

Generic List Exercise

Skapa en generisk klass med exempelvis namnet `Collection<T>`. Denna klass ska fungera på samma sätt som klassen `List<T>` men endast implementera följande (begränsande) interface.

```
internal interface ICollection<T>
{
    T this[int index] { get; set; }

    int Count { get; }

    void Add(T item);
    bool Remove(T item);
    void RemoveAt(int index);

void Clear(); // Extra exercise!
bool Contains(T item); // Extra exercise!
T Find(Predicate<T> match); // Extra exercise!
}
```

Säkerställ att din kod hela tiden kompilerar! Implementera och testa en metod (klassmedlem) i taget!

Klassen lagrar kollektionens element i en privat array av `T (T[])`. Klassen ska ha två konstruktorer. En konstruktor ska vara parameterlös och initiera array:ens längd till fyra element. Den andra konstruktorn ska ta en parameter som anger kollektionens initiala kapacitet, d.v.s. det initiala antalet element i array:en. Array:ens kapacitet (antal element) får dock aldrig understiga fyra element.

Om en instans av klassen - exempelvis `myCollection` - adderar ett element - `myCollection.Add(element)` - som inte kommer att få plats i den privata array:en ska array:en expanderas till dubbla kapaciteten. D.v.s. om array:ens nuvarande längd är fyra element ska den expanderas till åtta element. Därefter läggs det nya elementet till i array:en.

Om användaren av klassen tar bort ett element och antalet element i array:en därefter är lika med eller mindre än hälften av array:ens totala kapacitet ska array:ens kapacitet minskas till hälften av dess nuvarande kapacitet. Array:ens kapacitet (antal element) får dock aldrig understiga fyra element.

Tips: Undersök vilka metoder i klassen `Array` som kan hjälpa dig att expandera eller minska den privata array:ens storlek.

Varje publik medlem (se interfacet) ska XML-dokumenteras på samma sätt som motsvarande publika medlemmar i klassen `List<T>`.

Säkerställ att din kod hela tiden kompilerar! Implementera och testa en metod i taget! Testa med minst två olika datatyper i din kollection, exempelvis `Collection<Person>` och `Collection<int>`.

Extra: Utöka det publika interfacet med metoderna `"void Clear()"`, `"bool Contains(T item)"`, samt `"T Find(Predicate<T> match)"`.

Extra: Implementera stöd för `"foreach"` (vilket automatiskt implementerar stöd för LINQ) på klassen `Collection<T>`. Vad får du för kompileringsfel om du försöker tillämpa `"foreach"` och/eller LINQ metoder på en instans av `Collection<T>`? Googla kompileringsfelen!