[0841] IEnumerable

Classes that implement IEnumerable can be used in a "foreach/in" loop. IEnumrable is part of the System.Collections namespace which we will do more with soon. (It turns out that "foreach/in" is also tightly associated with Collections.)

Here's an example:

```
public class Deck : IEnumerable
{
    private Card[] cards;
    private static Random rng = new Random();       // static helps prevent duplicate rng's

    public Deck()
    {
        Array suits = Enum.GetValues(typeof(Suit));
        Array ranks = Enum.GetValues(typeof(Rank));

        int size = suits.Length * ranks.Length;
        cards = new Card[size];

        int i = 0;
        foreach (Suit suit in suits) {
            foreach (Rank rank in ranks) {
                Card card = new Card(suit, rank);
                cards[i++] = card;
            }
        }
    }

    public IEnumerator GetEnumerator()
    {
        foreach (Card card in cards) {
            yield return card;
        }
    }

    // More Deck methods go here.

}
```

The "yield return" statement does most of the work here. It returns the current card and then "pauses" the GetEnumerator() method so that the next time it is called, it resumes execution right in the middle of the loop where "yield return" was called. In other words, each time GetEnumerator() is called, it returns one card and the next time it is called it returns the next card.