

1. Stwórz klasę `Osoba`, dodaj w niej prywatne pola imię i wiek. Następnie dodaj konstruktor domyślny i parametryczny. Dodaj metody na oddzielne ustawienie i pobranie wieku i imienia.
2. Rozważ następujące kolekcje: `Stack`, `Stack<T>`, `Queue`, `Queue<T>`, `ArrayList`, `List<T>`, `Dictionary<TKey, TValue>`, `SortedDictionary<TKey, TValue>`, `SortedList<TKey, TValue>`, `LinkedList<T>`, `BitArray`, `HashSet<T>`, `SortedSet<T>`, `Hashtable`, `OrderedDictionary`, `ListDictionary`, `HybridDictionary`, `Collection<T>`, `CollectionBase`.
3. Z listy kolekcji z punktu 2 wybierz trzy kolekcje i wykonaj dla nich następujące operacje:
  - a. stwórz odpowiednią kolekcję (jedną z elementami typu `int`, drugą typu `string`, trzecią typu `Osoba`)
  - b. zastanów się jaką przestrzeń nazw potrzebujesz do stworzenia kolekcji
  - c. spróbuj dodać na stworzone kolekcje element innego typu - czy wszystko jest możliwe?
  - d. różnymi operacjami spraw, żeby każda z kolekcji miała co najmniej 5 elementów
  - e. za pomocą `foreach` wyświetl kolejne elementy kolekcji na konsoli (dla `Osób` ma być ich wiek i imię)
  - f. spróbuj z kolekcji wydobyć pierwszy, trzeci i ostatni element - czy wszystko jest możliwe?
  - g. spróbuj posortować elementy w stworzonych kolekcjach, pomyśl czy to jest możliwe, czy musisz coś uzupełnić w kodzie, ew może nic nie musisz robić bo elementy po dodaniu są już sortowane?
  - h. wyświetl na konsoli liczbę elementów w danej kolekcji oraz pojemność kolekcji (`capacity`), skąd są różnice?
  - i. spróbuj usunąć czwarty element w każdej kolekcji, czy to jest możliwe?
  - j. odwróć kolejność elementów w danej kolekcji

#### Zadania z treścią:

1. Stwórz symulator gry w oczko (talía 24 kart). Dobierz odpowiednią kolekcję do przechowywania kart obu graczy. Podpowiedź: jeśli będzie łatwiej stwórz własny typ `enum` na przechowywanie kart.
2. Za pomocą słownika zrób tabliczkę mnożenia dla liczb zapisywanych w systemie rzymskim (key ustaw na liczby powiedzmy od 1 do 3, a value odpowiedni `string` - liczba w zapisie rzymskim).
3. Przeanalizuj program dostępny tu <https://github.com/Piotrek85/SzybkoscKolekcjiTest>, który służy do porównania szybkości dostępu do elementów na kolekcje. Zmodyfikuj program aby potestować szybkości innych kolekcji.