



Wprowadzenie

W celu wykonania ćwiczeń wymagane jest użycie następujących bibliotek:

System.Linq	Provides classes and interfaces that support LINQ queries
System.Collections.Generic	Allows users to create strongly typed collections that provide better type safety and performance than nongeneric strongly typed collections (LINQ to Objects)
System.Data.Linq	Provides the functionality to use LINQ to access relational databases (LINQ to SQL)
System.Xml.Linq	Provides functionality for accessing XML documents using LINQ (LINQ to XML)
System.Data.Linq.Mapping	Designates a class as an entity class associated with a database

Bardziej szczegółowe informacje o LINQ znajdują się w wykładzie i materiałach dodatkowych.

ZADANIE 1 – GENEROWANIE KWADRATÓW NIEKTÓRYCH LICZB – 1PKT

W zadaniu nie można używać pętli. Proszę wczytać ze standardowego wejścia liczbę N . Za pomocą połączenia metod `Range`, `Where`, `Select`, `OrderBy` stworzyć kolekcję kwadratów liczb od $1..N$. Liczba, która będzie podnoszona do kwadratu musi spełniać założenia(koniunkcja):

- Różna od 5,
- Różna od 9
- Nieparzysta lub wielokrotność 7,

Czyli np. 14 spełnia(różna od 5 i różna od 9 i wielokrotność 7), ale 10 nie spełnia.

Następnie proszę wypisać na standardowe wyjście:

- Sumę wszystkich elementów kolekcji
- Liczbę wszystkich elementów w kolekcji
- Pierwszy element kolekcji
- Ostatni element kolekcji

- Liczbę elementów kolekcji
- Trzeci element kolekcji

ZADANIE 2 – SUMA ELEMENTÓW LOSOWEJ MACIERZY – 1PKT

W zadaniu nie można używać pętli. Proszę wczytać liczby N i M. Proszę wygenerować listę list rozmiaru NxM i zainicjalizować ją losowymi liczbami całkowitymi. Następnie proszę użyć metody SelectMany w celu zsumowania wszystkich elementów i wypisania ich na standardowe wyjście.

ZADANIE 3 – GRUPOWANIE NAZW MIAST – 1PKT

Proszę napisać program, który wczytuje ze standardowego wejścia listę miast (oddzielonych nową linią). Proszę użyć metody GroupBy oraz OrderBy, aby pogrupować miasta po pierwszej literze oraz posortować wewnątrz tej grupy w porządku leksykograficznym. Do przechowywania zgrupowanych danych należy użyć słownika. Miasta wczytujemy aż do napotkania symbolu „X”. Następnie program oczekuje podania kolejnych liter. Po podaniu litery program wypisuje posortowaną listę miast zaczynającą się na daną literę lub „PUSTE”.

Wejście	Wyjście
Paryż Praga Kraków Pcim Płock X P B	Paryż, Pcim, Płock, Praga PUSTE

ZADANIE 4 – PRZEMIESZCZANIE SIĘ POMIĘDZY LOKACJAMI – 2PKT

Proszę zmodyfikować naszą grę, aby posiadała kolekcję możliwych lokacji. Na raz możemy być tylko w jednej lokacji. Zmodyfikujmy menu w ten sposób, aby umożliwić przemieszczanie się. Wybór opcji [T] (od travel) spowoduje zmianę lokacji:

```
Znajdujesz się w : {NazwaLokalizacji}. Co chcesz zrobić?
[1] Porozmawiaj z {NazwaNPC1}
[2] Porozmawiaj z {NazwaNPC2}
[T] Podróżuj
[X] Zamknij program
```

Po wybraniu opcji T proszę wyczyścić ekran konsoli i pokazać menu z dostępnymi lokacjami:

```
Znajdujesz się w : {NazwaLokalizacji}. Gdzie chcesz wyruszyć?
[1] Nazwa 1
[2] Nazwa 2
[X] Powrót
```

Wybranie jakiejś lokalizacji ma powodować wyświetlenie jej menu za pomocą już napisanej metody z poprzednich zadań.

Wybranie opcji powrót ma powodować powrót do aktualnej lokalizacji.

Oczywiście na liście lokalizacji nie ma być tej, w której znajdujemy się aktualnie. Ponadto dodajmy do klasy lokalizacji właściwość IsUnlocked. Proszę stworzyć 6 przykładowych lokalizacji, niech

dwie z nich będą miały ustawione `IsUnlocked` na `false` – bohater odblokuje je dopiero wraz z postępem w grze.

Podczas wyświetlania menu proszę użyć metody `LINQ Where` do odfiltrowania listy dostępnych lokalizacji tak, aby wyświetlać tylko te odblokowane różne od aktualnej. Ponadto proszę użyć metody `LINQ OrderBy` do posortowania lokalizacji leksykograficznie.

Proszę połączyć filtrowanie i sortowanie w łańcuch metod.