

# StringBuffer

ver 1.0

В езика C начинът, по който се реализират низове е масив от символи, който завършва с 0. Но така програмите, които работят с низове, се организират около кода, а не около данните (самите низове).

## Задача:

Да се реализира клас **StringBuffer**. Класът трябва да представя низ, на който могат да бъдат променяни стойността и дължината. Капацитетът на низа се променя така, че във всеки момент **StringBuffer** има капацитет по-голям от действителния низ, който представя (операцията **Append** може да се осъществява много ефективно).

| Конструктори                              |
|---|
| StringBuffer()                            |
| StringBuffer(char * str)                  |
| StringBuffer(int capacity)                |
| StringBuffer(char * str, int capacity)    |
| StringBuffer(StringBuffer & stringBuffer) |
| Деструктор                                |

| Методи  |
|---|
| int Length()  |
| int Capacity()  |
| int IndexOf(char character) * // -1, ако не е намерен           |
| char CharAt(int position) * // ако е дадена невалидна позиция 0 |
| bool Equals(StringBuffer & comparedString)                      |
| int CompareTo(StringBuffer & comparedString) // както в C       |
| void Append(char * stringToAppend)                              |
| void Append(StringBuffer & stringToAppend)                      |
| void Append(int i)// добавя числото i като поредица от символи  |
| void Print() // Този метод ще бъде заменен в последствие        |
| StringBuffer& Substring(int start, int end) *                   |

|   |
|---|
| <b>Методи</b>                               |
| //от start включително до символа преди end |

\* Позицията започва от 0.

За да бъде реалистичен този клас има нужда от още методи. Например в JAVA има още **replace**, **reverse**, **insert** и т.н.

Класът, който реализирате, **трябва да се подчинява на този интерфейс**. Вие сами трябва да прецените всичките **private** данни и методи, които ви трябва. Помислете преди да започнете да пишете!

**1.1** Като използвате написания клас, напишете програма **reverse.cpp**, която обръща низ въведен от потребителя.

**1.2** Като използвате написания клас, напишете програма, която заменя във **n** въведени от потребителя низа (**n** също се въвежда) тага **<br/>** с **\n** и ги извежда на екрана.

**Условието ще бъде допълвано! Пазете това, което сте писали.**