню");  Console.WriteLine(\$"Звіт про річні витрати на зарплатню:"); Console.WriteLine(\$"{salary}");</pre>	
Результат виконання прикладу Single	<pre>Р. Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Звіт про річні витрати на зарплатню: Архівний документ №: 0516e4e8-df16-440e-b2e9-f5f03f5985f7 Річні витрати на зарплатню від 17.04.2022</pre>	
SingleOrDefault [21]	<p>Вибрати єдиний елемент з колекції, який відповідає умові. Якщо умова не задана, і колекція складається з одного елемента, то SingleOrDefault працює як First.</p> <p>Якщо колекція порожня або елементів, що відповідають умові більше одного – буде повернуто значення за замовчуванням (тобто default(CurrentType)).</p> <p>Є безпечною версією Single.</p>	CurrentType
Приклад виконання SingleOrDefault	<pre>// Отримати єдиний документ про надходження отримання гранту var grant = _newDocuments?.SingleOrDefault(doc =&gt;     doc.Title.Contains("Грант"));  // Отримати єдиний документ про річні витрати var salary = _newDocuments?.Single(doc =&gt; doc?.Title ==     "Річні витрати на зарплатню");  Console.WriteLine(\$"Документ на отримання гранта:"); Console.WriteLine(\$"{grant}"); Console.WriteLine(\$"Звіт про річні витрати на зарплатню:"); Console.WriteLine(\$"{salary}");</pre>	
Результат виконання прикладу SingleOrDefault	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Документ на отримання гранта:  Звіт про річні витрати на зарплатню: Архівний документ №: a3d18c64-5b8e-4fa7-a8c8-c00cbf30fa80 Річні витрати на зарплатню від 02.05.2022</pre>	
ElementAt [22]	Отримує елемент з колекції за вказаним індексом (навіть, якщо в колекції не підтримується прямий доступ за	CurrentType

	індексом). Індекс починається з нуля. Якщо вказано неправильний індекс – буде згенеровано виключення.	
Приклад виконання ElementAt	<pre>// Отримати третій документ надісланий на опрацювання var third = _newDocuments?.ElementAt(2);  Console.WriteLine(\$"Третій документ надісланий на опрацювання:"); Console.WriteLine(\$"{third}");</pre>	
Результат виконання прикладу ElementAt	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Третій документ надісланий на опрацювання: Внутрішній документ №: 061e2f3f-6add-4af2-ab25-e7f7045233b1 Наказ декана від 04.04.2022</pre>	
ElementAtOrDefault [23]	Отримує елемент з колекції за вказаним індексом (навіть, якщо в колекції не підтримується прямий доступ за індексом). Індекс починається з нуля. Якщо вказано неправильний індекс – буде повернуто значення за замовчуванням (тобто default(CurrentType)). Є безпечною версією ElementAt.	CurrentType
Приклад виконання ElementAtOrDefault	<pre>// Отримати останній документ, надісланий на опрацювання var last = _newDocuments?.ElementAtOrDefault(6);  // Отримати третій документ надісланий на опрацювання var third = _newDocuments?.ElementAt(2);  Console.WriteLine(\$"Останній документ, надісланий на опрацювання:"); Console.WriteLine(\$"{last}"); Console.WriteLine(\$"Третій документ надісланий на опрацювання:"); Console.WriteLine(\$"{third}");</pre>	
Результат виконання прикладу ElementAtOrDefault	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Останній документ, надісланий на опрацювання: Третій документ надісланий на опрацювання: Внутрішній документ №: af121e10-9aaf-41ac-9466-abf0fb41b817 Наказ декана від 04.04.2022</pre>	
Last [24]	Отримати останній елемент колекції. Якщо колекція порожня — буде згенеровано виключення.	CurrentType
Приклад виконання Last	<pre>// Отримати третій документ надісланий на опрацювання var lastNewDocument = _newDocuments?.Last();</pre>	

	<pre>Console.WriteLine(\$"Останній видалений документ:"); Console.WriteLine(\$"{lastNewDocument}");</pre>	
Результат виконання прикладу Last	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Останній видалений документ: Архівний документ №: aa192efa-ee21-4623-9ab7-382c11826f09 Річні витрати на господарську частину від 26.04.2022</pre>	
LastOrDefault [25]	Отримати останній елемент колекції. Якщо колекція порожня — буде повернуто значення за замовчуванням (тобто default(CurrentType)). Є безпечною версією Last.	CurrentType
Приклад виконання LastOrDefault	<pre>var deletedDocs = new List&lt;IDocument&gt;(); // Отримати останній документ, надісланий на опрацювання var lastDeletedDocument = deletedDocs?.LastOrDefault();  // Отримати третій документ надісланий на опрацювання var lastNewDocument = _newDocuments?.Last();  Console.WriteLine(\$"Останній видалений документ:"); Console.WriteLine(\$"{lastDeletedDocument}"); Console.WriteLine(\$"Останній документ надісланий на опрацювання:"); Console.WriteLine(\$"{lastNewDocument}");</pre>	
Результат виконання прикладу LastOrDefault	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Останній видалений документ:  Останній документ надісланий на опрацювання: Архівний документ №: c24d774b-1b66-47d9-b7c4-63d61a22985f Річні витрати на господарську частину від 15.04.2022</pre>	
ЗМІНИ ПОРЯДКУ (ГРУПУВАННЯ, СОРТУВАННЯ, ДОДАВАННЯ)		
Append [26]	Додати елемент в кінець колекції	IEnumerable<CurrentType>
Приклад виконання Append	<pre>// Додати новий документ, та зберегти його в кінець переліку нових документів _newDocuments = _newDocuments?.Append(Document.CreateExternal(\$"Постанова МОН"));  // Вивести перелік документів Out(_newDocuments);</pre>	

Результат виконання прикладу Append	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Вхідний документ №:      b740f2b5-70c1-4390-811d-64e17e8a193d Розпорядження з міністерства від      25.04.2022  Архівний документ №:      2a850149-3c93-4fd4-b07c-4a3cb1f6a51f Витрати на конференції від      25.04.2022  Внутрішній документ №:      e52e6bfa-e43c-4ec2-b578-1df4fae2fa44 Наказ декана від      04.04.2022  Архівний документ №:      372edb05-09ab-45fe-a4db-dab510f3ede4 Річні витрати на зарплатню від      25.04.2022  Внутрішній документ №:      b113028b-a77f-48da-b5dc-1cf49cc0aff5 Наказ ректора від      14.04.2022  Архівний документ №:      63c744b4-8c54-49c5-bc89-eb90c08c748d Річні витрати на господарську частину від      24.04.2022  Вхідний документ №:      dc08114b-29f8-4456-80f0-a9e58994899c Постанова МОН від      11.04.2022 </pre>
Prepend [27]	Додати елемент на початок колекції      IEnumerable<CurrentType>
Приклад виконання Prepend	<pre> // Додати новий документ, та зберегти // його на початок переліку нових документів _newDocuments = _newDocuments?.Prepend(Document.CreateExternal(\$"Постанова МОН"));  // Вивести перелік документів Out(_newDocuments); </pre>
Результат виконання прикладу Prepend	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Вхідний документ №:      444a526c-1057-46cd-a06e-0bede8b0e964 Постанова МОН від      03.05.2022  Вхідний документ №:      4ac15078-a873-4995-93b8-173f245875a0 Розпорядження з міністерства від      30.04.2022  Архівний документ №:      381cc31b-14d5-42f0-82cc-cabf29b8443b Витрати на конференції від      01.05.2022  Внутрішній документ №:      cd691b7d-ce22-45d2-b5c9-78b340da458f Наказ декана від      04.04.2022  Архівний документ №:      aa26d9cf-f890-4ea1-8004-0cba499b276e Річні витрати на зарплатню від      24.04.2022  Внутрішній документ №:      b28532e9-a551-46d2-ac61-629d16e1e442 Наказ ректора від      25.04.2022  Архівний документ №:      2669251d-2ea2-40ec-a0f2-cf02aaf9cdd9 Річні витрати на господарську частину від      07.04.2022 </pre>



OrderBy [28]	Відсортувати колекцію за зростанням (від найменшого до найбільшого) за основним (primary) ключем (властивістю). Якщо ключ сортування є класом, то цей клас повинен реалізовувати інтерфейс <code>IEnumerator&lt;&gt;</code> .	<code>IOrderedEnumerable&lt;CurrentType&gt;</code>
Приклад виконання OrderBy	<pre>// Відсортувати за зростанням // від старих до нових документів var oldToNewer = _newDocuments?.OrderBy(doc =&gt; doc?.CreationDate);  // Вивести перелік документів Out(oldToNewer);</pre>	
Результат виконання прикладу OrderBy	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Архівний документ №:      8b6c8282-8186-4980-9158-2e060f3abc63 Витрати на конференції від  04.04.2022  Внутрішній документ №:    9307a195-52af-434a-b3c9-b760c65302df Наказ декана від          04.04.2022  Архівний документ №:      149f6ea9-d091-413e-88e7-5c8fa160b5ab Річні витрати на зарплатню від  06.04.2022  Внутрішній документ №:    ace4c28a-2411-4268-9b96-24ffb6f1f643 Наказ ректора від         14.04.2022  Вхідний документ №:       abd5c284-4465-40c7-a31f-3cd028d09945 Розпорядження з міністерства від  18.04.2022  Архівний документ №:      d825364a-a6db-4516-8531-c61bc6f5da3a Річні витрати на господарську частину від  21.04.2022</pre>	
OrderByDescending [29]	Відсортувати колекцію за спаданням (від найбільшого до найменшого) за основним (primary) ключем (властивістю). Якщо ключ сортування є класом, то цей клас повинен реалізовувати інтерфейс <code>IEnumerator&lt;&gt;</code> .	<code>IOrderedEnumerable&lt;CurrentType&gt;</code>
Приклад виконання OrderByDescending	<pre>// Відсортувати за зростанням // від нових до старих документів var newerToOld = _newDocuments?.OrderByDescending(doc =&gt; doc?.CreationDate);  // Вивести перелік документів Out(newerToOld);</pre>	

<p>Результат виконання прикладу OrderByDescending</p>	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Внутрішній документ №: 8f44a568-214d-4b3a-be14-091b270f59d7 Наказ ректора від 02.05.2022  Архівний документ №: 71b8a7bc-1eeb-4c0d-be8b-e0d5717812f9 Витрати на конференції від 23.04.2022  Архівний документ №: ad3d5da3-39bf-438d-9aa9-395c5d4d8d9a Річні витрати на господарську частину від 21.04.2022  Архівний документ №: 0ac3cd0d-d90b-44a1-91e8-e510eda16b6e Річні витрати на зарплатню від 10.04.2022  Вхідний документ №: 846f068b-841a-4c62-a745-389da3c1b704 Розпорядження з міністерства від 09.04.2022  Внутрішній документ №: 2d3c7090-d7a3-4633-99db-abb72654b0d5 Наказ декана від 04.04.2022 </pre>	
<p>ThenBy [30]</p>	<p>Сортує відсортовану за зростанням колекцію, по вторинному (secondary) ключу (властивості). Використовується коли відсутній унікальний ідентифікатор, тобто сортування відбувається за декількома ознаками.</p>	<p>IOrderedEnumerable&lt;CurrentType&gt;</p>
<p>Приклад виконання ThenBy</p>	<pre> // Відсортувати за зростанням: // спочатку всі інші документи, // потім за "Наказами", // потім за "Витратами" var orders = _newDocuments?.OrderBy(doc =&gt; doc?.Title?.Contains(\$"витрати",     StringComparison.InvariantCultureIgnoreCase))     ?.ThenBy(doc =&gt; doc?.Title?.Contains(\$"наказ",     StringComparison.InvariantCultureIgnoreCase));  // Вивести перелік документів Out(orders); </pre>	

Результат виконання прикладу ThenBy	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Вхідний документ №:      a29548b9-503b-4b51-97e0-616c02584b9f Розпорядження з міністерства від      04.04.2022  Внутрішній документ №:  610da8f6-8520-47a8-9ac3-ee31ab012de4 Наказ декана від      04.04.2022  Внутрішній документ №:  61244264-2f13-483a-ab60-b5458ba76720 Наказ ректора від      05.04.2022  Архівний документ №:    948d952d-af55-4a54-8c29-ea02ed6400f6 Витрати на конференції від      07.04.2022  Архівний документ №:    7eced3c1-8ab2-45b0-b575-303a439ff38c Річні витрати на зарплатню від      16.04.2022  Архівний документ №:    b53374a3-246c-4fc4-8f38-bacaa129e810 Річні витрати на господарську частину від      22.04.2022 </pre>
ThenByDescending [31]	<div>Сортує відсортовану за спаданням колекцію, по вторинному (secondary) ключу (властивості). Використовується коли відсутній унікальний ідентифікатор, тобто сортування відбувається за декількома ознаками.</div> <div>IOrderedEnumerable&lt;CurrentType&gt;</div>
Приклад виконання ThenByDescending	<pre> // Відсортувати за зростанням // спочатку за "Витратами" // потім за "Наказами", // всі інші документи var orders = _newDocuments?.OrderByDescending(doc =&gt;     doc?.Title?.Contains(\$"витрати",         StringComparison.InvariantCultureIgnoreCase))     ?.ThenByDescending(doc =&gt;         doc?.Title?.Contains(\$"наказ",             StringComparison.InvariantCultureIgnoreCase));  // Вивести перелік документів Out(orders); </pre>

Результат виконання прикладу ThenByDescending	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Архівний документ №:      7784e3f5-7a9b-4808-a44e-ab37f3ce1281 Витрати на конференції від 12.04.2022  Архівний документ №:      dc295bbf-e496-4aa3-a1e5-c65c5a43dafb Річні витрати на зарплатню від 17.04.2022  Архівний документ №:      60277d2c-97a7-4bdc-99b7-c908b1cfc175 Річні витрати на господарську частину від 14.04.2022  Внутрішній документ №:    6bf00062-73f8-4bf9-bec2-ae18aaa4f094 Наказ декана від 04.04.2022  Внутрішній документ №:    9786fbea-db90-4308-a669-35db9321892b Наказ ректора від 05.04.2022  Вхідний документ №:       c0f913a5-a4a2-4b10-80ba-328f8d0477d6 Розпорядження з міністерства від 07.04.2022 </pre>
GroupBy [32]	Групування елементів колекції за ключем (властивістю). <div> IEnumerable&lt;IGrouping&lt;TKey,CurrentType&gt;&gt; </div>
Приклад виконання GroupBy	<pre> // згрупувати документи за їх типом var groupedDocs = _newDocuments?.GroupBy(doc =&gt; doc.Type);  // Пройти по колекції груп документів foreach(var group in groupedDocs) {     Console.WriteLine(_delimiter);     // Доступ до ключа відбувається     // через властивість Key     Console.WriteLine(\$"Тип документа {group?.Key}:");     Console.WriteLine(_delimiter);     // Вивести документи,     // які входять в поточну групу     Out(group); } </pre>

Результат виконання прикладу GroupBy	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 ----- Тип документа Вхідний: ----- Вхідний документ №:      0e8c9421-4e35-4752-bcee-8829744a3e62 Розпорядження з міністерства від      22.04.2022 ----- Тип документа Архівний: ----- Архівний документ №:      4d2837be-5a1c-4c1d-892d-d4965ac6643e Витрати на конференції від      11.04.2022  Архівний документ №:      14ea0fcd-4a05-450a-b155-5c7f94d62a2d Річні витрати на зарплатню від      11.04.2022  Архівний документ №:      ef0e66e2-95b9-4357-a600-329ee4a628d1 Річні витрати на господарську частину від      01.05.2022 ----- Тип документа Внутрішній: ----- Внутрішній документ №:      714f3d8b-58fe-4bc4-bdb9-beb6577f15c8 Наказ декана від      04.04.2022  Внутрішній документ №:      4dc5dd60-edb5-4357-b2f7-31bb52b29855 Наказ ректора від      22.04.2022 </pre>	
ToLookup [33]	<p>Перетворює поточну колекцію в колекцію <code>Lookup&lt;TKey, AnotherType&gt;</code>, яка зберігає елементи у вигляді «ключ : багато значень».</p> <p>Перетворення відбувається за вказаним ключем (властивістю) та функцією формування багатьох значень ключа.</p>	<code>ILookup&lt;TKey, AnotherType &gt;</code>
Приклад виконання ToLookup	<pre> // згрупувати документи за їх типом var groupedDocs = _newDocuments?.ToLookup(doc =&gt; doc?.Type, // Ключ doc =&gt; {     // Формування елементів, які будуть     // зберігатись за дним ключем     return \$"{doc?.Title} від {doc?.CreationDate}"; });  // Пройти по колекції груп документів foreach (var group in groupedDocs) {     Console.WriteLine(_delimiter);     // Доступ до ключа відбувається     // через властивість Key     Console.WriteLine(\$"Тип документа {group?.Key}:");     Console.WriteLine(_delimiter);     // Вивести документи,     // які знаходяться за     // поточним ключем     Out(group); } </pre>	

<p>Результат виконання прикладу ToLookup</p>	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 ----- Тип документа Вхідний: ----- Розпорядження з міністерства від 29.04.2022 17:32:18 ----- Тип документа Архівний: ----- Витрати на конференції від 20.04.2022 17:32:18 Річні витрати на зарплатню від 22.04.2022 17:32:18 Річні витрати на господарську частину від 29.04.2022 17:32:18 ----- Тип документа Внутрішній: ----- Наказ декана від 04.04.2022 17:32:18 Наказ ректора від 19.04.2022 17:32:18 </pre>
<p>GroupJoin [34]</p>	<p>Зпівставлення елементів двох колекцій на основі рівності їх ключів (властивостей), та на основі ключів групується результуючі елементи (їх потрібно сформувані).</p> <p>IEnumerable&lt;AnotherType&gt;</p>
<p>Приклад виконання GroupJoin</p>	<pre> // Максимальний час зберігання документа за його типом var documentStoringDeadline = new[]{     new {         // Архівний документ         Type = ArchivedDocument.ArchivedTypeTitle,         // Зберігається 5 років         Duration = 5, },     new {         // Внутрішній документ         Type = Document.InternalDocumentTypeTitle,         // Зберігається 1 рік         Duration = 1, },     new {         // Вхідний документ         Type = Document.ExternalDocumentTypeTitle,         // Зберігається 2 роки         Duration = 2, }, };  // Зпівставлення типу документа та його часу зберігання var documentsStoringDeadline =     _newDocuments?.GroupJoin(documentStoringDeadline,         // ключі повинні співпадати за типом даних         newDoc =&gt; newDoc?.Type,         // ключі повинні співпадати за типом даних         docType =&gt; docType?.Type,         (newDoc, docTypeDurationCollection) =&gt;         {             // Сформувати новий елемент колекції             return new             {                 Title = newDoc?.Title,                 DocTypeTitle = newDoc?.Type,                 // Тривалість зберігання документа                 StorageDuration = docTypeDurationCollection?.Select(d =&gt; d?.Duration),});}); </pre>



	<pre>// Вивести інформація про час зберігання документа foreach (var doc in documentsStoringDeadline) {     Console.WriteLine(_delimiter);     Console.WriteLine(\$"Назва документа:\t{doc?.Title}");     Console.WriteLine(\$"Тип документа:\t{doc?.DocTypeTitle}");     Console.WriteLine(\$"Час зберігання (в роках):");     Out(doc?.StorageDuration); } }</pre>	
Результат виконання прикладу GroupJoin	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 ----- Назва документа:      Розпорядження з міністерства Тип документа:   Вхідний Час зберігання (в роках): 2 ----- Назва документа:      Витрати на конференції Тип документа:   Архівний Час зберігання (в роках): 5 ----- Назва документа:      Наказ декана Тип документа:   Внутрішній Час зберігання (в роках): 1 ----- Назва документа:      Річні витрати на зарплатню Тип документа:   Архівний Час зберігання (в роках): 5 ----- Назва документа:      Наказ ректора Тип документа:   Внутрішній Час зберігання (в роках): 1 ----- Назва документа:      Річні витрати на господарську частину Тип документа:   Архівний Час зберігання (в роках): 5</pre>	
Reverse [35]	Змінює порядок слідування елементів в колекції на протилежний.	IEnumerable<CurrentType>
Приклад виконання Reverse	<pre>ShowAllDocuments(_newDocuments,     // Змінити порядок елементів     // колекції на протилежний     _newDocuments?.Reverse(),     "До", "Після");</pre>	

Результат виконання прикладу Reverse	<div>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1</div> <table><thead><tr><th>До</th><th>Після</th></tr></thead><tbody><tr><td>Вхідний документ №: 81d00e94-9a7c-4f08-9db3-0803ba512aed Розпорядження з міністерства від 16.04.2022</td><td>Архівний документ №: 5738f1ca-606f-4f1d-ab0b-02d8fe10399f Річні витрати на господарську частину від 19.04.2022</td></tr><tr><td>Архівний документ №: 3913d097-943f-425f-a7ab-16d81939e37b Витрати на конференції від 07.04.2022</td><td>Внутрішній документ №: 5adcb9fd-7d44-4be3-944b-6bfc87820cd Наказ ректора від 07.04.2022</td></tr><tr><td>Внутрішній документ №: 94d9c39a-7bbb-4dda-bea6-6fca2d4ba6d2 Наказ декана від 04.04.2022</td><td>Архівний документ №: 32f30830-f672-439d-b4e0-965822cb33b6 Річні витрати на зарплатню від 27.04.2022</td></tr><tr><td>Архівний документ №: 32f30830-f672-439d-b4e0-965822cb33b6 Річні витрати на зарплатню від 27.04.2022</td><td>Внутрішній документ №: 94d9c39a-7bbb-4dda-bea6-6fca2d4ba6d2 Наказ декана від 04.04.2022</td></tr><tr><td>Внутрішній документ №: 5adcb9fd-7d44-4be3-944b-6bfc87820cd Наказ ректора від 07.04.2022</td><td>Архівний документ №: 3913d097-943f-425f-a7ab-16d81939e37b Витрати на конференції від 07.04.2022</td></tr><tr><td>Архівний документ №: 5738f1ca-606f-4f1d-ab0b-02d8fe10399f Річні витрати на господарську частину від 19.04.2022</td><td>Вхідний документ №: 81d00e94-9a7c-4f08-9db3-0803ba512aed Розпорядження з міністерства від 16.04.2022</td></tr></tbody></table>		До	Після	Вхідний документ №: 81d00e94-9a7c-4f08-9db3-0803ba512aed Розпорядження з міністерства від 16.04.2022	Архівний документ №: 5738f1ca-606f-4f1d-ab0b-02d8fe10399f Річні витрати на господарську частину від 19.04.2022	Архівний документ №: 3913d097-943f-425f-a7ab-16d81939e37b Витрати на конференції від 07.04.2022	Внутрішній документ №: 5adcb9fd-7d44-4be3-944b-6bfc87820cd Наказ ректора від 07.04.2022	Внутрішній документ №: 94d9c39a-7bbb-4dda-bea6-6fca2d4ba6d2 Наказ декана від 04.04.2022	Архівний документ №: 32f30830-f672-439d-b4e0-965822cb33b6 Річні витрати на зарплатню від 27.04.2022	Архівний документ №: 32f30830-f672-439d-b4e0-965822cb33b6 Річні витрати на зарплатню від 27.04.2022	Внутрішній документ №: 94d9c39a-7bbb-4dda-bea6-6fca2d4ba6d2 Наказ декана від 04.04.2022	Внутрішній документ №: 5adcb9fd-7d44-4be3-944b-6bfc87820cd Наказ ректора від 07.04.2022	Архівний документ №: 3913d097-943f-425f-a7ab-16d81939e37b Витрати на конференції від 07.04.2022	Архівний документ №: 5738f1ca-606f-4f1d-ab0b-02d8fe10399f Річні витрати на господарську частину від 19.04.2022	Вхідний документ №: 81d00e94-9a7c-4f08-9db3-0803ba512aed Розпорядження з міністерства від 16.04.2022
До	Після															
Вхідний документ №: 81d00e94-9a7c-4f08-9db3-0803ba512aed Розпорядження з міністерства від 16.04.2022	Архівний документ №: 5738f1ca-606f-4f1d-ab0b-02d8fe10399f Річні витрати на господарську частину від 19.04.2022															
Архівний документ №: 3913d097-943f-425f-a7ab-16d81939e37b Витрати на конференції від 07.04.2022	Внутрішній документ №: 5adcb9fd-7d44-4be3-944b-6bfc87820cd Наказ ректора від 07.04.2022															
Внутрішній документ №: 94d9c39a-7bbb-4dda-bea6-6fca2d4ba6d2 Наказ декана від 04.04.2022	Архівний документ №: 32f30830-f672-439d-b4e0-965822cb33b6 Річні витрати на зарплатню від 27.04.2022															
Архівний документ №: 32f30830-f672-439d-b4e0-965822cb33b6 Річні витрати на зарплатню від 27.04.2022	Внутрішній документ №: 94d9c39a-7bbb-4dda-bea6-6fca2d4ba6d2 Наказ декана від 04.04.2022															
Внутрішній документ №: 5adcb9fd-7d44-4be3-944b-6bfc87820cd Наказ ректора від 07.04.2022	Архівний документ №: 3913d097-943f-425f-a7ab-16d81939e37b Витрати на конференції від 07.04.2022															
Архівний документ №: 5738f1ca-606f-4f1d-ab0b-02d8fe10399f Річні витрати на господарську частину від 19.04.2022	Вхідний документ №: 81d00e94-9a7c-4f08-9db3-0803ba512aed Розпорядження з міністерства від 16.04.2022															
Distinct [36]	Повертає сукупність окремих елементів колекції, тобто отримується список унікальних елементів колекції. Для порівняння використовується	IEnumerable<CurrentType>														
Приклад виконання Distinct	<pre>// Копія переліку нових документів var origin = new IDocument[_newDocuments.Count()];  // Скопіювати елементи _newDocuments?.Select(doc =&gt; doc)?.ToList()?.CopyTo(origin);  var id = _newDocuments.ElementAt(0).Id;  (_newDocuments?.ElementAt(3) as DocumentBase)?.UpdateId(id); (_newDocuments?.ElementAt(4) as DocumentBase)?.UpdateId(id);  ShowAllDocuments(origin,     // Змінити порядок елементів</pre>															

	<pre>// колекції на протилежний _newDocuments?.Distinct(), "До", "Після");</pre>	
Результат виконання прикладу Distinct	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 ----- До   Після ----- Вхідний документ №:   Вхідний документ №: 3eab3dfff-03b9-43e6-b7f6-110f75d2b2b2   3eab3dfff-03b9-43e6-b7f6-110f75d2b2b2 Розпорядження з міністерства   Розпорядження з міністерства від 16.04.2022   від 16.04.2022 ----- Архівний документ №:   Архівний документ №: edfe88f2-ca24-436d-9ee1-b54836ccfbc8   edfe88f2-ca24-436d-9ee1-b54836ccfbc8 Витрати на конференції   Витрати на конференції від 29.04.2022   від 29.04.2022 ----- Внутрішній документ №:   Внутрішній документ №: e513a314-3b3a-4ba5-bd50-3ae27b6b1236   e513a314-3b3a-4ba5-bd50-3ae27b6b1236 Наказ декана   Наказ декана від 04.04.2022   від 04.04.2022 ----- Архівний документ №:   Архівний документ №: 3eab3dfff-03b9-43e6-b7f6-110f75d2b2b2   946c206e-6c9c-4316-9410-ec4a8dad6923 Річні витрати на зарплатню   Річні витрати на господарську частину від 23.04.2022   від 27.04.2022 ----- Внутрішній документ №:   документ №: 3eab3dfff-03b9-43e6-b7f6-110f75d2b2b2    Наказ ректора    від 13.04.2022   від ----- Архівний документ №:   документ №: 946c206e-6c9c-4316-9410-ec4a8dad6923    Річні витрати на господарську частину    від 27.04.2022   від</pre>	
Concat [37]	Поєднання двох колекцій однотипних елементів.	IEnumerable<CurrentType>
Приклад виконання Concat	<pre>var united = _newDocuments?.Concat(_archivedDocuments);  Console.WriteLine("Об'єднані переліки документів:"); Out(united);</pre>	

Результат виконання прикладу Concat	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Об'єднані переліки документів: Вхідний документ №:      2a50238b-ef8f-442d-b662-68688722146c Розпорядження з міністерства від      19.04.2022  Архівний документ №:      eab80dfc-ee71-4979-b92e-b1790c655d54 Витрати на конференції від      09.04.2022  Внутрішній документ №:      794fdb69-3379-4d66-8a60-63415d4eb520 Наказ декана від      04.04.2022  Архівний документ №:      b3db08a0-112e-48c6-94af-c650e86e3818 Річні витрати на зарплатню від      22.04.2022  Внутрішній документ №:      f95e81ca-fe2d-4a7d-a1f5-5e7fdacb8d37 Наказ ректора від      17.04.2022  Архівний документ №:      d16d5458-2d9a-42c9-a80d-6d5cf9ddb08b Річні витрати на господарську частину від      29.04.2022  Вхідний документ №:      ead100c9-4993-44e6-b41f-ffb13b9ba592 Постанова з міністерства від      05.04.2022  Архівний документ №:      16b7462f-4fcd-4117-823e-f75b384beb0e Про дохід від      17.04.2022  Вхідний документ №:      1a42c93b-b13c-40c3-ae62-115583d53129 Наказ з міністерства від      04.04.2022  Архівний документ №:      4b0d7f86-165b-4e15-9bfc-a26deed90cf6 Соціальні виплати від      11.04.2022  Внутрішній документ №:      6f2faa3b-21c1-4306-abc1-39325d9f84a0 Внутрішнє розпорядження від      06.04.2022  Архівний документ №:      f26e4415-d1e2-416d-82fc-34c81fbdcd8c Оплата контракту від      24.04.2022 </pre>	
Range [38]	Згенерувати послідовність чисел з заданим діапазоном	IEnumerable<Int32>
Приклад виконання Range	<pre> // Згенерувати послідовність цілих чисел від 1 до 5 var numbers = Enumerable.Range(1, 5);  Console.WriteLine("Послідовність чисел:"); Out(numbers); </pre>	
Результат виконання прикладу Range	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Послідовність чисел: 1 2 3 4 5 </pre>	

Empty [39]	Створює порожню колекцію вказаного типу даних.	IEnumerable<CurrentType>
Приклад виконання Empty	<pre>// Порожній перелік документів var docs = Enumerable.Empty&lt;IDocument&gt;();  Console.WriteLine(\$"Кількість документів:\t{docs?.Count()}");</pre>	
Результат виконання прикладу Empty	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Кількість документів: 0</pre>	
Repeat [40]	Генерує колекцію з повторюваним елементом	IEnumerable<CurrentType>
Приклад виконання Repeat	<pre>// Створити перелік повторюваних документів var docs = Enumerable.Repeat(Document.CreateInternal(\$"Заява на звільнення"), 3);  Console.WriteLine(\$"Перелік документів:"); Out(docs);</pre>	
Результат виконання прикладу Repeat	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Перелік документів: Внутрішній документ №: 3bfcd5b5-42ea-4453-95e1-1aa272de8ba6 Заява на звільнення від 27.04.2022  Внутрішній документ №: 3bfcd5b5-42ea-4453-95e1-1aa272de8ba6 Заява на звільнення від 27.04.2022  Внутрішній документ №: 3bfcd5b5-42ea-4453-95e1-1aa272de8ba6 Заява на звільнення від 27.04.2022</pre>	
Zip [41]	Застосувати вказану функцію до пар елементів з двох колекцій, тобто елементів з однаковими індексами. Якщо одна колекція має менше елементів ніж інша, то функція буде застосована тільки до меншої кількості елементів.	IEnumerable<CurrentType>
Приклад виконання Zip	<pre>var docs = _newDocuments?.Zip(_archivedDocuments,     (newDoc, archiveDoc) =&gt;     {         // Повернути новіший (за датою створення) документ         if (newDoc?.CreationDate &gt; archiveDoc?.CreationDate)             return newDoc;         else             return archiveDoc;     });  ShowAllDocuments();  Console.WriteLine(\$"Перелік документів:"); Out(docs);</pre>	

Результат  
виконання  
прикладу Zip

```

Оберіть пункт меню:
1.) Виконати програму;
2.) Завершити роботу;
1
-----
Нові документи | Архівні документи
-----
Вхідний документ №: | Вхідний документ №:
52a8790d-6668-4705-8168-83cd224c1fb6 | af22161a-e26f-4126-a898-351dd12033ad
Розпорядження з міністерства | Постанова з міністерства
від 26.04.2022 | від 18.04.2022
-----
Архівний документ №: | Архівний документ №:
0f29ac3d-4b0f-4d56-8f69-005b29a9d15a | 8dcbe960-4437-4b56-80bd-42f6c331498e
Витрати на конференції | Про дохід
від 06.04.2022 | від 06.04.2022
-----
Внутрішній документ №: | Вхідний документ №:
c3bd43d4-fe02-430f-994d-7aaf54969014 | 5f2b7e46-ab6e-480f-a1a1-d8ca8641bd32
Наказ декана | Наказ з міністерства
від 04.04.2022 | від 04.04.2022
-----
Архівний документ №: | Архівний документ №:
0b612940-2b04-4c1f-a470-1f1c39a2f2e0 | 4583e451-c6ca-4506-81b2-d36b6af3b9f9
Річні витрати на зарплатню | Соціальні виплати
від 26.04.2022 | від 26.04.2022
-----
Внутрішній документ №: | Внутрішній документ №:
911e6d66-3bc1-4160-8999-5c8c83cf9a2b | c38ae1c5-5a07-4e41-989d-409449b9a20d
Наказ ректора | Внутрішнє розпорядження
від 10.04.2022 | від 03.05.2022
-----
Архівний документ №: | Архівний документ №:
97860180-1b45-4309-812a-c76f544e5be8 | c1f459b7-31b1-4514-bf37-0bed7ace31b1
Річні витрати на господарську частину | Оплата контракту
від 26.04.2022 | від 26.04.2022
-----
Перелік документів:
Вхідний документ №: 52a8790d-6668-4705-8168-83cd224c1fb6
Розпорядження з міністерства від 26.04.2022

Архівний документ №: 0f29ac3d-4b0f-4d56-8f69-005b29a9d15a
Витрати на конференції від 06.04.2022

Внутрішній документ №: c3bd43d4-fe02-430f-994d-7aaf54969014
Наказ декана від 04.04.2022

Архівний документ №: 0b612940-2b04-4c1f-a470-1f1c39a2f2e0
Річні витрати на зарплатню від 26.04.2022

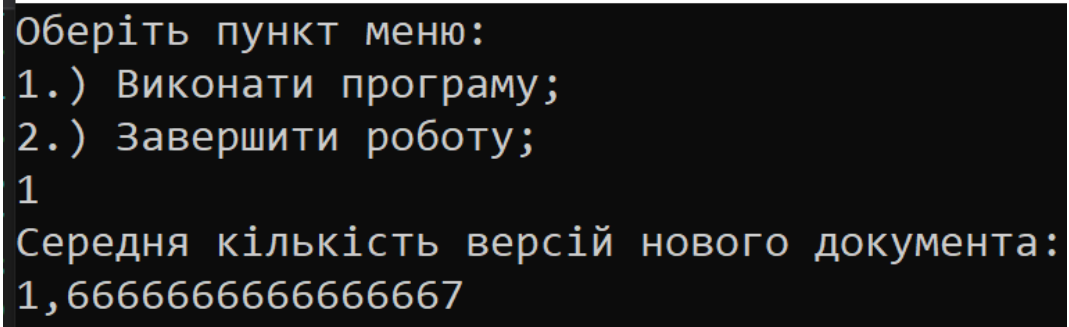
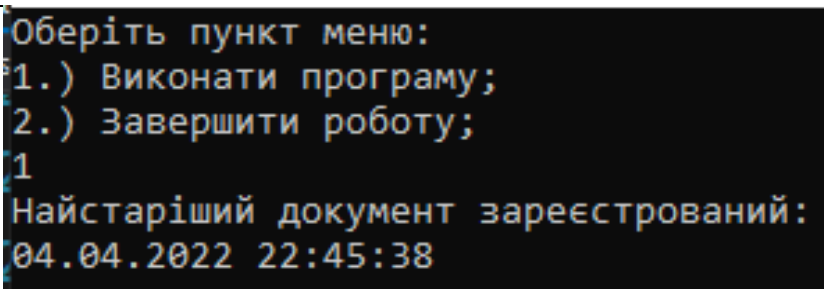
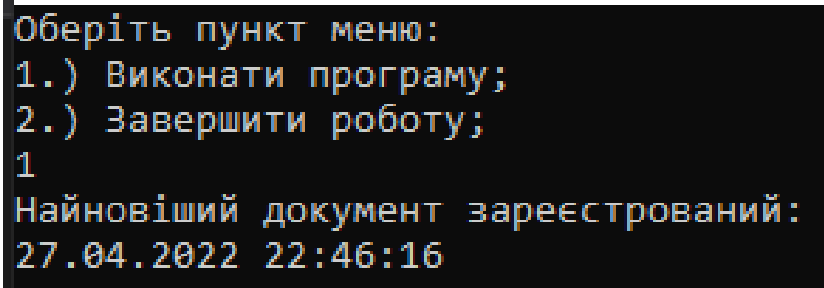
Внутрішній документ №: c38ae1c5-5a07-4e41-989d-409449b9a20d
Внутрішнє розпорядження від 03.05.2022

Архівний документ №: 97860180-1b45-4309-812a-c76f544e5be8
Річні витрати на господарську частину від 26.04.2022

```



АГРЕГАЦІЇ		
Aggregate [42]	Застосування заданої функції зі спеціальним початковим значенням, до кожного елемента колекції.	AnotherType
Приклад виконання Aggregate	<pre>var archivedDocsCount = _newDocuments?.Aggregate(0, // Початкове значення     (count, doc) =&gt;     {         // Підрахувати кіл-сть архівних         // документів         if (doc?.Type ==             ArchivedDocument.ArchivedTypeTitle)             count++;         return count; },     // повернути результат підрахунку     count =&gt; count);  Console.WriteLine(\$"Кількість архівних документів:\t{archivedDocsCount}");</pre>	
Результат виконання прикладу Aggregate		
Count [43]	Підрахувати кількість елементів в колекції	Int32
Приклад виконання Count	<pre>var newDocsCount = _newDocuments?.Count();  Console.WriteLine(\$"Кількість нових документів:\t{newDocsCount}");</pre>	
Результат виконання прикладу Count		
Sum [44]	Сума всіх елементів колекції чисел або сума певної властивості елементів колекції.	Int, Single, Double, Decimal
Приклад виконання Sum	<pre>var docVersionsCount = _newDocuments?.Sum(doc =&gt; doc?.Versions?.Count());  Console.WriteLine(\$"Загальна кількість версій нових документів:"); Console.WriteLine(\$"{docVersionsCount}");</pre>	
Результат виконання прикладу Sum		

Average [45]	Середнє арифметичне всіх елементів колекції чисел або середнє арифметичне певної властивості елементів колекції.	Int, Single, Double, Decimal
Приклад виконання Average	<pre>var docVersionsAvg = _newDocuments?.Average(doc =&gt; doc?.Versions?.Count()); Console.WriteLine(\$"Середня кількість версій нового документа:"); Console.WriteLine(\$"{docVersionsAvg}");</pre>	
Результат виконання прикладу Average	 <pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Середня кількість версій нового документа: 1,6666666666666667</pre>	
Min [46]	Мінімальний елемент в колекції.	CurrentType
Приклад виконання Min	<pre>var oldestDoc = _newDocuments?.Min(doc =&gt; doc?.CreationDate); Console.WriteLine(\$"Найстаріший документ:"); Console.WriteLine(\$"{oldestDoc}");</pre>	
Результат виконання прикладу Min	 <pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Найстаріший документ зареєстрований: 04.04.2022 22:45:38</pre>	
Max [47]	Максимальний елемент колекції.	CurrentType
Приклад виконання Max	<pre>var newestDoc = _newDocuments?.Max(doc =&gt; doc?.CreationDate); Console.WriteLine(\$"Найновіший документ зареєстрований:"); Console.WriteLine(\$"{newestDoc}");</pre>	
Результат виконання прикладу Max	 <pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Найновіший документ зареєстрований: 27.04.2022 22:46:16</pre>	
All [48]	Чи всі елементи колекції відповідають умові?	bool
Приклад виконання All	<pre>var isAllInternal = _newDocuments.All(doc =&gt; doc?.Type == Document.InternalDocumentTypeTitle); Console.WriteLine(\$"Чи всі нові документи внутрішні:"); Console.WriteLine(\$"{((isAllInternal)? "Так": "Ні")}");</pre>	

Результат виконання прикладу All	<div>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Чи всі нові документи внутрішні: Ні</div>	
Any [49]	Чи хоча б один елемент колекції відповідає умові?	bool
Приклад виконання Any	<pre>var isAnyArchive = _newDocuments.Any(doc =&gt; doc?.Type == ArchivedDocument.ArchivedTypeTitle);  Console.WriteLine(\$"Чи наявні архівні документа для опрацювання:"); Console.WriteLine(\$"{{{(isAnyArchive) ? "Так": "Ні"}}}");</pre>	
Результат виконання прикладу Any	<div>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Чи наявні архівні документа для опрацювання: Так</div>	
Contains [50]	Чи містить колекція вказаний елемент?	bool
Приклад виконання Contains	<pre>var isNewDocsToday = _newDocuments.Contains(new Document(){     CreationDate = DateTime.Now, })? "Так" : "Ні";  Console.WriteLine(\$"Чи сьогодні надходили нові документи: {isNewDocsToday}");</pre>	
Результат виконання прикладу Contains	<div>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Чи сьогодні надійшли нові документи: Ні</div>	
УПРАВЛІННЯ ЗАПИТАМИ		
AsParallel [51]	Виконання LINQ-запиту в паралельному режимі	ParallelQuery
Приклад виконання AsParallel	<pre>_newDocuments?.AsParallel()?.ForAll(doc =&gt; {     // Змінити дату створення документа     // на сьогоднішню з використанням     // паралелізму     doc.CreationDate = DateTime.Now; });</pre>	

	<pre>Console.WriteLine(\$"Документи, які надійши сьогодні:"); Out(_newDocuments);</pre>	
Результат виконання прикладу AsParallel	<pre>Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Документи, які надійши сьогодні: Вхідний документ №:      db602d08-4c12-4826-a191-6174922218b4 Розпорядження з міністерства від      04.04.2022  Архівний документ №:      a5bb2150-581e-4962-802c-1416a2027ffc Витрати на конференції від      04.04.2022  Внутрішній документ №:      89bb678f-ac56-41ab-b201-55c6ec02852e Наказ декана від      04.04.2022  Архівний документ №:      5a766393-dc22-4c55-ac37-9c24a6054fa3 Річні витрати на зарплатню від      04.04.2022  Внутрішній документ №:      a1058287-99a5-4cb4-9982-28b79ba29bec Наказ ректора від      04.04.2022  Архівний документ №:      cdedac34-4526-4594-85ff-2ebc3189dc33 Річні витрати на господарську частину від      04.04.2022</pre>	
AsQueryable [52]	<p>Перетворює колекцію IEnumerable&lt;CurrentType&gt; в колекцію IQueryable&lt;CurrentType&gt;.</p> <p>Інтерфейс IQueryable&lt;&gt; потрібен коли необхідно забезпечити підтримку певної мови запитів для роботи з джерелом даних.</p>	IQueryable<CurrentType>
Приклад виконання AsQueryable	<pre>// Інтерфейс IQueryable наслідується від // IEnumerable, що забезпечує використання LINQ/PLINQ var docs = _newDocuments?.AsQueryable()?.Select(doc =&gt; doc);  Console.WriteLine(\$"Документи, отримані з непідтримуваного джерела даних:"); Out(docs);</pre>	

Результат виконання прикладу AsQueryable	<pre> Оберіть пункт меню: 1.) Виконати програму; 2.) Завершити роботу; 1 Документи, отримані з непідтримуваного джерела даних: Вхідний документ №:      ea633a16-3726-46cc-90da-4c5ac8143020 Розпорядження з міністерства від      18.04.2022  Архівний документ №:      612c70c8-4109-4ef2-8a88-91814c11a39a Витрати на конференції від      21.04.2022  Внутрішній документ №:    4169d851-176f-4ac8-8204-84450ce2b83c Наказ декана від      04.04.2022  Архівний документ №:      983087a3-2ff8-4b89-91b1-a0c8011a1fc9 Річні витрати на зарплатню від      20.04.2022  Внутрішній документ №:    ce778f7b-10b0-4af6-b14f-416a6a4d94df Наказ ректора від      08.04.2022  Архівний документ №:      a1f5ce2c-3fe3-4bc1-9f54-ae8c336dda17 Річні витрати на господарську частину від      06.04.2022 </pre>	
--	---	--

Таблиця 2 — Опис та приклади використання методів розширення LINQ/PLINQ

**Теми для завдання**

№	Назва
1	Система бухгалтерського обліку
2	Система бібліографічного обліку
3	Система авторизації
4	Текстовий редактор
5	Графічний редактор
6	Метеорологічна система
7	Органайзер
8	Система автоматизації документообігу
9	Система збору інформації про комп'ютер
10	Система відслідковування стану локальної мережі
11	Система обліку товарів на складі
12	Система контролю доступу користувачів
13	Криптографічна система
14	Система керування змістом (CMS)
15	Навчальна система
16	Система тестування програмного забезпечення
17	Математичний калькулятор
18	Система управління вікнами
19	Система управління IoT-пристроями
20	REST-сервіс для взаємодії з WEB-сервісами
21	Синтаксичний аналізатор
22	Панель керування програмним забезпеченням
23	Генератор документів
24	Встановлювальник (installer) програмного забезпечення
25	Система контролю версій
26	Веб-сервер
27	Гра «Шахи»
28	Гра «Монополія»
29	Гра «Тетріс»
30	Гра «П'ять в ряд»
31	Система управління базою даних
32	Табличний редактор
33	Редактор формул
34	Система проектування логічних схем
35	Система диспетчеризації потягів
36	Система 3D-моделей
37	Система проведення
38	Музикальний програвач
39	Програма обміну повідомленнями (messenger)
40	Редактор форм
41	Пошуковий бот (web-crawler)
42	Програма для статистичного аналізу
43	Програма прогнозування результатів матчів
44	Система управління турнірами
45	Автоматизація торгівлі на біржі
46	Механізм блокчейну

**Загальні завдання**

1. Ознайомитися з теоретичною частиною;
2. Перейти в локальний репозиторій з лабораторними роботами;
3. Створити окрему гілку (`git checkout -b`) в системі контролю версій Git для поточної лабораторної роботи;
4. Створити новий консольний проект на мові програмування C# в локальному репозиторії;
5. Визначити та створити необхідні колекції, які будуть використовуватись в обраній темі;
6. Використати мінімум 3 методи розширення вибірки даних (на власний розсуд);
7. Використати мінімум 3 методи розширення зміни порядку даних (на власний розсуд);
8. Використати мінімум 2 методи розширення вибірки даних (на власний розсуд);
9. Використати мінімум 1 метод розширення управління запитами (на власний розсуд);
10. Продемонструвати роботу програми;
11. Результат роботи програми зберегти у вигляді скріншоту (png або jpeg) в репозиторії з проектом;
12. Індексувати проект (`git add`);
13. Зафіксувати зміни (`git commit`);
14. Надіслати зміни у віддалений репозиторій (`git push`);
15. Створити та виконати запит на зміни (pull request);
16. Надіслати посилання на поточну лабораторну роботу у віддаленому репозиторії в GitHub.

**Контрольні запитання**

1. Основні операції опрацювання даних.
2. Що таке мова запитів? Які існують мови запитів?
3. Для яких джерел даних існують розширення LINQ/PLINQ?
4. Різниця між LINQ to SQL, LINQ to DataSet, LINQ to Entities.
5. Різниця між використанням методів розширення та синтаксису запитів LINQ/PLINQ.
6. Призначення ключового слова `let` під час використання синтаксису запитів LINQ/PLINQ?
7. Призначення методів розширення вибірки. Приклади таких методів.
8. Призначення методів розширення зміни порядку. Приклади таких методів.
9. Призначення методів агрегації. Приклади таких методів.
10. Призначення методів управління запитами. Приклади таких методів.

**Інформаційні джерела**

1. Троелсен Э., Джепикс Ф. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. : Пер. с англ. — СПб.: ООО «Диалектика», 2018. — 1328 с. : ил.
2. Introduction to LINQ Queries (C#) — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries> Date of access: 01.02.2022.
3. Query keywords (C# Reference) — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/query-keywords> Date of access: 01.02.2022.
4. Enumerable.Select Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.select?view=net-6.0> Date of access: 01.02.2022.
5. Enumerable.SelectMany Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.selectmany?view=net-6.0> Date of access: 01.02.2022.
6. Enumerable.Where Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.where?view=net-6.0> Date of access: 01.02.2022.
7. Enumerable.OfTResult<TResult>(IEnumerable) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.oftype?view=net-6.0> Date of access: 01.02.2022.
8. Сулима Д.О. Система електронного документообігу факультету — 2017. — Mode of access: <https://www.e-journals.npu.edu.ua/index.php/ikt/article/view/120> Date of access: 01.02.2022.



9. Enumerable.Cast<TResult>(IEnumerable) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.cast?view=net-6.0> Date of access: 01.02.2022.
10. Enumerable.Join Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.join?view=net-6.0> Date of access: 01.02.2022.
11. Enumerable.Except Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.except?view=net-6.0> Date of access: 01.02.2022.
12. Enumerable.Union Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.union?view=net-6.0> Date of access: 01.02.2022.
13. Enumerable.Intersect Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.intersect?view=net-6.0> Date of access: 01.02.2022.
14. Enumerable.Take Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.take?view=net-6.0> Date of access: 01.02.2022.
15. Enumerable.TakeWhile Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.takewhile?view=net-6.0> Date of access: 01.02.2022.
16. Enumerable.Skip<TSource>(IEnumerable<TSource>, Int32) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.skip?view=net-6.0> Date of access: 01.02.2022.
17. Enumerable.SkipWhile Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.skipwhile?view=net-6.0> Date of access: 01.02.2022.
18. Enumerable.First Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.first?view=net-6.0> Date of access: 01.02.2022.
19. Enumerable.FirstOrDefault Метод — 2022. — Mode of access: <https://docs.microsoft.com/ru-ru/dotnet/api/system.linq.enumerable.firstordefault?view=net-6.0> Date of access: 01.02.2022.
20. Enumerable.Single Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.single?view=net-6.0> Date of access: 01.02.2022.
21. Enumerable.SingleOrDefault Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.singleordefault?view=net-6.0> Date of access: 01.02.2022.
22. Enumerable.ElementAt Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.elementat?view=net-6.0> Date of access: 01.02.2022.
23. Enumerable.ElementAtOrDefault Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.elementatordefault?view=net-6.0> Date of access: 01.02.2022.
24. Enumerable.Last Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.last?view=net-6.0> Date of access: 01.02.2022.
25. Enumerable.LastOrDefault Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.lastordefault?view=net-6.0> Date of access: 01.02.2022.
26. Enumerable.Append<TSource>(IEnumerable<TSource>, TSource) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.append?view=net-6.0> Date of access: 01.02.2022.
27. Enumerable.Prepend<TSource>(IEnumerable<TSource>, TSource) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.prepend?view=net-6.0> Date of access: 01.02.2022.
28. Enumerable.OrderBy Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.orderby?view=net-6.0> Date of access: 01.02.2022.
29. Enumerable.OrderByDescending Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.orderbydescending?view=net-6.0> Date of access: 01.02.2022.
30. Enumerable.ThenBy Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.thenby?view=net-6.0> Date of access: 01.02.2022.

31. Enumerable.ThenByDescending Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.thenbydescending?view=net-6.0> Date of access: 01.02.2022.
32. Enumerable.GroupBy Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.groupby?view=net-6.0> Date of access: 01.02.2022.
33. Enumerable.ToLookup Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.tolookup?view=net-6.0> Date of access: 01.02.2022.
34. Enumerable.GroupJoin Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.groupjoin?view=net-6.0> Date of access: 01.02.2022.
35. Enumerable.Reverse<TSource>(IEnumerable<TSource>) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.reverse?view=net-6.0> Date of access: 01.02.2022.
36. Enumerable.Distinct Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.distinct?view=net-6.0> Date of access: 01.02.2022.
37. Enumerable.Concat<TSource>(IEnumerable<TSource>, IEnumerable<TSource>) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.concat?view=net-6.0> Date of access: 01.02.2022.
38. Enumerable.Range(Int32, Int32) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.range?view=net-6.0> Date of access: 01.02.2022.
39. Enumerable.Empty<TResult> Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.empty?view=net-6.0> Date of access: 01.02.2022.
40. Enumerable.Repeat<TResult>(TResult, Int32) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.repeat?view=net-6.0> Date of access: 01.02.2022.
41. Enumerable.Zip Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.zip?view=net-6.0> Date of access: 01.02.2022.
42. Enumerable.Aggregate Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.aggregate?view=net-6.0> Date of access: 01.02.2022.
43. Enumerable.Count Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.count?view=net-6.0> Date of access: 01.02.2022.
44. Enumerable.Sum Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.sum?view=net-6.0> Date of access: 01.02.2022.
45. Enumerable.Average Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.average?view=net-6.0> Date of access: 01.02.2022.
46. Enumerable.Min Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.min?view=net-6.0> Date of access: 01.02.2022.
47. Enumerable.Max Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.max?view=net-6.0> Date of access: 01.02.2022.
48. Enumerable.All<TSource>(IEnumerable<TSource>, Func<TSource, Boolean>) Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.all?view=net-6.0> Date of access: 01.02.2022.
49. Enumerable.Any Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.any?view=net-6.0> Date of access: 01.02.2022.
50. Enumerable.Contains Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.contains?view=net-6.0> Date of access: 01.02.2022.
51. ParallelEnumerable.AsParallel Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.parallelenumerable.asparallel?view=net-6.0> Date of access: 01.02.2022.
52. Queryable.AsQueryable Method — 2022. — Mode of access: <https://docs.microsoft.com/en-us/dotnet/api/system.linq.queryable.asqueryable?view=net-6.0> Date of access: 01.02.2022.

**Додаток 1. Код головного меню програми — Program.cs**

```
using System;

// Перевірка на відповідність CLS
[assembly: CLSCompliant(true)]
namespace EDMSConsole
{
    class Program
    {
        /// <summary>
        /// Пункти меню
        /// типу значень (ValueType)
        /// </summary>
        enum MenuOptions : byte
        {
            // Виконати програму
            Execute = 1,
            // Вийти з програми
            Exit,
        }

        static void Menu()
        {
            Console.WriteLine($"Оберіть пункт меню:");
            Console.WriteLine($"1.) Виконати програму;");
            Console.WriteLine($"2.) Завершити роботу;");

            Byte menuOption;

            while (!Byte.TryParse(Console.ReadLine(), out menuOption))
            {
                Console.WriteLine($"Введено неправильні дані, перевірте та спробуйте знову");
            }

            MenuOptions selectedOption = (MenuOptions)menuOption;

            switch (selectedOption)
            {
                case MenuOptions.Execute:
                    ExecuteProgram.Run();
                    Console.ReadLine();
                    break;
                case MenuOptions.Exit:
                    Environment.Exit(0);
                    break;
                default:
                    break;
            }

            Console.ReadLine();
        }

        static void Main(string[] args)
        {
            // Підтримка кириличних символів
            Console.OutputEncoding = System.Text.Encoding.Unicode;
            Console.InputEncoding = System.Text.Encoding.Unicode;

            while (true)
            {
                Menu();
            }
        }
    }
}
```

**Додаток 2. Модель документа — IDocument.cs**

```
using System;
using System.Collections.Generic;

namespace EDMS_DAL.Models.Documents
{
    public interface IDocument
    {
        /// <summary>
        /// Ідентифікаційний номер документа
        /// </summary>
        public Guid Id { get; }

        /// <summary>
        /// Назва документа
        /// </summary>
        public string Title { get; set; }

        /// <summary>
        /// Дата створення документа
        /// </summary>
        public DateTime CreationDate { get; set; }

        /// <summary>
        /// Тип документа
        /// </summary>
        public string Type { get; }

        /// <summary>
        /// Версії документа для відслідковування змін
        /// </summary>
        IEnumerable<string> Versions { get; set; }
    }

    public abstract class DocumentBase : IDocument
    {
        public Guid Id { get; protected set; }
        public string Title { get; set; }
        public DateTime CreationDate { get; set; }
        public string Type { get; protected set; }
        public IEnumerable<string> Versions { get; set; }

        protected DocumentBase(string title,
                                string type)
        {
            Title = title ?? String.Empty;
            Id = Guid.NewGuid();
            CreationDate = GetRandomDate();
            Type = type;
            Versions = GetVersions();
        }

        /// <summary>
        /// Згенерувати випадковий перелік
        /// версій документа
        /// </summary>
        /// <returns></returns>
        private IEnumerable<string> GetVersions()
        {
            var rnd = new Random();
            var versions = new List<string>();

            for (short version = 0; version < rnd.Next(1, 3); version++)
            {
                versions.Add($"v.{version + 1}.0");
            }
        }
    }
}
```

```
    }

    return versions;
}

/// <summary>
/// Оновити ідентифікатор документа
/// </summary>
/// <param name="newId"></param>
public void UpdateId(Guid newId)
{
    this.Id = newId;
}

/// <summary>
/// Згенерувати випадкову дату
/// </summary>
/// <returns></returns>
private DateTime GetRandomDate()
{
    var rnd = new Random();
    // Отримати сьогоднішню дату
    var today = DateTime.Now;
    // Додати випадкову кількість днів,
    // до сьогоднішньої дати
    return today.AddDays(rnd.Next(0, DateTime.DaysInMonth(today.Year,
                                                            today.Month)));
}

protected abstract IDocument Convert(IDocument document);

/// <summary>
/// Порівняння документів відбувається
/// за їх ідентифікаторами
/// </summary>
/// <param name="obj"></param>
/// <returns>
/// Якщо ідентифікатори обох документів
/// однакові, то вони однакові,
/// в усіх інших випадках - різні
/// </returns>
public override bool Equals(object obj)
{
    return this?.Id == (obj as IDocument)?.Id;
}

/// <summary>
/// Отримати хеш ідентифікатора документа
/// </summary>
/// <returns></returns>
public override int GetHashCode()
{
    return this.Id.GetHashCode();
}

/// <summary>
/// Виведення метаданих документа
/// </summary>
/// <returns></returns>
public override string ToString()
{
    return $"{Type} документ №:\t{Id}" +
           $"{Environment.NewLine}{Title}\t" +
           $"Від\t{CreationDate.ToString("dd.MM.yyyy")}" +
           $"{Environment.NewLine}";
}
}
```

```

/// <summary>
/// Архівний документ
/// </summary>
public class ArchivedDocument : DocumentBase
{
    public const string ArchivedTypeTitle = "Архівний";

    private ArchivedDocument(string title,
                              string type)
        : base(title,
               type)
    { }

    /// <summary>
    /// Фабричний метод для створення
    /// архівного документа
    /// </summary>
    /// <returns> Архівний документ
    /// </returns>
    public static IDocument ArchiveDocument(string Title = default)
    {
        return new ArchivedDocument(Title, ArchivedTypeTitle);
    }

    protected override IDocument Convert(IDocument document)
    {
        var newArchivedDocument = new ArchivedDocument(document?.Title,
                                                         ArchivedTypeTitle);

        newArchivedDocument.Id = document.Id;
        newArchivedDocument.CreationDate = document.CreationDate;
        newArchivedDocument.Versions = document.Versions;

        return newArchivedDocument;
    }

    /// <summary>
    /// Зробити зі звичайного документа архівний
    /// </summary>
    /// <param name="doc"></param>
    public static explicit operator ArchivedDocument(Document doc)
    {
        if (doc != null)
        {
            var archived = new ArchivedDocument(doc?.Title, ArchivedTypeTitle);

            return archived?.Convert(doc) as ArchivedDocument;
        }
        else return null;
    }
}

/// <summary>
/// Документ
/// </summary>
public class Document : DocumentBase
{
    public const string InternalDocumentTypeTitle = "Внутрішній";
    public const string ExternalDocumentTypeTitle = "Вхідний";

    private Document(string title,
                      string type)
        : base(title,
               type)
    { }
}

```

```
/// <summary>
/// Фабричний метод для створення
/// внутрішнього документа
/// </summary>
/// <returns> Внутрішній документ
/// </returns>
public static IDocument CreateInternal(string Title = default)
{
    return new Document(Title, InternalDocumentTypeTitle);
}

/// <summary>
/// Фабричний метод для створення
/// вхідного документа
/// </summary>
/// <returns> Вхідний документ
/// </returns>
public static IDocument CreateExternal(string Title = default)
{
    return new Document(Title, ExternalDocumentTypeTitle);
}

protected override IDocument Convert(IDocument document)
{
    var newDocument = new Document(document?.Title,
                                    InternalDocumentTypeTitle);

    newDocument.Id = document.Id;
    newDocument.CreationDate = document.CreationDate;
    newDocument.Versions = document.Versions;

    return newDocument;
}

/// <summary>
/// Зробити з архівного документа внутрішній
/// </summary>
/// <param name="archived"></param>
public static explicit operator Document(ArchivedDocument archived)
{
    var doc = new Document(archived?.Title, InternalDocumentTypeTitle);

    return doc?.Convert(archived) as Document;
}
}
}
```



**Додаток 3. «Дані-заглушки» для програми — *IDocumentDataService.cs***

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using EDMS_DAL.Models.Documents;

namespace EDMS_DAL.DataServices.DocumentsDataService
{
    /// <summary>
    /// Управління документами
    /// Приклад, реалізації шаблону
    /// Repository
    /// </summary>
    public interface IDocumentDataService
    {
        public ICollection<IDocument> GetAllArchiveDocuments();
        public ICollection<IDocument> GetAllRecentDocuments();
    }
    /// <summary>
    /// Отримати перелік всіх документів
    /// </summary>
    /// <returns></returns>
    public ICollection<IDocument> GetAllArchiveDocuments()
    {
        return new List<IDocument>() {
            Document.CreateExternal("Постанова з міністерства"),
            ArchivedDocument.ArchiveDocument("Про дохід"),
            Document.CreateExternal("Наказ з міністерства"),
            ArchivedDocument.ArchiveDocument("Соціальні виплати"),
            Document.CreateInternal("Внутрішнє розпорядження"),
            ArchivedDocument.ArchiveDocument("Оплата контракту"),
        };
    }
    /// <summary>
    /// Отримати перелік всіх звітів
    /// </summary>
    /// <returns></returns>
    public ICollection<IDocument> GetAllRecentDocuments()
    {
        return new List<IDocument>() {
            Document.CreateExternal($"Розпорядження з міністерства"),
            ArchivedDocument.ArchiveDocument($"Витрати на конференції"),
            Document.CreateInternal($"Наказ декана"),
            ArchivedDocument.ArchiveDocument($"Річні витрати на зарплатню"),
            Document.CreateInternal($"Наказ ректора"),
            ArchivedDocument.ArchiveDocument($"Річні витрати на господарську частину"),
        };
    }
}
}
```

**Додаток 4. ExecuteProgram.cs**

```

using System;
using System.Collections.Generic;
using System.Linq; // LINQ to Objects

using EDMS_DAL.DataServices.DocumentsDataService;
using EDMS_DAL.Models.Documents;
using EDMS_DAL.Resources.Mocks.DocumentsMock;

namespace EDMSConsole
{
    public static class ExecuteProgram
    {
        /// <summary>
        /// Колекція архівних документів
        /// </summary>
        private static IEnumerable<IDocument> _archivedDocuments;

        /// <summary>
        /// Колекція нових документів
        /// </summary>
        private static IEnumerable<IDocument> _newDocuments;

        /// <summary>
        /// Доступ до тестових даних
        /// </summary>
        private static IDocumentDataService _documentsRepository;

        private static string _delimiter;

        static ExecuteProgram()
        {
            _documentsRepository = new DocumentDataServiceMock();
            _archivedDocuments = _documentsRepository?.GetAllArchiveDocuments();
            _newDocuments = _documentsRepository?.GetAllRecentDocuments();

            short index = 2; // Індекс документа в колекції з однаковими датами створення
            // Задати однакові дати створення документів
            _archivedDocuments.ElementAt(index).CreationDate = DateTime.Now;
            _newDocuments.ElementAt(index).CreationDate = DateTime.Now;

            for (short i = 0; i < Console.BufferWidth; i++)
                _delimiter += "-";
        }

        private static void Out<T>(IEnumerable<T> sequence)
        {
            if (sequence != null)
            {
                foreach (var elem in sequence)
                {
                    Console.WriteLine($"{elem}");
                }
            }
        }

        /// <summary>
        /// Додати необхідну кількість пробілів
        /// </summary>
        /// <param name="text"></param>
        /// <param name="centerPosition"></param>
        private static void ExtendTitleToTableCenter(string text,
                                                    int centerPosition)
        {
            for (int i = text.Length; i < centerPosition; i++)
            {

```

```

        Console.WriteLine($" ");
    }
}

/// <summary>
/// Вивести поточну інформацію про колекції документів
/// </summary>
public static void ShowAllDocuments(IEnumerable<IDocument> newDocs = null,
                                   IEnumerable<IDocument> archiveDocs = null,
                                   string leftHeaderText = "Нові документи",
                                   string rightHeaderText = "Архівні документи")
{
    if (newDocs == null)
        newDocs = _newDocuments;
    if (archiveDocs == null)
        archiveDocs = _archivedDocuments;

    // Центр таблиці
    var center = 40;

    var docs = new List<dynamic>();

    var enum1 = newDocs?.GetEnumerator();
    var enum2 = archiveDocs?.GetEnumerator();

    while(enum1.MoveNext() | enum2.MoveNext())
    {
        docs.Add(new
        {
            NewDoc = enum1?.Current,
            ArchiveDoc = enum2?.Current,
        });
    }

    Console.WriteLine(_delimiter);
    Console.WriteLine($"{leftHeaderText}");
    ExtendTitleToTableCenter(leftHeaderText, center);

    Console.WriteLine($"|\\t\\t{rightHeaderText}");
    Console.WriteLine(Environment.NewLine);
    Console.WriteLine(_delimiter);

    foreach (var doc in docs)
    {
        Console.WriteLine($"{doc?.NewDoc?.Type} документ №:\\t\\t\\t|");
        Console.WriteLine($"{\\t\\t{doc?.ArchiveDoc?.Type} документ №:");
        Console.WriteLine(Environment.NewLine);
        Console.WriteLine($"{doc?.NewDoc?.Id}\\t|\\t{doc?.ArchiveDoc?.Id}");

        Console.WriteLine($"{doc?.NewDoc?.Title}");
        ExtendTitleToTableCenter(doc.NewDoc.Title, center);
        Console.WriteLine($"{\\t\\t{doc?.ArchiveDoc?.Title}");

        Console.WriteLine(Environment.NewLine);
        Console.WriteLine($"Bid\\t{doc?.NewDoc?.CreationDate.ToString("dd.MM.yyyy")}\\t\\t\\t|");
        Console.WriteLine($"\\tbid\\t{doc?.ArchiveDoc?.CreationDate.ToString("dd.MM.yyyy")}");
        Console.WriteLine(Environment.NewLine);

        Console.WriteLine(_delimiter);
    }
}

public static void Run()
{
    // Код прикладів необхідно розміщувати тут
}
}

```