Подсчет ссылок реализован с помощью класса \_str, который в качестве полей хранит непосредственно счетчик ссылок и указатель на данные строки типа char\*. Его методы доступны только дружественному классу Str, который в качестве поля содержит только указатель на объект класса подсчета ссылок.

Таким образом, при создании копии объекта класса Str вместо создания ненужной копии данных новый объект Str начинает ссылаться на тот же объект подсчета ссылок \_str, что и тот, копию которого мы создаем, а счетчик ссылок \_str увеличивается на 1. В итоге, данные копируемой строки становятся доступны для новой копии через объект \_str, так как он содержит указатель на данные копируемой строки.

При вызове деструктора Str мы не уничтожаем ресурс, а вызываем метод класса \_str, который уменьшает счетчик ссылок. Если счетчик ссылок становится равным 0, то \_str удаляет сам себя.

Разделение при изменении – механизм, который позволяет избежать изменение данных всех связанных объектов (копий) при изменении одного из них. То есть, если мы хотим изменить одну из копий, нужно сделать так, чтобы она начала ссылаться на новый объект класса подсчета ссылок, который в свою очередь будет содержать указатель на новые, измененные данные.

Вариант 4

Нужно реализовать публичный метод void reverse (), который изменяет порядок символов в строке на обратный. Поскольку строка после применения такого метода, как правило, меняется, в методе реализован механизм разделения памяти. Сначала я реализовал его методом перестановки значений непосредственно в копии данных строки StrTmp (необходима для реализации механизма разделения памяти), но чтобы избежать лишнего использования функции копирования strcpy\_s, я выбрал другой метод, в котором символы сразу размещаются на нужные места в StrTmp. Выбирая соответствующие символы из начала и конца исходной строки до тех пор, пока не дойдем до середины, записываем «начальный» символ в конец StrTmp, а «конечный» - в начало. Последний элемент новой строки обозначаем нулевым символом \0.

Тестирование работоспособности реализованного метода заключалось в проверке работоспособности механизма разделения памяти (создавалась копия строки, после чего изменялась исходная строка и проверялось, была ли вслед за ней изменена копия), а также проверке правильного функционирования метода в случае строки нулевой или единичной длины и палиндрома.