

Отчет по заданию “САОД. Задание 3”

1. Создание проекта RefStrings:
 - Я создал новый консольный проект под названием 'RefStrings' для выполнения задания.
2. Добавление классов Str и _str:
 - Добавил в проект два inline класса: 'Str' и '_str', для работы с подсчетом ссылок, разместив их полностью в заголовочных файлах.
3. Перенос кода из лекции:
 - Скопировал соответствующий код из лекции в '_str.h', добавив строку '#include <string.h>' сразу после '#pragma once'.
 - Перенес код с конструктором и деструктором из лекции в 'Str.h'. Добавил '#include "_str.h"' в начало файла 'Str.h' для решения проблемы зависимости.
4. Подключение класса Str к основному модулю:
 - Включил 'Str.h' в основной файл 'main.cpp' и создал объект класса 'Str' в функции 'main'. Проверил, что проект компилируется и собирается без ошибок.
5. Реализация конструктора копирования и операторов:
 - Перенес в класс 'Str' конструктор копирования и все операторы, реализованные в лекции. Это включает оператор присваивания и оператор '+='.
6. Добавление оператора преобразования к типу const char*:
 - Реализовал оператор преобразования 'Str' к типу 'const char*', что позволяет выводить объекты класса 'Str' на консоль.
7. Проверка работы оператора +=:
 - Убедился, что оператор '+=' работает правильно, объединив две строки и выведя результат на консоль.
8. Добавление оператора [] для чтения символов строки:
 - Реализовал оператор 'char operator[](int i) const', который позволяет получать значение i-того символа строки. Добавил проверку границ строки

и генерацию исключения `std::out_of_range`, если индекс выходит за пределы строки.

- Убедился, что теперь можно читать каждый отдельный символ строки и для индексов за пределами строки возникает исключение.

9. Доработка оператора `[]` для изменения символов строки:

- Реализовал оператор `char& operator[](int i)`, который позволяет изменять значение *i*-того символа строки. Добавил проверку границ строки и реализацию разделения данных при необходимости.
- Убедился, что при наличии двух копий одной строки, при замене символа в одной из них, вторая остается неизменной.
- Проверил, что при замене нескольких символов одной строки не происходит многократного разделения данных.

Вывод программы:

Программа выводит следующие строки и символы, демонстрируя корректную работу всех операторов и функций:

Hello World

Hello

Hello World

e

Hallo World

Hello

Этот вывод подтверждает, что:

1. Строки `s1` и `s2` корректно создаются и копируются.
2. Конкатенация строк работает правильно.
3. Операторы `[]` для чтения и изменения символов строки работают корректно и вызывают исключение при выходе за границы строки.
4. Разделение данных происходит корректно при изменении символов строки, не затрагивая копии.

Заключение:

В результате выполнения задания был создан и протестирован класс для работы со строками с подсчетом ссылок, который поддерживает конкатенацию, преобразование к типу `const char*`, а также доступ и изменение отдельных символов строки.

Реализованные проверки подтверждают корректность работы всех операторов и методов класса.