

# Colecciones

7

Programación II y Laboratorio de Computación II

Edición 2018

# Colecciones

- Existen dos formas de agrupar objetos: mediante la creación de matrices de objetos y mediante la creación de colecciones de objetos.
- Las matrices son muy útiles para crear y trabajar con un número fijo de objetos fuertemente tipados.
- Las colecciones proporcionan un método más flexible para trabajar con grupos de objetos.
- A diferencia de las matrices, el grupo de objetos con el que trabaja puede aumentar y reducirse dinámicamente a medida que cambian las necesidades de la aplicación.

# Colecciones

- Una colección es una clase, de modo que antes de poder agregar elementos a una nueva colección, debe declararla.
- Una colección genérica cumple la seguridad de tipos para que ningún otro tipo de datos se pueda agregar a ella.
- Cuando se recupera un elemento de una colección genérica, no tiene que determinar su tipo de datos ni convertirlo.

# Colecciones

```
List<string> palabras;  
palabras = new List<string>();  
// Agrego  
palabras.Add("Hola");  
palabras.Add("Hello");  
palabras.Add("Olá");  
palabras.Add("Ciao");  
// Quito  
palabras.Remove("Hello");  
// Recorro y muestro  
foreach (string palabra in palabras)  
{  
    Console.WriteLine(palabra);  
}
```

# Colecciones

```
List<string> palabras;  
palabras = new List<string>() { "Chau", "Bye", "Ciao" };  
// Recorro y muestro  
for (int i = 0; i < palabras.Count; i++)  
{  
    Console.WriteLine(palabras[i]);  
}
```

# Colecciones Genéricas

- Se puede crear una colección genérica utilizando una de las clases en el espacio de nombres `System.Collections.Generic`.
- Una colección genérica es útil cuando todos los elementos de la colección tienen el mismo tipo de datos.
- Una colección genérica cumple el tipado fuerte al permitir agregar sólo el tipo de datos deseado.

# Colecciones Genéricas

- **Dictionary:**
  - Representa una colección de pares de clave y valor que se organizan por claves.
- **List:**
  - Representa una lista de objetos que pueden ser obtenidos mediante un índice.
  - Proporciona métodos para buscar, ordenar y modificar listas.
- **Queue**
  - Representa una colección de objetos con el orden primero en entrar, primero en salir (FIFO).

# Colecciones Genéricas

- **SortedList:**
  - Representa una colección de pares de clave y valor que se ordenan por claves según la implementación de la interfaz `IComparer<T>` asociada.
- **Stack:**
  - Representa una colección de objetos con el orden último en entrar, primero en salir (LIFO).

# Colecciones No Genéricas

- Son las incluidas en el espacio de nombres `System.Collections`.
- Estas no almacenan los elementos como objetos de un tipo específico, sino como objetos de tipo `Object`.
- Siempre que sea posible, se deberían utilizar las colecciones genéricas de otros tipos en lugar de estas.

# Colecciones No Genéricas

- **ArrayList**
  - Representa una matriz de objetos cuyo tamaño aumenta dinámicamente según sea necesario.
- **Hashtable**
  - Representa una colección de pares de clave y valor que se organizan por código hash de la clave.
- **Queue**
  - Representa una colección de objetos con el orden primero en entrar, primero en salir (FIFO).
- **Stack**
  - Representa una colección de objetos con el orden último en entrar, primero en salir (LIFO).

# Colecciones No Genéricas

```
//using System.Collections;  
// Colas  
Queue queue = new Queue();  
queue.Enqueue("A");  
queue.Enqueue("B");  
string s = queue.Dequeue().ToString();  
// Pilas  
Stack stack = new Stack();  
stack.Push(1);  
stack.Push(2);  
int i = (int)stack.Pop();
```

# Colecciones Concurrentes

- Son las colecciones en el espacio de nombres `System.Collections.Concurrent`.
- Proporcionan operaciones eficaces y seguras para subprocesos con el fin de obtener acceso a los elementos de colección desde varios subprocesos (hilos).
- Deben utilizarse en lugar de sus equivalentes en los espacios de nombres `System.Collections.Generic` cuando varios subprocesos tienen acceso a la colección simultáneamente.
- Son `BlockingCollection<T>`, `ConcurrentDictionary<TKey, TValue>`, `ConcurrentQueue<T>` y `ConcurrentStack<T>`.

# General

- No todas las colecciones tienen las mismas propiedades.
- Por ejemplo:
  - Las colas y pilas no pueden ordenarse en si mismas.
  - Las colas y pilas no se pueden serializar (pasar a archivos).