

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Детали реализации

Необходимо реализовать класс для работы с рациональными числами `Rational`.

Рациональное число представляется двумя **взаимно простыми** целыми числами - числителем и **положительным** знаменателем. Этот инвариант необходимо поддерживать при выполнении любых операций над объектами класса `Rational`. Публичный интерфейс класса должен включать:

- Конструктор по умолчанию, который инициализирует число нулем;
- Неявный конструктор преобразования от целых чисел (должен работать следующий код `Rational x = 5;`). Не забудьте добавить комментарий `// NOLINT` (см. замечания);
- Конструктор от двух целых чисел (числитель и знаменатель). При этом **не** гарантируется, что числа взаимно простые и ненулевые (об этом далее);
- Методы `GetNumerator()` и `GetDenominator()`, возвращающие числитель и знаменатель соответственно;
- Методы `SetNumerator(int)` и `SetDenominator(int)`, устанавливающие числитель и знаменатель в требуемые значения;
- Бинарные арифметические операции (+, -, /, *) и их присваивающие версии (+=, -=, /=, *=);
- Унарные операции (+, -), а также префиксные и постфиксные инкременты и декременты (++, --);
- Операции сравнения;
- Операции ввода из потока и вывода в поток (>>, <<);

При выводе числа в поток необходимо выводить его в виде несократимой дроби в формате <числитель>/<знаменатель> либо <числитель>, если знаменатель равен 1. Ввод осуществляется в том же формате (но дробь на входе необязательно несократима).

Пример:

```
Rational x = 5;           // дробь 5/1
std::cout << x;           // вывод: 5
x.SetDenominator(15);    // дробь 1/3 (5/15 сократилась)
std::cout << x;           // вывод: 1/3
std::cin >> x;            // ввод: 14/-6, дробь: -7/3 (общий множитель 2; знаменатель > 0)
```