Часть 1: Математическое множество на основе set

В этой части задания был создан проект MathSet, в котором реализованы операции объединения и пересечения множеств на основе контейнеров set и unordered\_set из STL.

Ключевые решения:

Функция для вывода множества на консоль:

Названа printSet для std::set<int> и printUnorderedSet для std::unordered\_set<int>.

Функции возвращают void и принимают константные ссылки на множества для предотвращения копирования.

Операция объединения:

Перегружен оператор +, который создает новое множество, включающее все элементы исходных множеств.

Параметры передаются по константной ссылке для эффективности.

Операция пересечения:

Перегружен оператор \*, который создает новое множество, содержащее элементы, присутствующие в обоих исходных множествах.

Часть 2: Умные указатели shared\_ptr и weak\_ptr

В этой части задания был создан проект SmartPointers, в котором продемонстрирована работа умных указателей shared\_ptr и weak\_ptr.

Назначение и действия умных указателей:

shared\_ptr:

Управляет совместным владением объекта. Когда последний shared\_ptr, владеющий объектом, уничтожается, объект также уничтожается.

Предоставляет метод use\_count(), показывающий количество shared\_ptr, которые владеют объектом.

weak\_ptr:

Создает слабую ссылку на объект, управляемый shared\_ptr, без увеличения счетчика ссылок.

Используется для решения проблемы циклических ссылок и для проверки существования объекта перед использованием.

Метод lock() пытается создать shared\_ptr из weak\_ptr, если объект еще существует.

Пример использования weak\_ptr:

Представьте, что у вас есть два объекта, которые ссылаются друг на друга через shared\_ptr. Это создаст циклическую ссылку, и объекты никогда не будут удалены, так как счетчики ссылок никогда не достигнут нуля. weak\_ptr решает эту проблему, позволяя одному из объектов иметь слабую ссылку на другой, предотвращая циклические зависимости.