

## Деревья, обходы деревьев

1. Реализовать класс «Двоичное дерево целых чисел».

Методы:

- служебные (конструкторы, деструктор, операторы присваивания),
- вставка элемента в дерево. Метод получает на вход число  $x$ , которое нужно вставить, и последовательность из 0 и 1, кодирующую путь в дереве: 0 – переход к левому потоку текущего узла, 1 – к правому. Если путь ведет в уже существующий узел, то число в нем заменяется на  $x$ . Если путь проходит через лист дерева и заканчивается на уровень ниже, то в это место добавляется новый узел как потомок листа. В остальных случаях — ошибка.
- оператор вывода (вывод всех элементов дерева в поток слева направо).

2. Добавьте в написанный класс методы:

- количество четных чисел в дереве,
- проверка того, что в дереве только положительные числа (в пустом дереве считаем результат true),
- удаление в дереве всех листьев (при этом листьями станут новые узлы, их не трогаем),
- среднее арифметическое всех чисел в дереве,
- поиск заданного элемента  $x$  в дереве (если такое число входит в дерево несколько раз, то одного любого вхождения), результат — путь от корня к найденному узлу в виде массива из 0 и 1. Массив можно хранить в виде объекта класса `std::vector`.

3. \*Добавьте в класс метод проверки того, что дерево является деревом двоичного поиска.

4. \*Вывод дерева в поток по уровням. Дерево при выводе можно развернуть на 90 градусов, т.е. вместо

50

20 70

10 33 60 100

выводить

50

20

10

33

70

60

100