

В **задача 1** написахме клас **String**, който ни дава средства за лесно създаване и употреба на динамични символни низове. Сега, ще използваме вече написаното, за да предоставим още повече средства за лесна употреба на този клас.

Разширете класа **String**, като добавите следните методи:

1. **String substring(int beg, int end) const** - Връща **String**, който съдържа само символите от индекс **beg** до индекс **end** (**end** се намира извън масива).
Ако **beg < 0**, за начален индекс се взима **0**;
Ако **beg >= length**, за начален индекс се взима **length - 1**;
Ако **end > length**, за краен индекс се взима **length**;
Ако **end <= beg**, за краен индекс се взима **beg + 1**;
2. **String substring(int beg) const** - Връща **String**, който съдържа само символите, от индекс **beg** до края. Важат горните ограничения за **beg**;
3. **String toLowerCase() const** - Връща **String**, в който всички главни латински букви, са заменени с малки;
4. **String toUpperCase() const** - Връща **String**, в който всички малки латински букви, са заменени с главни;
5. **String trim() const** - Връща **String**, от който са премахнати всички излишни интервали, т.е. всички интервали в началото и края му, както и всички интервали по-дълги от един символ;
6. **int lastIndexOf(char symbol) const** - Връща индексът на последното място, на което се среща подаденият символ. Ако не се среща - връщаме **-1**.
7. **int firstIndexOf(char symbol) const** - Връща индексът на първото място, на което се среща подаденият символ. Ако не се среща - връщаме **-1**.

Всички тези методи **НЕ ПРОМЕНЯТ** обектът, върху който се извикват, а връщат **НОВИ** инстанции на клас **String** (с изключение на последните два).

Подказка: Напишете **private** конструктор **String(char* data, int length)**, който наготово получава данните и дължината им. Използва ги без да копира динамичната памет, т.е. отговорността за нея се прехвърля на инстанцията създадена с този конструктор. Деструктора ще освободи тази памет.

Каква е ползата от това? Тъй като публичните конструктори не ни позволяват, просто да създадем инстанция на класа, без тя да копира каквато и да е памет, няма как удобно да дадем на инстанцията предварително създаден динамичен масив, без да го копираме.

Защо е private? Този конструктор не прави никакви проверки и валидация на входните данни, нито прави собствено копие на подадените стойности. Неправилната му употреба може да остави инстанцията в невалидно състояние и дори да доведе до грешка. Той е подходящ, само и единствено, за извикване в методи на класа **String**.