Абстрактное дерево

1. Первая реализация
   1. Класс node содержит:
      1. id – номер вершины
      2. value – значение вершины
      3. list leftChildrens – список детей, у которых номер вершины меньше, чем у родительской
      4. list rightChildrens – список детей, у которых номер вершины больше, чем у родительской
   2. Поиск можно совершать по ключу, содержащему последовательность из id вершин, например 100 50 70 или последовательно сравнивая значения дочерних вершин
   3. Пример дерева:
2. Вторая реализация
   1. Класс node содержит:
      1. value – значение вершины
      2. list leftChildrens – список детей, у которых значение вершины меньше, чем у родительской или равно
      3. list rightChildrens – список детей, у которых значение вершины больше, чем у родительской
   2. Поиск совершается по значениям вершины путем сравнения значений
   3. Пример аналогичный первому примеру, но в вершинах значения вершин
3. Третья реализация
   1. Класс node содержит:
      1. value – значение вершины
      2. list childrens – список дочерних вершин
   2. Поиск совершается по всему списку вершин, если при добавлении сортировать дочерние вершины, то можно искать бинарным поиском
4. Четвертая реализация аналогична 3 но с id