

Задание 6.1. Англо-русский словарь (коллекции, строки)

Реализуйте программу с функциональностью простого англо-русского словаря.

Функциональность программы:

1. Пользователь должен иметь возможность добавляет пары слов в словарь (одно из которых - русское, а второе – английское).
При добавлении слова и перевода к нему в словарь автоматически должен добавляться и перевод в обратную сторону.
Программа должна корректно добавлять перевод для уже существующего слова в словаре. Также программа должна корректно работать при добавлении уже существующего перевода для слова (не добавлять дубликаты).
2. Пользователь должен иметь возможность запрашивать перевод слова (русского или английского – программа должна определить самостоятельно), и получать возможные варианты перевода.
3. При этом одному русскому слову может соответствовать несколько английских слов, и наоборот, одному английскому – несколько русских.
4. Желательно, некоторое первоначальное количество пар слов завести в программе.

Требования по реализации:

5. Функциональность двустороннего перевода должна быть реализована в виде отдельного класса, а программа должна предоставлять к этому классу внешний пользовательский интерфейс. Словарь не должен напрямую взаимодействовать с пользователем.
6. Словарь должен быть не чувствительным к регистру. Т.е. hello и Hello – это одинаковые слова.

Примечания:

Для организации двустороннего перевода можно использовать стандартные коллекции .NET. А именно – класс `Dictionary<string, List<string>>` или `Dictionary<string, HashSet<string>>` для организации перевода в одну сторону (т.е. по слову получать список соответствующих ему слов).

Естественно, всякие пробелы и т.п. вокруг пользовательского ввода нужно устранять; для этого можно воспользоваться методом `string Trim()` класса `string`.

```
string s = " что-то ";  
string ns = s.Trim();
```

А также необходимо игнорировать регистр слов, например, преобразовывая слова к строчным буквам

```
string smalls = s.ToLower();
```

Фактически словарь должен содержать 2-3 публичных метода:

```
void Add(string word, string translation);
```

```
и/или void Add(string word, List<string> translations);
```

```
List<string> GetTranslations(string word);
```

Задание 6.2. (по желанию, необязательное) Бесконечная последовательность простых чисел (последовательности, `IEnumerable<T>`)

Реализуйте класс, представляющий из себя “бесконечную” последовательность всех простых чисел, начиная с 1.

Используя этот класс, распечатайте на экране первые N элементов последовательности простых чисел, где N – произвольное целое число вводимое пользователем.