# **StringBuilder in C#**

Об'єкт **String** є незмінним, тобто його не можна змінити після створення. Кожен раз, коли ви використовуєте будь-який з методів класу System.String, ви створюєте новий об'єкт рядка в пам'яті. Наприклад, рядок "GeeksForGeeks" займає пам'ять в купі, тепер, змінивши початковий рядок "GeeksForGeeks" на "GFG", ви створите новий об'єкт рядка в купі пам'яті замість того, щоб модифікувати початковий рядок в тій же області пам'яті. У ситуаціях, коли потрібно виконати багаторазову модифікацію рядка, нам знадобиться клас StringBuilder. Щоб уникнути заміни рядка, додавання, видалення або вставки нових рядків у початковий рядок, у мові C# введено поняттяStringBuilder**.** StringBuilder є динамічним об'єктом. Він не створює новий об'єкт в пам'яті, а динамічно розширює необхідну пам'ять для розміщення зміненого або нового рядка.

**Оголошення та ініціалізація StringBuilder**

**StringBuilder** можна оголошувати та ініціалізувати так само, як і клас,

StringBuilder s = new StringBuilder();

//or

StringBuilder s = new StringBuilder("GeeksforGeeks");

***"s"*** - це об'єкт класу **StringBuilder**. Також ми можемо передати рядкове значення (тут "GeeksforGeeks") як аргумент конструктору **StringBuilder.**

**Визначення ємності StringBuilder**

Хоча StringBuilder є динамічним об'єктом, який дозволяє розширювати кількість символів у рядку, який він інкапсулює, ви можете вказати значення максимальної кількості символів, яку він може вмістити. Це значення називається ємністю об'єкта StringBuilder.

StringBuilder s = new StringBuilder(20);

or

StringBuilder s = new StringBuilder("GeeksForGeeks", 20);

Ось тут,

* У 1-му операторі ми передаємо ціле значення як аргумент конструктору. Це максимальна кількість символів, яку може вмістити рядок.
* У 2-му операторі ми передаємо в якості аргументу конструктору значення рядка з цілим числом (це максимальна кількість символів, яку може вмістити рядок).

**Important Methods of StringBuilder Class:**

* Append(string value)
* AppendFormat()
* Insert(int index, string value)
* Remove(int start, int length)
* Replace(old\_val, new|\_val)

**Метод StringBuilder.Append(рядкове значення - string value)**

Метод **Append** можна використовувати для додавання або вставки рядкового значення об'єкта в кінець рядка, представленого поточним об'єктом StringBuilder. Метод **AppendLine()** також відноситься до цього методу. Цей метод додає рядок з новим рядком в кінці.

using System;

using System.Text;

class GFG {

// Main Method

public static void Main()

{

// "20" is capacity

StringBuilder s = new StringBuilder("HELLO ", 20);

s.Append("GFG");

// after printing "GEEKS"

// a new line append

s.AppendLine("GEEKS");

s.Append("GeeksForGeeks");

Console.WriteLine(s);

}

}

**Output:**

HELLO GFGGEEKS

GeeksForGeeks

**StringBuilder.AppendFormat()**

Цей метод використовується для форматування вхідного рядка у вказаний формат, а потім його додавання. Цей метод також додає текст в кінець об'єкту **StringBuilder**.

using System;

using System.Text;

class GFG {

// Main Method

public static void Main()

{

StringBuilder s = new StringBuilder("Your total amount is ");

// using the method

s.AppendFormat("{0:C} ", 50);

Console.WriteLine(s);

}

}

Your total amount is Â¤50.00