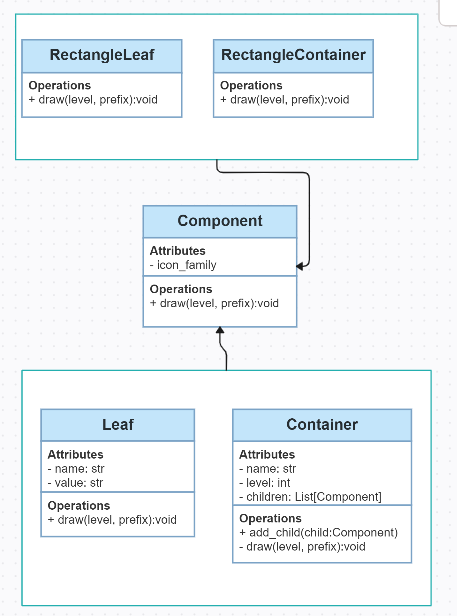
## 类及方法说明



* **Component（抽象类）**

icon\_family: 图标族，用于指定中间节点和叶节点的图标。

draw(level, prefix): 抽象方法，用于绘制节点。

* **Leaf（继承自 Component）**

name: 节点名称。

Value: 节点值。

draw(level, prefix): 绘制叶节点。

* **Container（继承自 Component）**

name: 节点名称。

level: 节点层级。

children: 子节点列表。

add\_child(child: Component): 添加子节点。

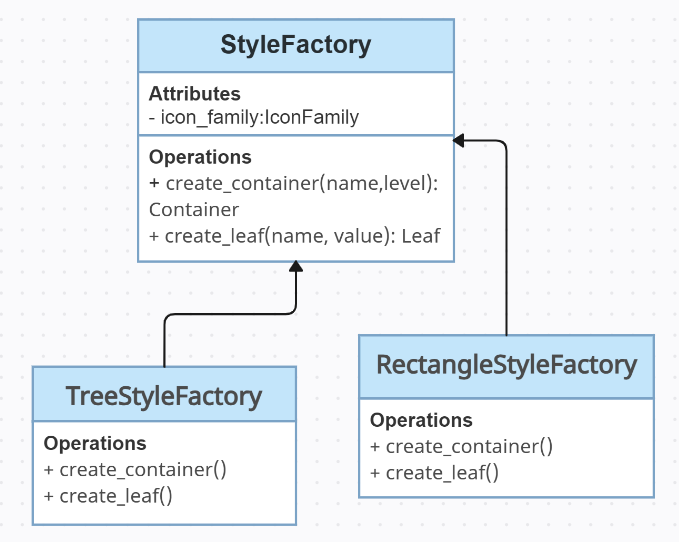
draw(level, prefix): 绘制中间节点及其子节点

* **RectangleLeaf（继承自 Leaf）**

draw(level, prefix): 绘制矩形风格的叶节点。

* **RectangleContainer（继承自 Container）**

draw(level, prefix): 绘制矩形风格的中间节点及其子节点。



* **StyleFactory（抽象类）**

icon\_family: 图标族。

create\_container(name, level): 创建中间节点。

create\_leaf(name, value): 创建叶节点。

* **TreeStyleFactory（继承自 StyleFactory）**

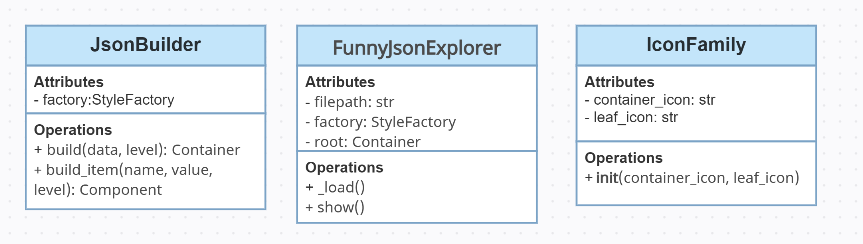
create\_container(name, level): 创建树形风格的中间节点。

create\_leaf(name, value): 创建树形风格的叶节点。

* **RectangleStyleFactory（继承自 StyleFactory）**

create\_container(name, level): 创建矩形风格的中间节点。

create\_leaf(name, value): 创建矩形风格的叶节点。



* **JsonBuilder**

factory: 节点工厂，用于创建节点。

build(data, level): 构建 JSON 数据的根节点。

build\_item(name, value, level): 构建 JSON 数据的子节点。

* **FunnyJsonExplorer**

filepath: JSON 文件路径。

factory: 节点工厂，用于创建节点。

root: 根节点。

\_load(): 加载 JSON 文件数据并构建节点树。

show(): 显示节点树。

* **IconFamily**

container\_icon: 中间节点的图标。

leaf\_icon: 叶节点的图标。

init(container\_icon, leaf\_icon): 初始化图标族。

## 使用的设计模式及作用

* **组合模式（Composite Pattern）**

作用：将对象组合成树形结构以表示“部分-整体”的层次结构，使得客户端可以统一对待单个对象和组合对象。

应用：Component、Leaf 和 Container 类。

* **抽象工厂模式（Abstract Factory Pattern）**

作用：提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

应用：StyleFactory、TreeStyleFactory 和 RectangleStyleFactory 类。

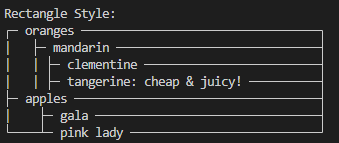
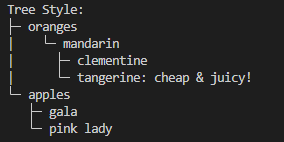
* **建造者模式（Builder Pattern）**

作用：将一个复杂对象的构建过程与它的表示分离，使得同样的构建过程可以创建不同的表示。

应用：JsonBuilder 类。

## 运行结果

* **无图标：**



* **有图标：**

