Вариант 2

Тема

Классы с подсчетом ссылок. Переопределение операций, работа с динамической памятью.

Представляем, что у нас есть доска, на которой записана строка. Вместо того, чтобы каждый раз, когда нам нужна эта строка, делать новую копию доски, мы просто даем каждому, кому нужна строка, указатель на эту доску.

**Подсчет:** к доске прикреплен счетчик. Каждый раз, когда кто-то начинает “смотреть” на доску, счетчик увеличивается. Когда кто-то перестает “смотреть”, счетчик уменьшается.

**Удаление:** когда счетчик доходит до нуля, это означает, что никто больше не смотрит на доску. В этот момент доску (память, выделенную под строку) можно безопасно убрать.

В моем случае, доска - это область памяти, содержащая символы строки.

Чтобы избежать того, что при изменении 1 человеком, у другого не менялась строка, используется техника “разделения при изменении”:

**Перед изменением:** прежде чем разрешить изменить строку, нужно проверить, сколько людей “смотрят” на доску.

**Если смотрящих больше одного:** если счетчик больше 1, это означает, что строку используют несколько объектов. В этом случае:

Нужно сделать *копию* доски (выделить новую память и скопировать туда строку).

Изменить *только* эту копию.

Обновить указатель объекта, который хотел изменить строку, чтобы он указывал на эту *новую* доску.

**Если смотрящий только один:** если счетчик равен 1, это означает, что строку использует только один объект. В этом случае можно безопасно изменить строку прямо на доске, без создания копии.

**Описание тестирования:**

Были проведены следующие тесты для проверки корректности реализации:

1. **Создание и копирование строк:** проверено, что конструктор копирования правильно создает новый объект Str и увеличивает счетчик ссылок.
2. **Изменение одной строки:** проверено, что при изменении одной строки, скопированные строки не изменяются.
3. **Чтение символов:** проверено, что оператор [] для чтения (const) работает правильно и возвращает правильный символ. Проверена генерация исключения std::out\_of\_range при выходе за границы строки.
4. **Многократное изменение символов:** проверено, что при многократном изменении символов одной строки разделение данных происходит только один раз.
5. **Присваивание:** проверена работа оператора присваивания, в том числе корректное уменьшение счетчика ссылок у старого объекта и увеличение счетчика ссылок у нового.