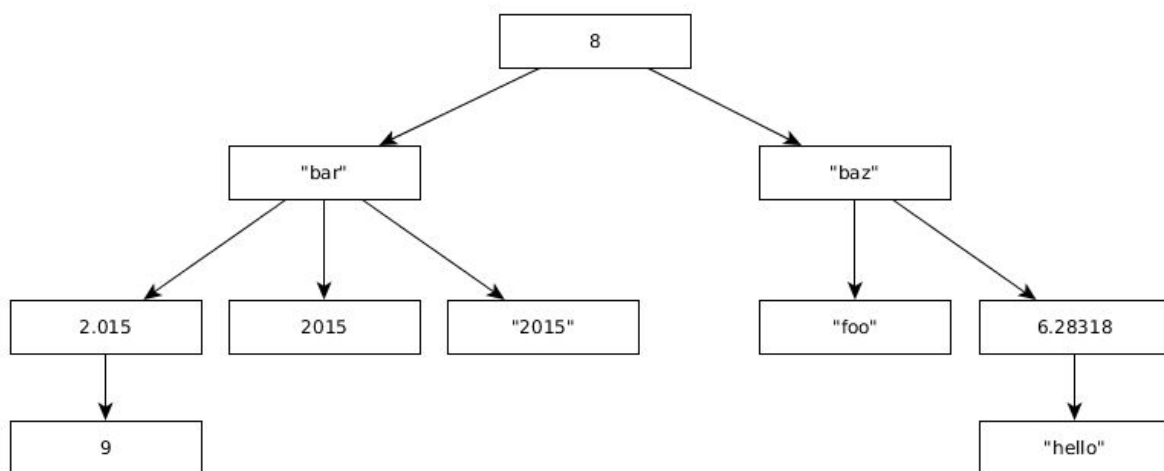


Гетерогенное дерево

Дано дерево, каждый узел которого имеет произвольное количество потомков и хранит данные разных типов, например:

- строка;
- целое число;
- вещественное число.

Пример такого дерева показан на рисунке ниже.



Необходимо:

- реализовать структуру данных, которая способна представлять подобное дерево;
- реализовать вывод такого дерева в консоль в произвольном виде, позволяющем увидеть иерархию его узлов;
- разработать формат хранения сериализованного дерева;
- реализовать сериализацию дерева в разработанном формате в файл;
- реализовать десериализацию дерева в разработанном формате из файла.

Представление дерева, его распечатка, сериализация и десериализация должны быть реализованы самостоятельно, без использования сторонних библиотек.

Для работы с деревом нужно реализовать консольное приложение, которое принимает на вход следующие параметры:

- `-i` — имя входного файла, из которого будет десериализовано дерево;
- `-o` — имя выходного файла, в который будет сериализовано дерево.

Например, программа может быть запущена так:

```
app -i input -o output
```

В этом случае программа должна десериализовать дерево из файла `input`, вывести его на экран и сериализовать в файл `output`.

Для разбора аргументов командной строки можно использовать сторонние библиотеки.

Мы ожидаем увидеть пример качественного production-кода, в частности:

- соблюдение единого стиля кодирования;
- разумное расходование памяти;
- расчёт на производительность;
- library-like архитектуру с возможностью расширения;
- задел на работу с деревьями произвольного размера;
- следование принципам ООП;
- использование стандартного C++ и, при необходимости, сторонних кроссплатформенных библиотек;
- использование modern C++ best practices.