

Отчет

1. Операторы преобразования типа в C++ позволяют неявно преобразовывать объекты одного типа в объекты другого типа. Они объявляются с помощью ключевого слова "operator" и типа, в который будет выполняться преобразование. Например, в классе Str мы объявили оператор преобразования типа "operator const char*() const", который позволяет неявно преобразовывать объекты класса Str в тип "const char*".

2. Квалификатор const в C++ используется для объявления константных объектов, ссылок и функций. Он гарантирует, что объект не будет изменен после его создания или что функция не будет изменять объект, к которому она обращается. В классе Str мы использовали квалификатор const в конструкторе копирования, операторе присваивания и операторе преобразования типа, чтобы гарантировать, что они не будут изменять объект, к которому они обращаются.

3. Оператор "+=" в классе Str реализует добавление строки, переданной в качестве параметра, к текущему объекту. При этом он освобождает память, выделенную под текущую строку, выделяет новую память для новой строки и копирует содержимое обеих строк в новую строку. Оператор возвращает ссылку на текущий объект, чтобы можно было выполнять цепочки операций.

В ходе выполнения задания разработал оператор "+", который получает новый объект, равный сумме двух строк. Он реализован с использованием оператора "+=" и возвращает новый объект, а не ссылку на текущий объект, так как исходный объект не изменяется.

Кроме того, изучил особенности работы с динамической памятью в C++ и применили их при реализации класса Str. Использовал оператор "new" для выделения памяти под строку и оператор "delete" для освобождения памяти.

В результате выполнения задания получил класс Str, который позволяет работать со строками, как с объектами, и реализует необходимые операции для работы со строками, такие как добавление строк, копирование строк и преобразование строк в другие типы.