Темы

* Классы с динамическими полями;
* Операторы преобразования типа и другие операции;
* Конструктор копирования;
* Операции с памятью;

Операторы преобразования типа в C++ (например, operator const char\*() const) представляют собой специальный вид перегруженных операторов, которые позволяют преобразовывать объекты пользовательского класса в другие типы данных.

Главное отличие заключается в их синтаксисе: они не имеют явного имени. Применение: операторы преобразования полезны, когда необходимо обеспечить взаимодействие объектов пользовательского класса с функциями, ожидающими другие типы данных.

Квалификатор const играет критическую роль в обеспечении безопасности кода и улучшении читаемости. Он указывает на то, что переменная, аргумент функции или метод класса не может быть изменена.

Применение:

const переменные: объявляют переменную, значение которой не может быть изменено после инициализации.

const аргументы функций: гарантируют, что функция не изменит значение аргумента, переданного по ссылке или указателю.

const методы класса: гарантируют, что метод не изменяет состояние объекта, для которого он вызван. const методы могут быть вызваны для const объектов.

Логика оператора += для класса Str:

Оператор += реализован для конкатенации C-style строки (const char\*) к объекту Str. Его логика:

1. Вычисление новой длины: определяется длина добавляемой строки (strlen(sz)) и вычисляется новая длина строки (m\_nLength + strlen(sz)).
2. Выделение новой памяти: выделяется память для новой строки, учитывая длину исходной строки, добавляемой строки и символ конца строки (‘\0’).
3. Копирование данных: содержимое существующей строки копируется в выделенную память (strcpy(newText, m\_pszText)). Затем, содержимое добавляемой строки копируется в конец новой строки (strcat(newText, sz)).
4. Освобождение старой памяти: освобождается память, выделенная для исходной строки (delete[] m\_pszText;).
5. Обновление указателя и длины: указатель m\_pszText переназначается на новую память, и обновляется значение m\_nLength.

Такой подход обеспечивает безопасное и эффективное добавление строк, при этом гарантируя, что объект Str содержит корректные данные после выполнения операции. Использование const char\* в качестве параметра позволяет избежать лишних копирований, делая код более производительным.