



LINQ

Celem lekcji jest omówienie LINQ w języku programowania C#.
Poznamy też nowe operacje, które można wykonać do uzyskania
elementów z bazy danych.



Przypomnienie



1. Do czego służy Entity Framework?

- **Entity Framework** to narzędzie dla programistów .NET, które zamienia dane z bazy na obiekty w kodzie i odwrotnie. Ułatwia pracę z bazami danych, automatyzując wiele zadań i pozwalając skupić się na tworzeniu funkcji aplikacji.

2. Co to jest Model Builder?

- **Model Builder** w Entity Framework to narzędzie do definiowania schematu bazy danych w kodzie, umożliwiające konfigurację tabel, kolumn i relacji między nimi. Pozwala nam na precyzyjne ustawienie właściwości modeli.

3. Na czym polega programowanie asynchroniczne?

- **Programowanie asynchroniczne** polega na wykonywaniu operacji, które mogą trwać długo, bez blokowania głównego wątku aplikacji, dzięki czemu aplikacja pozostaje responsywna. Pozwala to na wykonywanie innych zadań w tle, podczas gdy operacje takie jak dostęp do bazy danych czy pobieranie zasobów z sieci są przetwarzane równolegle.





LINQ (Language Integrated Query) to narzędzie w języku C# umożliwiające wykonywanie zapytań na różnych źródłach danych (takich jak kolekcje, bazy danych) w sposób zintegrowany z językiem.

1. **Czytelność i zwięzłość kodu:** LINQ umożliwia pisanie złożonych zapytań w sposób czytelny i zwięzły.
2. **Różnorodne źródła danych:** LINQ działa z różnymi typami danych, od kolekcji w pamięci, przez bazy danych.
3. **Silne typowanie:** LINQ jest silnie typowane, co oznacza, że błędy mogą być wykrywane na etapie kompilacji. Oznacza to, że każda zmienna, funkcja, parametr czy wynik operacji ma jasno zdefiniowany typ, który nie może być zmieniony w trakcie działania programu bez jawnej konwersji.
4. **Łatwe filtrowanie, sortowanie i grupowanie danych:** LINQ oferuje wbudowane metody do filtrowania, sortowania i grupowania danych.



Wyrażenia lambda



Wyrażenia lambda są fundamentalnym elementem LINQ (Language Integrated Query) w C#. Pozwalają na zwięzłe i elastyczne definiowanie zapytań do różnych źródeł danych, takich jak kolekcje obiektów albo bazy danych. Wyrażenia lambda są często używane jako argumenty do metod LINQ, takich jak Where, Select, OrderBy i innych.

Wyrażenia lambda mają postać:

- *(parametry) => wyrażenie*
- *(parametry) => { instrukcje }*

```
List<int> liczby = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
```

```
var parzyste = liczby.Where(n => n % 2 == 0).ToList();
```



FirstOrDefault



FirstOrDefault bierze pierwsze wystąpienie danego elementu bądź używa domyślnej wartości, jeżeli nie może znaleźć żadnego elementu.

- Dla typów referencyjnych, takich jak obiekty domyślną wartością jest **null**.
- Oznacza to, że jeśli kolekcja jest pusta lub żaden element nie spełnia warunku, FirstOrDefault zwróci null.





Distinct służy do wyświetlenia tylko **unikalnych** wyników.

Wybranie tylko unikalnych aktorów:

```
var wszyscyAktorzyUnikalni = filmy  
    .SelectMany(f => f.Aktorzy)  
    .Distinct()  
    .ToList();
```

