

Programowanie komputerów.

Ćwiczenia 12.

LINQ

Przed przystąpieniem do zadań należy przeanalizować treść wykładu na temat **Technologii Linq** oraz z rozdziałem 6. podręcznika „*Programowanie komputerów w zadaniach. Język C#*” (zadania 7.1-7.6).

Zadanie 1.

Napisz program, który definiuje listę 20 liczb całkowitych (zainicjalizowaną przykładowymi wartościami z zakresu od 0 do 100), a następnie wykonaj przy pomocy zapytań LINQ następujące operacje:

- a) Utworzenie sekwencji z liczbami z listy, których różnica względem wartości maksymalnej jest nie większa niż 25. Sekwencja ma być uporządkowana malejąco.
- b) Utworzenie sekwencji zawierającej 10 najmniejszych liczb z listy wejściowej, po uprzednim pozbawianiu jej duplikatów.
- c) Pogrupowanie danych z listy według liczby cyfr: liczby jednocyfrowe, liczby dwucyfrowe i liczby trzycyfrowe. Sekwencja ma być uporządkowana rosnąco, co dotyczy zarówno grup jak i liczb w grupach.

Wszystkie wymienione działania dotyczą kolekcji źródłowej (listy). Wyniki operacji należy zaprezentować na ekranie. Wykonaj to zadanie w dwóch wersjach: w składni zapytań i w składni płynnej.

Zadanie 2.

Napisz program, który definiuje dwie listy:

```
List<string> biegacze = new List<string> {"Ania", "Tomek", "Jola",  
"Adam", "Jan"};
```

```
List<string> plywacy = new List<string> {"Tomek", "Adam", "Robert",  
"Tomek"};
```

a następnie wykonaj przy pomocy zapytań LINQ następujące operacje:

- a) Utworzenie sekwencji zawierającej imiona tych osób, które uprawiają zarówno bieganie jak i pływanie.
- b) Utworzenie sekwencji zawierającej imiona tych osób, które biegają, ale nie pływają.
- c) Utworzenie sekwencji zawierającej imiona tych osób, które uprawiają którykolwiek ze sportów – bieganie lub pływanie. Sekwencja wynikowa ma być pogrupowana osobno dla dziewczyn (imiona kończące się na „a”) oraz dla chłopców.

Wszystkie wymienione działania dotyczą kolekcji źródłowej (listy). Wyniki operacji należy zaprezentować na ekranie.

W zadaniu należy korzystać z operatów z grupy „Łączenie i operacje na zbiorach” (poniżej fragment z tabeli 7.1 z podręcznika *Programowanie komputerów w zadaniach. Język C#*).

Łączenie i operacje na zbiorach	Join – zwraca sekwencję z kombinacją wartości z dwóch sekwencji dopasowanych na podstawie klucza (tj. wspólnego składnika pozwalającego na dopasowanie); operator jest odpowiednikiem klauzuli INNER JOIN w SQL Concat – zwraca sekwencję będącą połączeniem dwóch innych Union – zwraca sekwencję będącą połączeniem dwóch innych bez duplikatów Intersect – zwraca te elementy, które występują w obu sekwencjach Except – zwraca elementy, które są w pierwszej sekwencji, ale nie ma ich w drugiej sekwencji
---------------------------------	--

Zadanie 3.

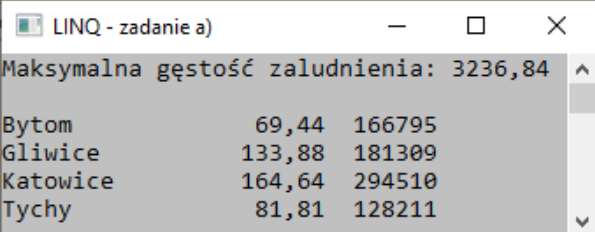
Na podstawie tabeli 7.2 z podręcznika wykonaj przy pomocy zapytań LINQ następujące operacje:

Tabela 7.2. Dane o miastach

Nazwa	Powierzchnia (w km ²)	Ludność
Chorzów	33,50	108434
Tychy	81,81	128211
Katowice	164,64	294510
Bytom	69,44	166795
Gliwice	133,88	181309

a)

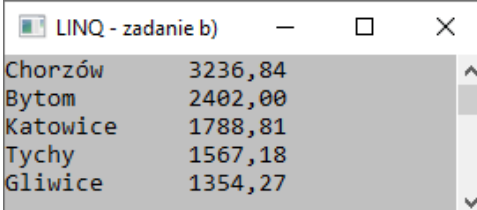
Utworzenie sekwencji zawierającej miasta wszystkie oprócz tego, które ma maksymalną gęstość zaludnienia. Sekwencja ma być uporządkowana alfabetycznie według nazw miast (przykładowy wynik programu na poniższym zrzucie ekranu).



Maksymalna gęstość zaludnienia: 3236,84		
Bytom	69,44	166795
Gliwice	133,88	181309
Katowice	164,64	294510
Tychy	81,81	128211

b)

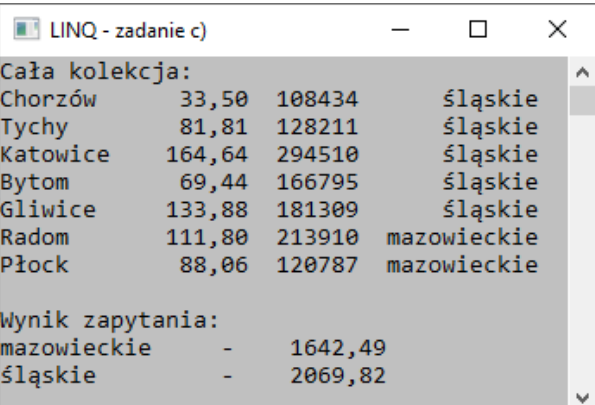
Utworzenie sekwencji zawierającej dane: nazwa miasta i gęstość zaludnienia, uporządkowanej według gęstości w kolejności malejącej (przykładowy wynik programu na poniższym zrzucie ekranu).



Chorzów	3236,84
Bytom	2402,00
Katowice	1788,81
Tychy	1567,18
Gliwice	1354,27

c)

Do klasy Miasto dodaj właściwość z nazwą województwa i dopisz do listy kilka miast z innego województwa. Należy pogrupować miasta według województwa i dla wyodrębnionych grup utworzyć sekwencję zawierającą dane: województwo i średnia gęstość zaludnienia w danym województwie. Sekwencja ma być uporządkowana według województw.



Cała kolekcja:			
Chorzów	33,50	108434	śląskie
Tychy	81,81	128211	śląskie
Katowice	164,64	294510	śląskie
Bytom	69,44	166795	śląskie
Gliwice	133,88	181309	śląskie
Radom	111,80	213910	mazowieckie
Płock	88,06	120787	mazowieckie
Wynik zapytania:			
mazowieckie	-	1642,49	
śląskie	-	2069,82	

Zadanie 4.

Napisz program, który odczytuje zawartość pliku *ubrania.csv* (dołączony do prezentacji) do kolekcji obiektów klasy *Ubrania*.

Uwaga. W niektórych rekordach są nieprawidłowe dane (tekst w polu z ceną), program ma odrzucić (zignorować) takie rekordy podczas odczytu danych do kolekcji obiektów.

Następnie wykonaj przy pomocy LINQ operacje:

a) Selekcja ubrań według kryteriów: kurtki od producenta CIMINY. Należy na konsoli wyświetlić dane o ubraniach spełniających te warunki (w postaci tabeli).

b) Należy pogrupować dane według podkategorii i wyświetlić na konsoli średnie cen ubrań w poszczególnych podkategoriach. Dane mają być uporządkowane alfabetycznie według grupy (czyli najpierw Bluzki obok średnia cen bluzek, poniżej kolejne podkategorie ze swoimi średnimi, a na końcu Topy).

c) Należy pogrupować dane według kategorii (Damskie/Męskie) i dane z wyodrębnionych grup zapisać do osobnych plików *xml* (jeden plik ma zawierać ubrania damskie, a drugi męskie). Do plików mają być wpisane dane uporządkowane według podkategorii i producenta.

Uwaga. Program ma poprawnie działać (bez zmian kodu) także wówczas, gdyby do pliku csv dodano inne kategorie ubrań (np. Unisex, Niemowlęce) – przetestuj swój program pod tym kątem.