Задание 7 , Вариант 2

# Тема

Знакомство с контейнерами STL. Приемы управления памятью. Изучение поведения умных указателей shared\_ptr.

**Цель:** Исследовать механизм управления динамической памятью в классе std::vector в C++ и оценить его эффективность.

**Метод:** Создана консольная программа, добавляющая элементы в vector и отслеживающая изменение его размера и вместимости. Данные о вместимости перенесены в Excel для визуализации и определения закономерности.

**Результаты:**

* Вместимость std::vector увеличивается кратно, как правило, удваиваясь при заполнении.
* Найденное правило минимизирует количество перевыделений памяти по сравнению с “наивным” подходом, увеличивающим вместимость на 1 элемент за раз. Вместо 1000 перевыделений для 1000 элементов, происходит около log2(1000) ~ 10 перевыделений.
* При кратной схеме может происходить избыточное резервирование памяти, но общие затраты ресурсов на управление памятью оказываются значительно ниже.

**Вывод:** std::vector использует эффективную стратегию выделения памяти, обеспечивающую амортизированную константную сложность для добавления элементов. Кратное увеличение вместимости значительно сокращает количество перевыделений и копирований памяти, повышая производительность. В целом, std::vector представляет собой хороший компромисс между экономией памяти и производительностью.