

## Lista zadań

### Przygotować program demonstrujący użycie zaimplementowanych rozwiązań.

1. Wykonać zadanie znalezienia 10 najczęstszych słów z listy 7 za pomocą LINQ (najlepiej za pomocą jednego wyrażenia).
2. Korzystając z klas z wykładu 8 stworzyć metodę z zapytaniem grupującym posortowanych studentów wg nazwiska (i numeru indeksu) w grupy po n elementów (n - parametr metody).
3. Korzystając z klas z wykładu 8 posortować tematy (Topics) wg częstości występowania wśród wszystkich studentów oraz w drugim przypadku z wcześniejszym podziałem na płcie.
4. Korzystając z klas z wykładu 8 stworzyć klasę Topic, pamiętając temat (i jego identyfikator). Stworzyć nową klasę StudentWithTopics, która różni się tym od klasy Student, że tematy są pamiętane jako lista identyfikator tematów (a nie string-i). Stworzyć listę właściwych tematów. Napisać zapytanie przekształcające listę obiektów Student, na listę obiektów klasy StudentWithTopics.
  - a. Dla chętnych – generacja listy tematów poprzez zapytanie
  - b. Dla chętnych – generacja listy tematów i listy nowego typu studentów w jednym zapytaniu.
  - c. Dla chętnych – przygotować drugą wersję (oprócz tej pierwszej), gdzie zamiast pamiętać listę tematów stworzymy klasę StudentWithoutTopics, natomiast relację „n do n” pamiętamy w nowej liście z elementami typu StudentToTopic pamiętającej pary identyfikatorów: studenta i tematu.
5. Korzystając z mechanizmu odbicia uruchomić metodę liczącą sumaryczną nośność w klasie wykonanej dla listy 6. Wykorzystać metody `GetMethod()` oraz `Invoke()`.
  - a. Dla chętnych – wykorzystać mechanizm odbicia to wywołania konstruktorów pojazdów.
6. Zadania 2-4 można również wykonać w narzędziu LINQPad wybierając analogiczne tabele z demonstracyjnej bazy danych i analogiczne pola z wybranych tabel.

Data I: Spotkanie 10 (max 100 punktów)

Data II: Spotkanie 11 (max 80 punktów)

Data III: Spotkanie 12 (max 50 punktów)