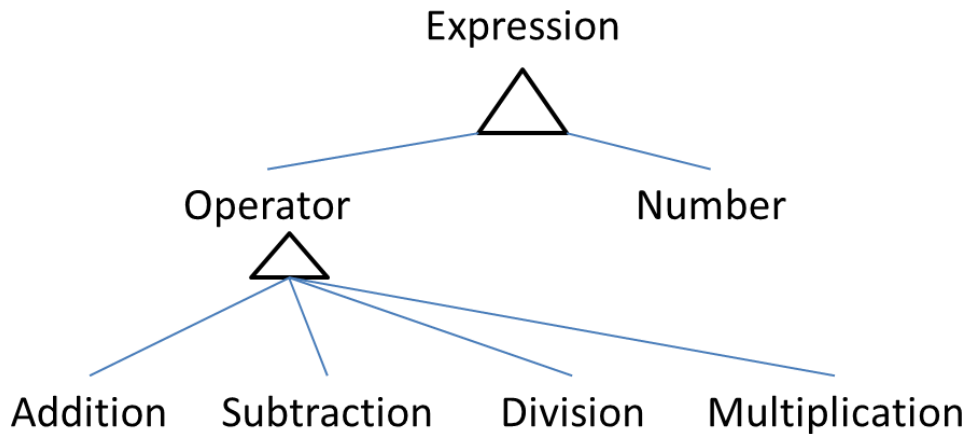


Задание 11. Наследование

Задача 1

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



Класс `Expression` объявляет методы *eval* и *print*. Метод *eval* должен вычислить значение выражения и вернуть результат типа `double`. Метод *print* будет печатать выражение в инфиксной форме в поток. В классе `Expression` эти функции объявляются как чистые виртуальные функции (без реализации).

```
class Expression
{
public:
    virtual double eval() const = 0;
    virtual void print(ostream &stream) 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 ссылки.

Для всех методов и операторов должны быть реализованы модульные тесты.

Пример использования:

```
Number two(2.0);
Number three(3.0);
Number anotherTwo(2.0);
Addition addition(two, three);

Multiplication m(anotherTwo, addition);
m.print(cout);
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 (унарная операция Negation)	– 20%.

Баллы за каждую задачу включают:

- правильность решения задания 60 %;
- оформление кода согласно <http://prog.tversu.ru/pr3/codeStyle.pdf> 10 %;
- программа компилируется без предупреждений компилятора на 4-ом уровне 10%;
- модульные тесты 20%.