Sovelto

Viikko 2, Generics

ACADEMY
A PART OF ACADEMIC WORK

Laskentaa

- Tee geneerinen MultiSet-luokka. MultiSet on avainnettu dynaaminen kokoelma, johon voi tallettaa samalla avaimella useita arvoja. Dictionaryyn voi tallettaa vain yhden arvon yhdellä avaimella ja tässä korjataan kyseinen rajoite/puute.
- Lisää projektiin uusi geneerinen luokka nimeltä MultiSet
- Tee MultiSet-luokkaan kokoelma, jolla talletat TItem-tyyppiset arvot. Käytetään Dictionary-luokkaa, joka sisältää List-olioita. List-oliot sisältää talletettavat arvot. Alusta kenttä konstruktorissa. List-kokoelmaan siis talletetaan kaikki samalla avaimella talletettavat arvot. Esimerkkikoodia:

```
public class MultiSet<TKey, TItem> {. . .
private Dictionary<TKey, List<TItem>> _multiset;
```

Luokan metodit

```
public int Add(TKey key, TItem item)
// palauttaa ensimmäisen tällä avaimella talletetun arvon, indexer
public TItem this[TKey key]
// palauttaa N:nnen arvon avaimesta
public TItem this[TKey key, int index]
// poistaa avaimen kaikki arvot
public void Remove(TKey key)
// poistaa avaimen tietyn arvon
public int Remove(TKey key, int index)
// palauttaa avaimella talletettujen arvojen lukumäärän
// -1 jos avaimella ei ole arvoja
public int ItemCount(TKey key)
```

Jatkuu

- Heitä poikkeus jos indekserissä tai Remove-metodissa index-parametri on virheellinen.
- Indexerien toteutuksessa huomaa, että koodin pitää palauttaa Null -arvon jos avaimella ei löydy mitään. Arvotyypeillä ei voi olla null-arvoa joten käytä Default-määrittelyä tarpeen mukaan.
  Mieti miten erotat arvotyyppejä käytettäessä onko avaimella löytynyt arvo vai ei. Tässä tilanteessa Nullable-tyyppien käyttö olisi muuten toimiva, mutta valitettavasti ei oikein käytännössä onnistu. Mieti miksi!
- Toteuta luokkaan property Count (readonly), joka palauttaa kaikkien talletettujen arvojen lukumäärän, ei pelkästään avainten lukumäärää.

IEnumerable<T> ja Yield

- Lisätään luokkaan toiminnot jotta sen sisältämät arvot on käytävissä läpi foreach-silmukalla.
- Lisää luokan määrittelyyn IEnumerable-rajapinta ja toteuta se. Käytä varattua sanaa yield.

```
public class MultiSet<TKey, TItem> : IEnumerable<TItem>
//...
#region IEnumerable<TItem> Members
public IEnumerator<TItem> GetEnumerator() {
    foreach (List<TItem> list in _multiset.Values) {
        foreach (TItem item in list) {
            yield return item;
        }
    }
}
#endregion

#region IEnumerable Members
System.Collections.IEnumerator System.Collections.IEnumerable.GetEnumerator() {
    return GetEnumerator();
}
#endregion
```