

Strings

Strings can be created or accessed like a char-array

Multiple strings can be connected with a `String`

Generics - not specific to a particular data type
generic type is declared by specifying a type parameter
in angle brackets after a type name: `Type Name <T>`

Generic class

- increases reusability
- can be a base class to other generic or non-generic class
- can be derived from — 11 —

A Method declared with a type parameter for its
own return type or parameters is a generic method

Generics are type safe, and have performance
advantage because they remove the possibility of
boxing/unboxing

Generic Collections

`List <T>` - contains elements of a specified type

`Dictionary <TKey, TValue>` contains key-value pairs

`SortedList <TKey, TValue>` — 11 — 11 — . Adds

elements in ascending order of key
`Queue <T>` - stores values FIFO style. Enqueue() method for
adding values and dequeue() to retrieve values.

Stack<T> - stores values ~~FIFO~~ ^{LIFO} style. Push() to add values and Pop() and Peek() to retrieve

HashSet<T> - list elements, eliminating duplicates

Non-Generic Collections

ArrayList - stores objects of any kind but there is no need to specify the ~~number~~ size of the list as it grows automatically

Hashtable - stores key-value pairs. retrieves values by comparing hash values of the keys

BitArray - manages an array of bit values (1 or 0) which are represented as a boolean

C# also includes non-generic versions of Queue, Stack and SortedList.

Tuple<T> - data structure that contains a sequence of elements of different datatypes. It's used when we want to hold an object with properties but don't want to create a separate type for it. ~~Then~~ the element can be accessed via Item<elementNumber> or Rest if it's the last item. If we want to store more than 8 elements in a tuple, we can do that by nesting another tuple object in the 8th element. We can access it via Rest.Item1.Item<elementNumber>. In theory we can nest tuple object anywhere in the sequence but we won't be able to access its elements.

Tuple is useful when:

- We want to return multiple values from a method
- We want to pass multiple values to a method with a single parameter
- We want to hold a database temporarily without creating separate ~~at~~ class