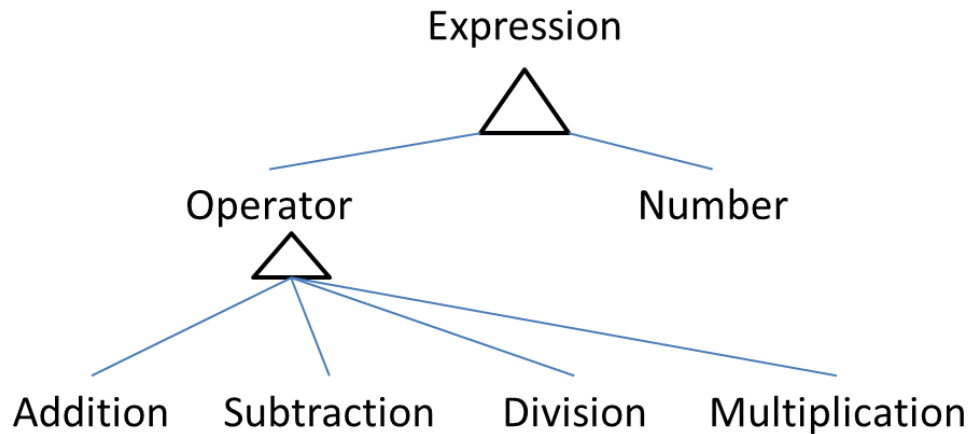


Упражнение 8. Наследование

Задание 1

Реализуйте иерархию классов для вычисления математических выражений, представленную на схеме:



Класс `Expression` содержит методы *eval* и *print*. Метод *eval* вычисляет значение выражения и возвращает результат типа `double`. Метод *print* печатает выражение в инфиксной форме.

```
class Expression
{
public:
    virtual double eval() const = 0;
    virtual void print() const = 0;
};
```

Класс `Number` хранит переданное в конструкторе число.

Класс `Operator` реализует понятие бинарного оператора и должен содержать два поля – левый и правый операнды и виртуальный метод `calculate(double left, double right)`. Реализации этого метода в потомках должны производить соответствующие классу действия над операндами, например, складывать в классе `Addition`.

Для передачи в конструктор и хранения адресов левого и правого операндов в классе `Operator` используйте «умный указатель» `shared_ptr`.

Для всех методов и операторов должны быть реализованы модульные тесты.

Пример использования:

```
Multiplication m(make_shared<Number>(2.0),
                 make_shared<Addition>(make_shared<Number>(2.0),
                                     make_shared<Number>(3.0)));
m.print();
cout << endl << m.eval() << endl;
```

Этот код должен напечатать:

```
(2*(2+3))
10
```

Дополнительное задание

Добавьте класс Negation, реализующий операцию изменения знака.

Оценка задания

Реализация вычислений (функция eval)	– 50%;
Реализация печати (функция print)	– 50%;
Дополнительное задание	– 20%.

Баллы за каждую задачу включают:

- правильность решения задания – 40 %;
- оформление кода согласно <http://prog.tversu.ru/pr3/codeStyle.pdf> – 10%;
- модульные тесты – 50%.