LINQ w praktyce — wyniki uczniów

Czas: 2 × 45 min | praca indywidualna | Wykład 1.19

Cel: Przeanalizujesz listę uczniów i ich ocen zapytaniami LINQ: filtrowaniem, sortowaniem, grupowaniem i agregacją.

Po zajęciach potrafisz:

- filtruje i sortuje dane metodami `Where`, `OrderBy`, `ThenBy`
- grupuje dane `GroupBy` i liczy agregaty (`Count`, `Average`, `MaxBy`)
- spłaszcza kolekcje zagnieżdżone `SelectMany`
- zapisuje zapytanie w składni metod i składni zapytania

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu
- pendrive lub konto w repozytorium do zapisania projektu

Oprogramowanie

- **Visual Studio Community 2026 (obciążenie „Programowanie aplikacji klasycznych platformy .NET")** — darmowe dla uczniów i nauczycieli; w pracowni zainstalowane visualstudio.microsoft.com/pl/downloads
- **.NET 10 SDK (LTS)** — instalowany razem z Visual Studio; sprawdzenie: `dotnet --version` dotnet.microsoft.com/pl-pl/download/dotnet/10.0

Bezpieczeństwo

- Zapisuj projekt w swoim folderze (np. `C:\Users\<login>\source\repos`), a na koniec zajęć skopiuj go na pendrive lub wyślij do repozytorium.
- Nie uruchamiaj programów i pakietów z nieznanych źródeł; pakiety NuGet instaluj tylko te, które wskazuje zadanie.

Przebieg zadania

1 Dane

Utwórz projekt `Analiza` i rekord ucznia.

1. `record Uczen(string Imie, string Nazwisko, string Klasa, List<int> Oceny)` z właściwością `Srednia`.
2. Utwórz listę co najmniej 8 uczniów z 3 klas (możesz użyć danych z rozwiązania nauczyciela lub własnych).
3. Zabezpiecz średnią dla ucznia bez ocen (zwróć 0).

2 Zapytania podstawowe

Każde zapytanie wypisz z nagłówkiem.

```
var klasy = uczniowie.GroupBy(u => u.Klasa)
    .Select(g => new { Klasa = g.Key, Liczba = g.Count(),
                       Srednia = g.Average(u => u.Srednia) })
    .OrderBy(k => k.Klasa);
```

1. Lista uczniów posortowana malejąco po średniej, przy równych — alfabetycznie po nazwisku.
2. Uczniowie zagrożeni — mający choć jedną ocenę 1 (`Contains`).
3. Uczniowie ze średnią co najmniej 4,75 (kandydaci do wyróżnienia).
4. Trzech najlepszych (`Take(3)`) zapisanych w składni zapytania (`from … orderby … select`).

3 Grupowanie

Zestawienia dla klas.

1. Liczba uczniów i średnia w każdej klasie (szablon powyżej).
2. Najlepszy uczeń w każdej klasie (`MaxBy` w grupie).
3. Klasa z najwyższą średnią.

4 Statystyki ocen

Wszystkie oceny razem.

1. Spłaszcz oceny `SelectMany` i policz: liczbę ocen, średnią, liczbę szóstek.
2. Rozkład ocen: ile jedynek, dwójek… (`GroupBy(o => o)`).
3. Sprawdź `Any` i `All`: czy ktoś ma wyłącznie oceny 5 i 6?
4. Sprawdź wyniki dwóch zapytań ręcznie (kalkulator).
5. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod programu
- ☐ wyniki wszystkich zapytań
- ☐ ręczne sprawdzenie dwóch wyników
- ☐ porównanie składni metod i składni zapytania

Pytania kontrolne

1. Co to jest wykonanie odroczone (deferred execution) w LINQ?
2. Czym różni się `Select` od `SelectMany`?
3. Co zwraca `GroupBy` i jak odczytać klucz grupy?
4. Kiedy `First()` zgłosi wyjątek i czym go zastąpić?

Kryteria oceny

Kryterium	Punkty
Dane i rekord ze średnią	3
Zapytania podstawowe (4)	6
Grupowanie (3)	5
Statystyki ocen	3
Sprawdzenie ręczne i pytania	3
Razem	**20**

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).