Задание по C++ №5, осень 2018 года

Создадим свой простой аналог стандартного класса string для удобной работы со строками.

Начнем мы с написания конструктора. В этой задаче вам требуется реализовать конструктор, который принимает на вход C-style строку, вычисляет ее размер (без учета завершающего 0 символа) и сохраняет его в поле size, кроме того, конструктор должен аллоцировать память достаточную для хранения копии переданной строки (вместе с завершающим 0 символом), копирует переданную строку в выделенную память и сохраняет указатель на начало этой области памяти в поле str. Т.е. в результате работы конструктора в поле str должен оказаться указатель на C-строку с копией исходной строки, а в поле size — длина строки без учета завершающего нулевого символа.

Конструкторов у структуры может быть несколько. Для строки может оказаться полезным заполняющий конструктор (например, чтобы создать строку пробелов). Заполняющий конструктор принимает число и символ, и создает строку с заданным количеством повторений переданного символа. Условия налагаемые на реализацию конструктора те же самые (в поле size размер без учета завершающего 0 символа, в поле str C-style строка, т.е. с завершающим 0 символом). Кроме конструктора в этой задаче вам нужно также реализовать и деструктор, который освободит выделенную память.

Для работы со строками можно придумать множество полезных методов (подумайте, какие методы пригодились бы вам, и чего вам не хватает для их реализации). Примером такого метода может послужить метод append — он добавляет копию строки-аргумента в конец текущей строки (т.е. в конец строки, у которой он был вызван).

```
String s1("Hello,");
String s2(" world!");

s1.append(s2); // теперь s1 хранит "Hello, world!"
```

Ваша задача реализовать метод append.

При выполнении задания будьте аккуратны при работе с памятью — при вызове метода не должно возникать утечек памяти. Кроме того, не забудьте, что size хранит размер без учета завершающего 0 символа.

Кроме того, ваша реализация должна корректно работать в следующем тесте:

```
String s("Hello");
s.append(s); // теперь s хранит "HelloHello"
```

Добавьте в класс String реализацию конструктора копирования. Инвариант класса остается тем же (в size хранится размер строки без завершающего 0 символа, str указывает на C-style строку, т. е. с завершающим нулевым символом).

Завершите класс String, добавив к нему оператор присваивания. Будьте аккуратны при работе с памятью. Инвариант класса остается тем же (в size хранится размер строки без завершающего 0 символа, str указывает на C-style строку, т. е. с завершающим нулевым символом).