

Нуждаем се от средство за работа със динамично заделени символни низове, но определено не желаем да се грижим за паметта. Ще напишем клас `String`, който да обединява в себе си всички удобства и бързодействие на символните низове, лесна работа с динамична памет, както и лесен начин за обработване на низовете.

Класът трябва да има следните полета:

1. `char* data` - Указател към динамичната памет, която ще се задели
2. `int length` - Дължината на динамично заделеният низ

Както е добре познато, символните низове завършват със символ `'\0'`, така че ще запазим това им свойство. Това означава, че `length` ще бъде броят на всички символи в низа, с изключение на `'\0'`.

Освен полетата, ще трябва да има, разбира се, и подходящо написана голяма четворка:

1. `String(const char* = "")` - Конструктор
2. `String(const String&)` - Копиращ конструктор
3. `~String()` - Деструктор
4. `String& operator = (const String&)` - Оператор за присвояване

Но как ще осигурим всичките яки неща, които класът ще предлага, ако не добавим и следните методи:

1. `String& append(const char*)` - Залепя подаденият ни низ за текущия, като връща новополученият обект.
2. `String& operator += (const char*)` - Същото.
3. `char& operator [] (int)` - Връща ни символът от динамичния низ, който се намира на желаната позиция, като ни позволява директно да го променяме. Напишете оператора и в константната му форма.
4. `operator const char* () const` - Връща самият динамичен масив, но не ни разрешава да го променяме.
5. `int getLength() const` - подразбира се
6. `String operator + (const char*) const` - Залепя текущия низ със подадения, като резултата записва в нов обект от тип `String`, който се връща от метода. Низовете участващи в операцията не се променят!