Отчет по заданию 7, Вариант - 2. Бочков Никита 2254

**Знакомство с контейнерами STL. Приемы управления памятью. Использование умных указателей.**

Основное правило выделения памяти вектором:

1. В начальный момент времени vector имеет нулевую вместимость.
2. При добавлении элементов методом push\_back, если текущая вместимость недостаточна для нового элемента, происходит перераспределение памяти.
3. Новая вместимость вектора обычно удваивается по сравнению с текущей вместимостью.

Эффективность такого подхода:

1. Удвоение вместимости позволяет уменьшить количество перераспределений до логарифмического уровня (O(log n)), где n — количество элементов.
2. При каждом увеличении вместимости выделяется немного больше памяти, чем необходимо в данный момент, что позволяет избежать частых перераспределений при последовательном добавлении элементов.

Оценка увеличения потребности в памяти по сравнению с наивным поведением:

График показывает зависимость вместимости (capacity) от размера (size) вектора. По графику видно, что вместимость увеличивается с коэффициентом примерно 1.5 на каждый шаг при использовании компилятора Visual Studio C++. Линейная аппроксимация имеет уравнение:

сapacity = 1.5104 × size − 0.0797

Для компилятора g++ коэффициент увеличения вместимости составляет 2.

**Использование умных указателей:**

Умные указатели в C++ - объекты реализующие поведение указателей, предназначены для автоматического управления динамически выделенной памятью, чтобы избежать утечек памяти и облегчить управление временем жизни объектов. Они предоставляют автоматическую деалокацию памяти, когда объект больше не используется, что позволяет более безопасно и эффективно управлять ресурсами.

Умные указатели автоматизируют управление памятью за счет автоматического удаления объекта, когда он больше не нужен. Основные действия умных указателей:

* Автоматическое освобождение памяти
* Избежание утечек памяти
* Счетчик ссылок
* Передача владения