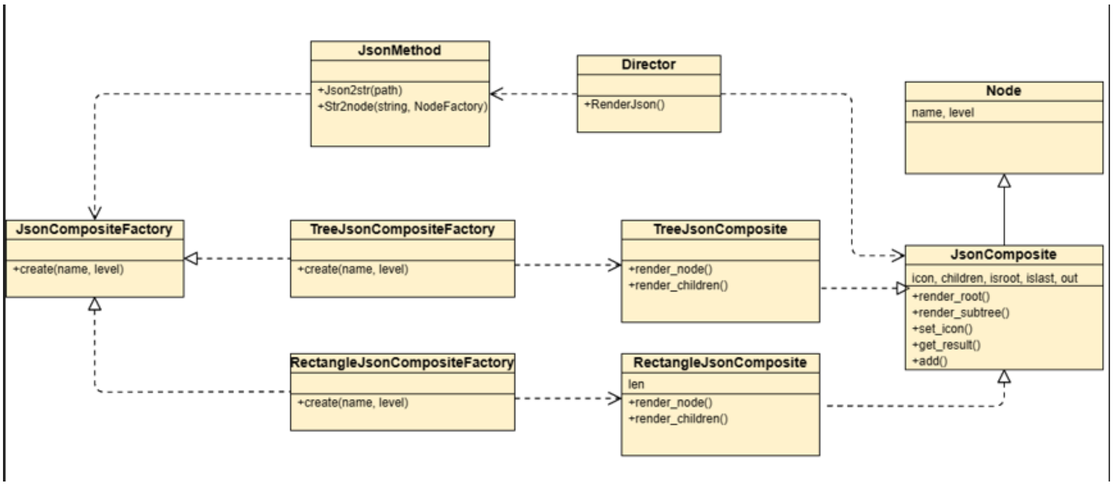
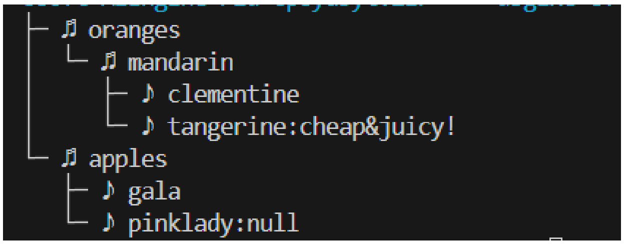


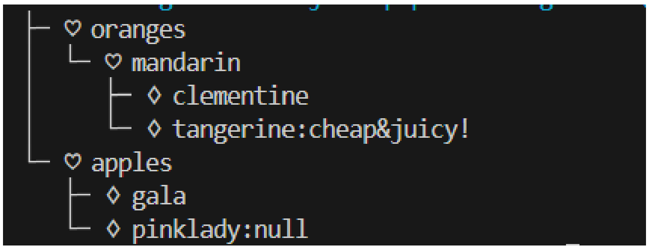
1. JsonCompositeFactory，TreeJsonCompositeFactory， RectangleJsonCompositeFactory 组成抽象工厂模式，同时也是工厂方法模式。当新增风格时，只需要新增一个继承 JsonCompositeFactory 的抽象工厂来生成对应风格的节点即可。
2. TreeJsonComposite、RectangleJsonComposite、JsonComposite、Director 组成建造者模式。当新增风格的节点时，只需要完成 render_root、render_subtree 的接口即可。
3. JsonComposite 为组合模式，为了方便把叶节点和中间节点一起表达，通过是否有子节点区分，通过组合模式，只需 render_root 一条指令即可完成对树的不同组合类型的渲染

./main -ftest.json -stree -imusic



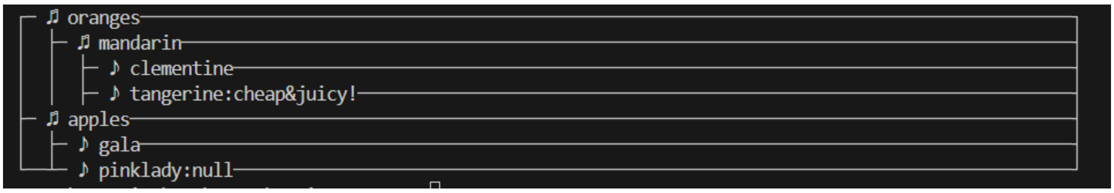
```
├─ ♫ oranges
│  └─ ♫ mandarin
│     ├─ ♪ clementine
│     └─ ♪ tangerine:cheap&juicy!
└─ ♫ apples
   ├─ ♪ gala
   └─ ♪ pinklady:null
```

./main -ftest.json -stree -ifunny



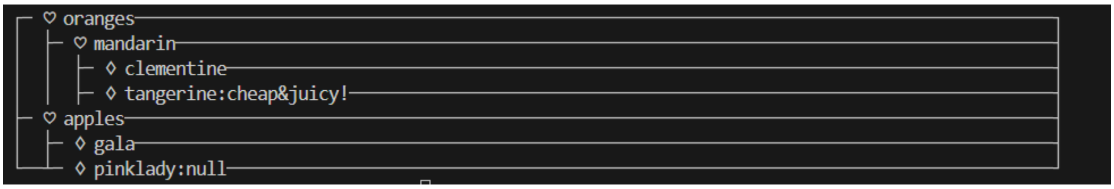
```
├─ ♡ oranges
│  └─ ♡ mandarin
│     ├─ ◊ clementine
│     └─ ◊ tangerine:cheap&juicy!
└─ ♡ apples
   ├─ ◊ gala
   └─ ◊ pinklady:null
```

./main -ftest.json -srectangle -imusic



```
┌─ ♫ oranges ─────────────────────────────────────────────┐
│  ├─ ♫ mandarin ─────────────────────────────────────────┤
│  │  ├─ ♪ clementine ────────────────────────────────────┤
│  │  ├─ ♪ tangerine:cheap&juicy! ────────────────────────┤
├─ ♫ apples ──────────────────────────────────────────────┤
│  ├─ ♪ gala ─────────────────────────────────────────────┤
└──┴─ ♪ pinklady:null ────────────────────────────────────┘
```

./main -ftest.json -srectangle -ifunny



```
┌─ ♡ oranges───────────────────────────────────────────────────────────────────────────────────────┐
│  ├─ ♡ mandarin───────────────────────────────────────────────────────────────────────────────────┤
│  │  ├─ ◊ clementine──────────────────────────────────────────────────────────────────────────────┤
│  │  ├─ ◊ tangerine:cheap&juicy!──────────────────────────────────────────────────────────────────┤
├─ ♡ apples────────────────────────────────────────────────────────────────────────────────────────┤
│  ├─ ◊ gala───────────────────────────────────────────────────────────────────────────────────────┤
└──┴─ ◊ pinklady:null──────────────────────────────────────────────────────────────────────────────┘
```