В этом задании вам предлагается разработать иерархию классов Expression для представления арифметических выражений. Конкретнее, вам нужно определить три класса: Expression — базовый класс иерархии, Number — для представления чисел и BinaryOperation — класс описывающий бинарную операцию (+, ­-, \* или /).

Класс Number должен хранить значение типа double.

Класс BinaryOperation должен хранить указатель на левый и правый операнды, которые сами являются арифметическими выражениями, а также тип операции (+, ­-, \* или /), которую нужно над ними произвести.

Во всех классах должен быть метод evaluate, который возвращает значение типа double — значение соответствующего арифметического выражения, например, значение экземпляра типа Number — это число, которое он хранит, а если у вас есть объект BinaryOperation с операцией +, то нужно вычислить значения левого и правого операнда и вернуть их сумму.

В данном задании вам нужно расставить ключевое слово virtual там, где это необходимо, определить метод evaluate там, где его не хватает, а также реализовать деструкторы, там где они нужны.

При выполнении этого задания учтите, что при уничтожении объекта BinaryOperation он отвечает за уничтожение левого и правого операндов (гарантируется, что они выделены в динамической памяти).

Например, выражению 3 + 4.5 \* 5 будет соответствовать следующий код:

// сначала создаём объекты для подвыражения 4.5 \* 5

Expression \* sube = new BinaryOperation(new Number(4.5), '\*', new Number(5));

// потом используем его в выражении для +

Expression \* expr = new BinaryOperation(new Number(3), '+', sube);

// вычисляем и выводим результат: 25.5

std::cout << expr->evaluate() << std::endl;

// тут освобождаются \*все\* выделенные объекты

// (например, sube будет правым операндом expr, поэтому его удалять не нужно)

delete expr;

**Требования к реализации:** при выполнении этого задания не нужно вводить или выводить что-либо. Вы можете заводить любые вспомогательные функции, методы или классы, но не нужно реализовывать функцию main.

Программирование — Напишите программу. Тестируется через stdin → stdout