



# Курс **Advanced – C#**

ЗАНЯТИЯ № - МОБА LING



# LINQ

## Що таке LINQ

Language-Integrated Query - представляє просту та зручну мову запитів до джерела даних. Як джерело даних може бути об'єкт, що реалізує інтерфейс `IEnumerable` (наприклад, стандартні колекції, масиви), набір даних `DataSet`, документ XML. Але незалежно від типу джерела LINQ дозволяє застосувати до всіх той самий підхід для вибірки даних.



# Плюси мови LINQ

Щоб зрозуміти переваги приведемо приклад:

Стандартний код:

```
string[] people = { "Tom", "Bob", "Sam", "Tim", "Tomas", "Bill" };

// Створюємо список результатів
var selectedPeople = new List<string>();
// проходимо по масиву
foreach (string person in people)
{
    //якщо строка починається з Т, то додаємо в список
    if (person.ToUpper().StartsWith("T"))
        selectedPeople.Add(person);
}
// сортуємо
selectedPeople.Sort();

foreach (string person in selectedPeople)
    Console.WriteLine(person);
```

Мовою LINQ:

```
string[] people = { "Tom", "Bob", "Sam", "Tim", "Tomas", "Bill" };

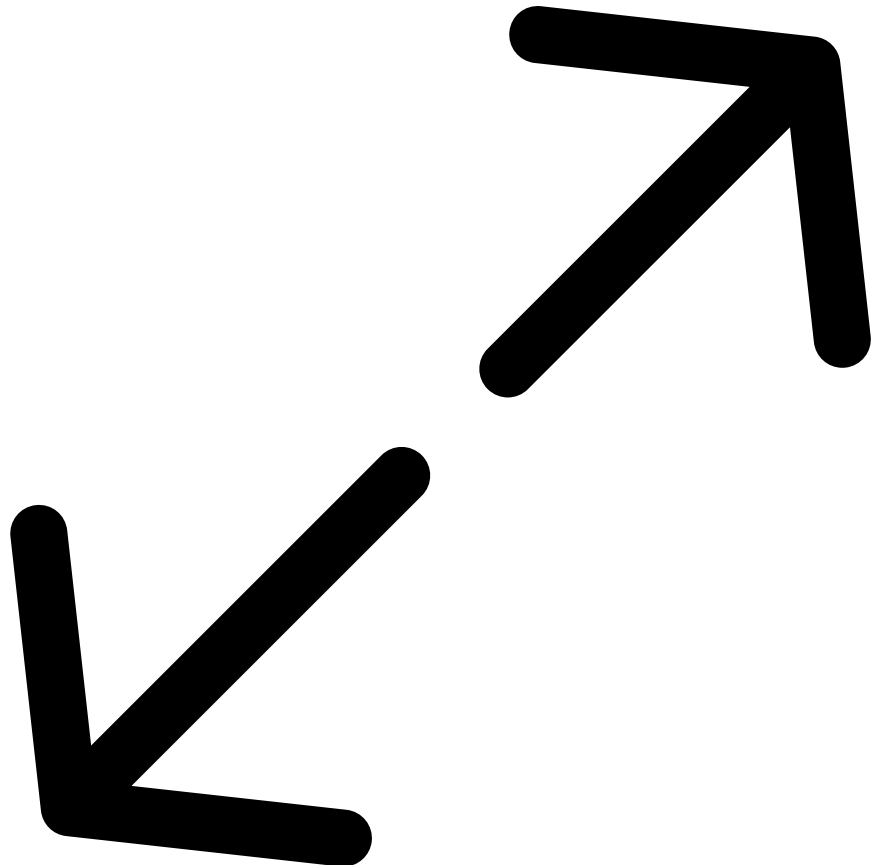
// Створюємо список результатів
var selectedPeople = from p in people // передаємо елемент из people у
змінну p
                      where p.ToUpper().StartsWith("T") //фільтрація
                      orderby p // сортування
                      select p; // вибірка

foreach (string person in selectedPeople)
    Console.WriteLine(person);
```





# Методи розширення LINQ



Крім стандартного синтаксису `from .. in .. select` для створення запиту LINQ, ми можемо застосовувати спеціальні методи розширення, які визначені для інтерфейсу `IEnumerable`. Як правило, ці методи реалізують ту ж функціональність, що і оператори типу LINQ

Наприклад:

```
var selectedPeople = people.Where(p => p.ToUpper().StartsWith("T")).OrderBy(p => p);
```



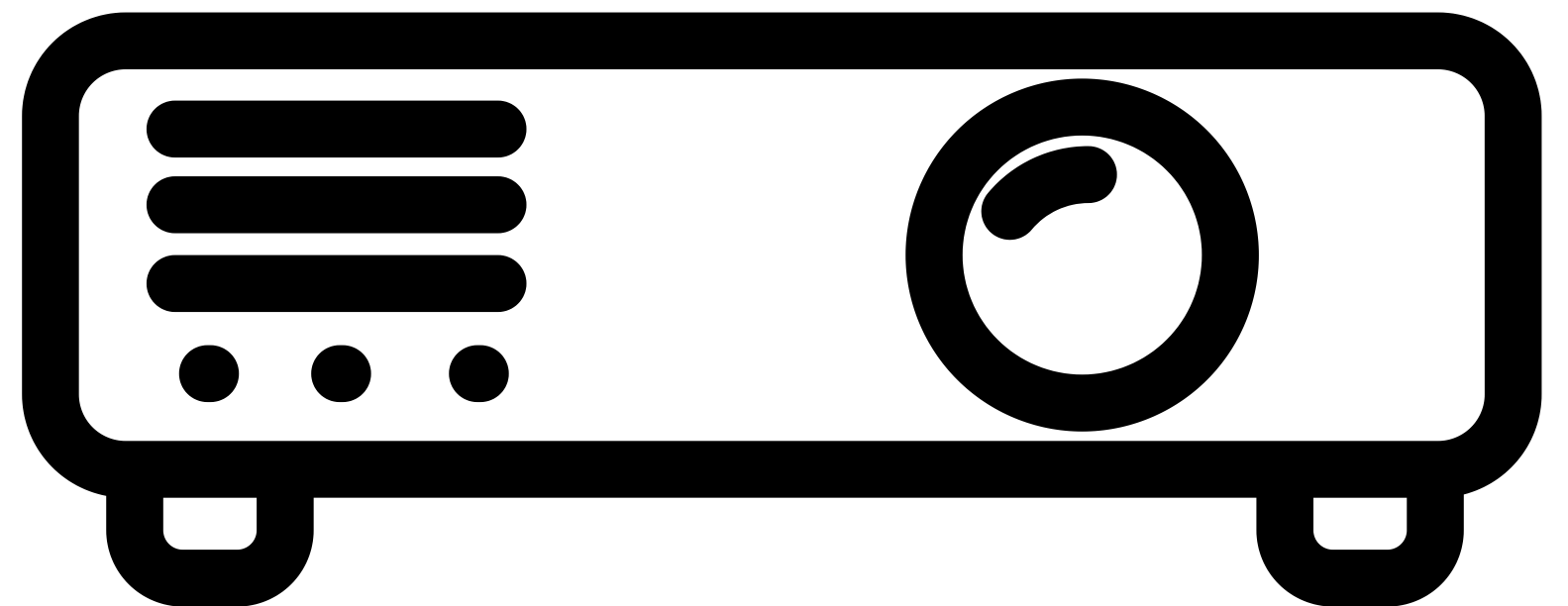
# Проекція даних

Проекція дозволяє перетворити об'єкт одного типу на об'єкт іншого типу. Для проекції використовується оператор `select`

```
var people = new List<Person>
{
    new Person ("Tom", 23),
    new Person ("Bob", 27),
    new Person ("Sam", 29),
    new Person ("Alice", 24)
};
```

```
var names = from p in people select p.Name;
```

```
foreach (string n in names)
    Console.WriteLine(n);
```





# Змінні у запитах

Іноді виникає потреба зробити у запитах LINQ якісь додаткові проміжні обчислення. Для цього ми можемо задати у запитах свої змінні за допомогою оператора `let` :

```
var personnel = from p in people
    let name = $"Mr. {p.Name}"
    let year = DateTime.Now.Year - p.Age
    select new
{
    Name = name,
    Year = year
};
```





Дякую за увагу