

**Тема:** инкапсуляция, конструкторы и перегрузка операторов

**Вариант:** 1.1.2

**Задача:** Реализовать класс для работы со строками задаваемой длины.

При этом в классе необходимо реализовать следующую функциональность:

1. Инициализация строки заданного размера с помощью:
  - a. Конструктора по умолчанию для инициализации строки нулевой длины
  - b. Конструктора, принимающего символ `char`, для инициализации новой строки длины 1, содержащей один этот символ
  - c. Конструктора, принимающего zero-terminated строку `const char *` для инициализации новой строки соответствующими символами
2. Перегружены операторы для:
  - a. Конкатенации строк. Реализовать два варианта через перегрузку оператора `+` и `+=`
  - b. Взятия подстроки. Реализовать через перегрузку оператора `()`:

```
String str("Hello world!");  
String str2 = str(6, 11);
```

После этого в `str2` должна быть строка `"world!"`.

При попытке взять подстроку в некорректном диапазоне программа должна заканчиваться с соответствующим сообщением об ошибке.

- с. Дубликации строки указанное количество раз. Для реализации перегрузить оператор \*:

```
String str("Hello! ");  
String str2 = str*3; // "Hello! Hello! Hello! "
```

- d. Взятия элемента по индексу. Для реализации необходимо перегрузить оператор []

При попытке получить доступ к элементу по некорректному индексу программа должна заканчиваться с соответствующим сообщением об ошибке.

- e. Удаление всех вхождений подстроки из строки. Для реализации перегрузить оператор -.

```
String str("hello world hello");  
String str2("hello");  
String3 = str - str2; // " world "
```

- 3. Получение длины строки со сложностью в худшем случае  $O(1)$
- 4. Ввод и вывод строки в заданный файл или на экран. При желании для этого пункта также можно перегружать операторы (на выбор студента)
- 5. Корректное управление динамической памятью во внутренней структуре класса. В классе не должно быть утечек памяти, некорректных указателей и т. д.

В качестве демонстрационного примера написать программу, считывающую из файла три строки *str1*, *str2*, *str3* и записывающую в выходной файл строку, полученную из *str1* заменой всех вхождений в нее строки *str2* на строку *str3*.

**Внимание:** при решении данной задачи запрещается пользоваться классами `std::string`, `std::wstring` и т.д. из стандартной библиотеки.

### Входные данные:

Во входном файле находятся строки *str1*, *str2*, *str3*, разделенные переносом строки.

### Выходные данные:

В выходной файл записать получившуюся строку.

### Пример входных и выходных данных:

input.txt	output.txt
Hello, World, Hello! Hello Goodbye	Goodbye, World, Goodbye!

---

### Дополнительное задание:

При решении основной задачи дополнительно поддерживать возможность работы с широкими строками.

Каждый символ в такой строке может занимать от 1 до 4 байт. Символы кодируются согласно UTF-8:

<https://ru.wikipedia.org/wiki/UTF-8>

В классе нужно добавить:

1. Новый параметр для конструкторов, обозначающий, должна ли строка трактоваться, как широкая или нет
2. Корректную обработку элементов различного размера при доступе к элементу через оператор [].
3. Корректное вычисление длины широкой строки. Под длиной понимается количество разных символов, а не байт в строке.