

#### Задание 4

Используя подход, предлагаемый паттернами компоновщик и приспособленец, реализуйте фрагмент системы классов для представления арифметических выражений.

Система должна включать классы для нескольких арифметических операторов, переменных (хранит имя переменной) и констант (хранит значение). Эти классы должны обеспечить представление выражения в виде дерева из операторов с переменными или константами в листьях.

В классах операторов, констант и переменных должны быть реализованы функции для:

- печати выражения;
- вычисления значения выражения. Этой функции в качестве параметра передаётся `std::map<string, int>` в которой хранятся значения для всех переменных из выражения.

Переменные и константы должны быть реализованы в виде приспособленцев – если в выражении несколько раз встречается одинаковая переменная или константа, все её вхождения должны быть реализованы в виде одного объекта.

Следует разработать фабрику, которая должна иметь методы для создания и удаления объектов переменных и констант. В фабрике должны быть заранее созданы (и никогда не удаляться) объекты для констант от -5 до 256 (так сделано в языке Python). Остальные объекты создаются по требованию и удаляются когда перестают использоваться. Для удаления объектов в фабрике должны быть предусмотрены соответствующие функции.

Ниже представлен пример, показывающий использование такой системы классов для вычисления значения выражения  $2 + x$  при  $x = 3$ .

```
ExpressionFactory factory;
Constant *c = factory.createConstant(2);
Variable *v = factory.createVariable("x");

Addition *expression = new Addition(c, v);

map<string, int> context;
context["x"] = 3;

cout << expression->calculate(context) << endl;

delete expression; // Все "нижележащие" объекты должны
                  // быть освобождены деструктором.
```