**Отчет по лабораторной работе № 2**

**Реализация подсчета ссылок в классе строк и разделение данных при изменении**

При выполнении данной лабораторной работы был реализован механизм подсчета ссылок в классе строк. Это позволяет обеспечить корректное освобождение памяти, когда несколько переменных ссылаются на одни и те же данные.

**Описание подсчета ссылок:**

При подсчете ссылок в представленном коде переменная *m\_nCoun*t используется для отслеживания количества ссылок на данные строки. Каждый раз, когда создается новый объект *\_str* (класс, содержащий строковые данные и информацию о количестве ссылок), переменная *m\_nCount* увеличивается на 1. При удалении объекта этот счетчик уменьшается. Когда *m\_nCount* становится равным 0, это означает, что на данные не осталось ссылок, и объект \_str уничтожается, а данные освобождаются.

Разделение при изменении данных означает, что при обновлении данных в одной копии строки, если на эту же строку ссылаются другие объекты *\_str*, данные будут скопированы перед изменением, чтобы каждая копия имела свою собственную версию. Это сохраняет целостность данных и гарантирует, что изменения в одной копии не повлияют на другие.

**Описание задания:**

Для реализации оператора, позволяющего получать значение *i*-того символа строки, был добавлен оператор *char operator [](int i) const*. Он позволяет читать каждый отдельный символ строки, при этом генерируется исключение, если индекс находится за пределами строки.

Затем оператор *char& operator [](int i*) был доработан для получения ссылки на символ строки и реализации разделения данных при изменении. Учтены условия, при которых можно обойтись без разделения данных, что значительно повышает эффективность класса при изменении символов.

**Тестирование:**

Было проведено тестирование, чтобы убедиться, что при замене символа в одной копии строки, другая копия остается неизменной.

Также было проверено, что при замене нескольких символов одной строки не происходит многократного “разделения данных”.

**Выводы:**

Реализация подсчета ссылок и разделения данных в классе строк позволяет обеспечить корректную работу с данными и избежать непредвиденных изменений при работе с несколькими копиями строки. Тестирование подтвердило корректность работы и эффективность реализованных методов.