

Лабораторна робота №5

Тема: Generic-колекції.

Мета роботи: дослідити можливості інтегрованого середовища розробки Visual Studio і отримати практичні навички зі створення додатків, що використовують generic-колекції на прикладі мови програмування C#. Зміст роботи

Завдання 1. Створіть додаток для обліку співробітників, який має зберігати наступну інформацію:

- П.І.Б.;
- посада;
- заробітна плата;
- корпоративний email.

Для зберігання даних співробітників використовуйте клас List. Додаток має надавати таку функціональність:

- додавати співробітників;
- видаляти співробітників;
- змінювати інформацію про співробітників;
- пошук співробітників за параметрами;
- сортування співробітників за параметрами.

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

```
namespace Task1
```

```
{  
    public class Employee  
    {  
        public string? FullName { get; set; }  
        public string? Position { get; set; }  
        public decimal Salary { get; set; }  
        public string? Email { get; set; }  
    }  
}
```

```
public class EmployeeManagement<T> : IList<T>, IReadOnlyList<T>
```

```
{  
    private readonly List<T> employees = new();
```

```
    public T this[int index]  
    {  
        get => employees[index];  
        set => employees[index] = value;  
    }  
}
```

```
    public int Count  
        => employees.Count;
```

```
    public bool IsReadOnly  
        => false;
```

```
    public void Add(T item)  
        => employees.Add(item);
```

ДУ «Житомирська політехніка».23.122.02.000 – Лр5

Змн.	Дод.	Підпис	Дата	Звіт з лабораторної роботи ФІКТ Гр. КН-21-2			
Розроб.	Волошиць А.Ю.						
Перевір.	Левківський В.Л.						
Керівник							
Н. контр.							
Зав. каф.							
				Лім.	Арк.	Аркуші	
					1		

```

public void Clear()
    => employees.Clear();

public bool Contains(T item)
    => employees.Contains(item);

public void CopyTo(T[] array, int arrayIndex)
    => employees.CopyTo(array, arrayIndex);

public int IndexOf(T item)
    => employees.IndexOf(item);

public void Insert(int index, T item)
    => employees.Insert(index, item);

public bool Remove(T item)
    => employees.Remove(item);

public void RemoveAt(int index)
    => employees.RemoveAt(index);

public IEnumerator<T> GetEnumerator()
    => employees.GetEnumerator();

IEnumerator IEnumerable.GetEnumerator()
    => GetEnumerator();

// Пошук за умовою
public IEnumerable<T> Search(Func<T, bool> predicate)
    => employees.Where(predicate);

// Оновлення інформації про співробітника за індексом
public void UpdateEmployeeInfo(int index, T newInfo)
    => employees[index] = newInfo;

// Сортювання за ключем
public void SortBy(Func<T, IComparable> keySelector)
    => employees.Sort((x, y) => keySelector(x).CompareTo(keySelector(y)));

// Пошук за умовою (перегрузка для рядка)
public IEnumerable<T> Search(string searchTerm)
    => Search(e => e?.ToString()?.Contains(searchTerm, StringComparison.OrdinalIgnoreCase) ?? false);

// Сортювання за ключем (перегрузка для властивості рядка)
public void SortBy(Func<T, string> keySelector)
    => employees.Sort((x, y) => string.Compare(keySelector(x), keySelector(y), StringComparison.OrdinalIgnoreCase));
}

class Program
{
    static void Main()
    {
        EmployeeManagement<Employee> employeeManagement = new();

        // Додавання співробітників
        Employee employee1 = new() { FullName = "John Doe", Position = "Developer", Salary = 50000, Email =
"john@example.com" };
        Employee employee2 = new() { FullName = "Jane Smith", Position = "Manager", Salary = 60000, Email =
"jane@example.com" };

        employeeManagement.Add(employee1);
        employeeManagement.Add(employee2);

        // Видалення співробітників
        employeeManagement.Remove(employee1);

```

		Волинець А.Ю..			ДУ «Житомирська політехніка».23.122.02.000 – Лр5	Арк.
		Левківський В. Л.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

// Пошук співробітників за параметрами
var searchResults = employeeManagement.Search(e => e.Position == "Manager");
Console.WriteLine("Search results:");
foreach (var result in searchResults)
{
    Console.WriteLine($"Name: {result.FullName}, Position: {result.Position}");
}

// Сортювання співробітників за параметрами
employeeManagement.SortBy(e => e.Salary);
}
}
}

```

Завдання 2. Створіть додаток для обліку книг, який зберігатиме наступну інформацію:

- назва книги;
- П.І.Б. автора;
- жанр книги;
- рік видання. Для зберігання даних працівників використовуйте клас `LinkedList`. Додаток має надавати таку функціональність:

- додавати книги;
- видаляти книги;
- змінювати інформацію про книги;
- пошук книг за параметрами;
- вставити книги у початок списку;
- вставити книги у кінець списку;
- вставити книги у певну позицію;
- видалити книги з початку списку;
- видалити книги з кінця списку;
- видалити книги з певної позиції.

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;

```

```

namespace Task2

```

```

{
    [Serializable]
    public class Book : ISerializable
    {
        public string? Title { get; set; }
        public string? Author { get; set; }
        public string? Genre { get; set; }
        public int YearPublished { get; set; }
    }

```

```

    public Book()
    {
    }

```

```

    public Book(string title, string author, string genre, int yearPublished)
    {
        Title = title;
        Author = author;
    }

```

		Волинець А.Ю.			ДУ «Житомирська політехніка».23.122.02.000 – Лр5	Арк.
		Левківський В. Л.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Genre = genre;
        YearPublished = yearPublished;
    }

    protected Book(SerializationInfo info, StreamingContext context)
    {
        Title = info.GetString(nameof(Title));
        Author = info.GetString(nameof(Author));
        Genre = info.GetString(nameof(Genre));
        YearPublished = info.GetInt32(nameof(YearPublished));
    }

    public void GetObjectData(SerializationInfo info, StreamingContext context)
    {
        info.AddValue(nameof(Title), Title);
        info.AddValue(nameof(Author), Author);
        info.AddValue(nameof(Genre), Genre);
        info.AddValue(nameof(YearPublished), YearPublished);
    }
}

public class BookManagement<T> : ICollection<T>, IReadOnlyCollection<T>, ISerializable, IDeserializationCallback
{
    private LinkedList<T> books = new();

    public T this[int index]
    {
        get
        {
            if (index < 0 || index >= Count)
                throw new ArgumentOutOfRangeException(nameof(index));

            return books.ElementAt(index);
        }
        set
        {
            if (index < 0 || index >= Count)
                throw new ArgumentOutOfRangeException(nameof(index));

            var node = books.Find(books.ElementAt(index));
            if (node != null)
                node.Value = value;
        }
    }

    public int Count => books.Count;

    public bool IsReadOnly => false;
    public bool IsSynchronized => false;
    public object SyncRoot => this;

    public void Add(T item) => books.AddLast(item);
    public bool Remove(T item) => books.Remove(item);
    public void Clear() => books.Clear();
    public bool Contains(T item) => books.Contains(item);
    public void CopyTo(T[] array, int arrayIndex) => books.CopyTo(array, arrayIndex);

    public IEnumerable<T> Search(Func<T, bool> predicate) => books.Where(predicate);

    public void Insert(int index, T item)
    {
        if (index < 0 || index > Count)
            throw new ArgumentOutOfRangeException(nameof(index));

        if (index == Count)
            Add(item);
    }
}

```

		Волинець А.Ю..			ДУ «Житомирська політехніка».23.122.02.000 – Лр5	Арк.
		Левківський В. Л.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

        else
        {
            var node = books.Find(books.ElementAt(index));
            if (node != null)
                books.AddBefore(node, item);
        }
    }

    public void InsertAtBeginning(T item) => books.AddFirst(item);
    public void InsertAtEnd(T item) => books.AddLast(item);
    public void RemoveFromBeginning() => books.RemoveFirst();
    public void RemoveFromEnd() => books.RemoveLast();

    public void RemoveAt(int index)
    {
        if (index < 0 || index >= Count)
            throw new ArgumentOutOfRangeException(nameof(index));

        var node = books.Find(books.ElementAt(index));
        if (node != null)
            books.Remove(node);
    }

    public void GetObjectData(SerializationInfo info, StreamingContext context)
    {
        info.AddValue(nameof(books), books.ToList());
    }

    public void OnDeserialization(object? sender)
    {
        if (books == null)
            books = new LinkedList<T>();
        else
            books = new LinkedList<T>(books);
    }

    public IEnumerator<T> GetEnumerator()
    {
        return books.GetEnumerator();
    }

    IEnumerator IEnumerable.GetEnumerator()
    {
        return GetEnumerator();
    }
}

class Program
{
    static void Main()
    {
        BookManagement<Book> bookManagement = new();

        Book book1 = new() { Title = "Book1", Author = "Author1", Genre = "Genre1", YearPublished = 2022 };
        Book book2 = new() { Title = "Book2", Author = "Author2", Genre = "Genre2", YearPublished = 2023 };

        bookManagement.Add(book1);
        bookManagement.Add(book2);

        bookManagement.Remove(book1);
    }
}

```

		Волинець А.Ю.			ДУ «Житомирська політехніка».23.122.02.000 – Лр5	Арк.
		Левківський В. Л.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

Завдання 3. Створіть додаток для менеджменту співробітників та паролів, який має зберігати наступну інформацію:

- логіни співробітників;
- паролі співробітників. Для зберігання інформації використовуйте Dictionary. Додаток має надавати таку функціональність:

Dictionary. Додаток має надавати таку функціональність:

- додати логін і пароль співробітника;
- видалити логін співробітника;
- змінити інформацію про логін і пароль співробітника;
- отримати інформацію про пароль за логіном співробітника. Закомітити рішення у репозиторій і надати доступ викладачам

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Runtime.Serialization;

namespace Task3
{
    [Serializable]
    public class EmployeeManager<TKey, TValue> : IDictionary<TKey, TValue>, ISerializable, IDeserializationCallback
        where TKey : notnull
    {
        private Dictionary<TKey, TValue> employeeData = new();

        public TValue this[TKey key]
        {
            get => employeeData[key];
            set => employeeData[key] = value;
        }

        public ICollection<TKey> Keys => employeeData.Keys;
        public ICollection<TValue> Values => employeeData.Values;
        public int Count => employeeData.Count;
        public bool IsReadOnly => false;

        public void Add(TKey key, TValue value)
        {
            employeeData.Add(key, value);
        }

        public void Add(KeyValuePair<TKey, TValue> item)
        {
            employeeData.Add(item.Key, item.Value);
        }

        public void Clear()
        {
            employeeData.Clear();
        }

        public bool Contains(KeyValuePair<TKey, TValue> item)
        {
            return employeeData.Contains(item);
        }

        public bool ContainsKey(TKey key)
        {
            return employeeData.ContainsKey(key);
        }
    }
}
```

		Волинець А.Ю..			ДУ «Житомирська політехніка».23.122.02.000 – Лр5	Арк.
		Левківський В. Л.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

public void CopyTo(KeyValuePair<TKey, TValue>[] array, int arrayIndex)
{
    ((ICollection<KeyValuePair<TKey, TValue>>)employeeData).CopyTo(array, arrayIndex);
}

public IEnumerator<KeyValuePair<TKey, TValue>> GetEnumerator()
{
    return employeeData.GetEnumerator();
}

public bool Remove(TKey key)
{
    return employeeData.Remove(key);
}

public bool Remove(KeyValuePair<TKey, TValue> item)
{
    return employeeData.Remove(item.Key);
}

public bool TryGetValue(TKey key, out TValue value)
{
    return employeeData.TryGetValue(key, out value!);
}

public void GetObjectData(SerializationInfo info, StreamingContext context)
{
    info.AddValue(nameof(employeeData), employeeData);
}

public void OnDeserialization(object? sender)
{
    if (employeeData == null)
        employeeData = new Dictionary<TKey, TValue>();
    else
        employeeData = new Dictionary<TKey, TValue>(employeeData);
}

IEnumerator IEnumerable.GetEnumerator()
{
    return GetEnumerator();
}
}

class Program
{
    static void Main()
    {
        EmployeeManager<string, string> employeeManager = new()
        {
            // Додавання логіну та паролю
            { "john_doe", "hashedPassword123" },
            { "jane_smith", "hashedSecurePassword" }
        };

        // Зміна інформації про логін і пароль
        employeeManager["john_doe"] = "newHashedPassword123";

        // Видалення логіну співробітника
        employeeManager.Remove("jane_smith");

        // Отримання інформації про пароль за логіном
        if (employeeManager.TryGetValue("john_doe", out var password))
        {
            Console.WriteLine($"Password for john_doe: {password}");
        }
    }
}

```

		Волинець А.Ю..			ДУ «Житомирська політехніка».23.122.02.000 – Лр5	Арк.
		Левківський В. Л.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
  }
}

```

Висновки:

У ході лабораторної роботи було успішно вивчено та впроваджено концепції генеріс-колекцій у мові програмування C#. Це дозволило створити додатки для обліку співробітників, книг та управління логінами та паролями з використанням універсальних та гнучких структур даних. Реалізований функціонал включає додавання, видалення, зміну інформації, пошук, сортування, а також різноманітні операції вставки та видалення елементів. Використання генеріс-колекцій забезпечило ефективне та універсальне управління даними в програмах.

		Волинець А.Ю..			ДУ «Житомирська політехніка».23.122.02.000 – Лр5	Арк.
		Левківський В. Л.				
Змн.	Арк.	№ докум.	Підпис	Дата		8