Listing 4-7 Using Iterators to Implement Enumeration

```
using System;
using System.Collections.Generic;
public class GenStack<T>: IEnumerable<T>
{
   // Use generics type parameter to specify array type
   private T[] stackCollection;
   private int count = 0;
   // Constructor
   public GenStack(int size)
   {
      stackCollection = new T[size];
   }
   // (1) Iterator
   public IEnumerator<T> GetEnumerator()
   {
      for (int i = 0; i < count; i++)
      {
         yield return stackCollection[i];
      }
   }
   // (2) Property to return the collection in reverse order
   public IEnumerable<T> Reverse
   {
      get
      {
         for (int i = count - 1; i >= 0; i--)
         {
            yield return stackCollection[i];
         }
      }
   }
   public void Add(T item)
   {
      stackCollection[count] = item;
      count += 1;
   }
   // other class methods go here ...
```