**Section 1 环境**

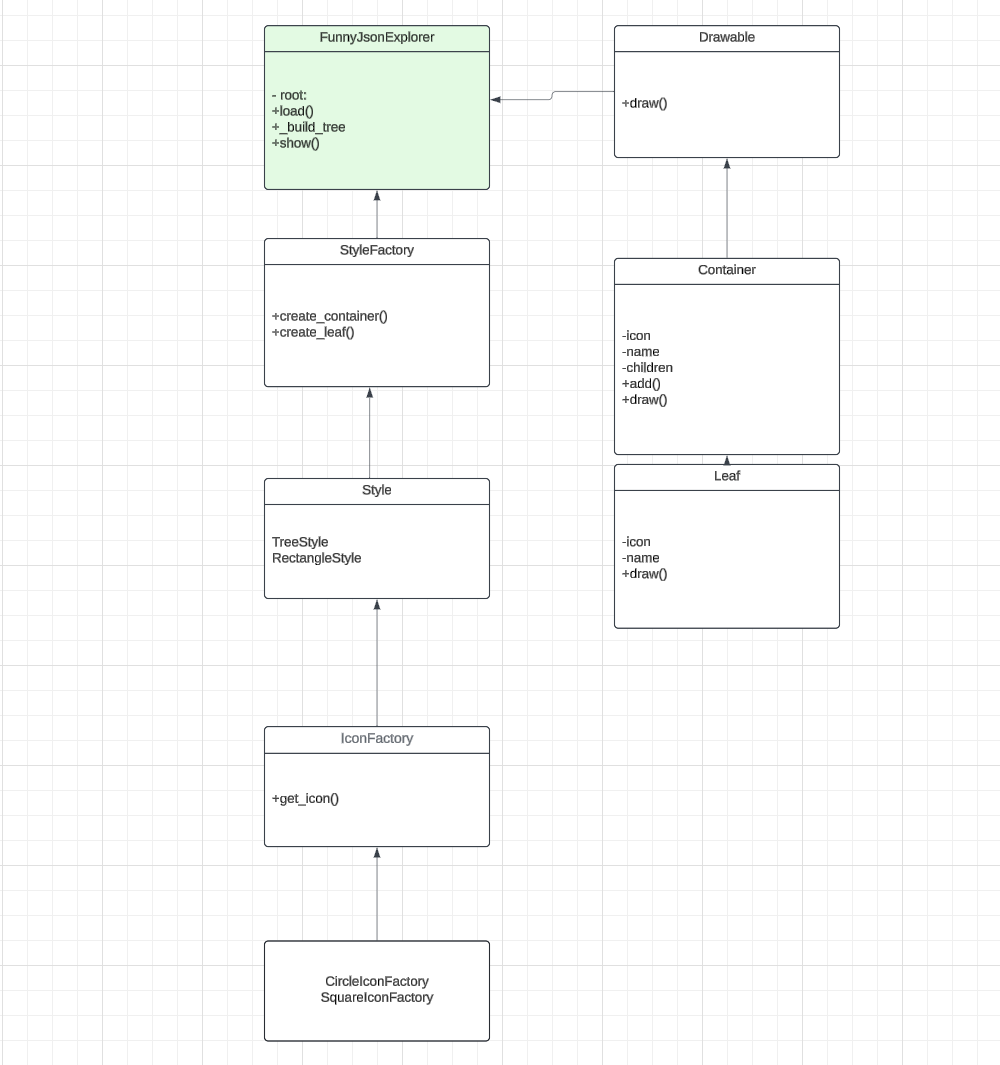
系统：windows10  
IDE：pycharm

**Section 2 设计模式**

* **工厂模式：**在这个项目中，TreeStyleFactory 和 RectangleStyleFactory 分别用于在树形结构和矩形结构下创建容器和叶子节点，它们根据参数动态地创建适当的对象，使得代码更加灵活可扩展。
* **抽象工厂模式：**在这个项目中，IconFactory 抽象工厂定义了一个接口来获取图标，而具体的图标工厂类 CircleIconFactory 和 SquareIconFactory 实现了该接口，并分别提供圆形和方形的图标。这样，当需要更换图标类型时，只需要更改工厂类即可，而不需要修改其他代码。

**Section 3 类图**

* explorer.py：定义了 FunnyJsonExplorer 类，负责加载 JSON 文件并构建树形结构。
* drawable.py：定义了可绘制对象的抽象类 Drawable，以及其子类 Container 和 Leaf，用于表示树形结构中的容器节点和叶子节点。
* factories.py：定义了工厂类，包括抽象工厂类 IconFactory 和其子类 CircleIconFactory、SquareIconFactory，以及样式工厂类 TreeStyleFactory 和 RectangleStyleFactory。
* tree\_style.py：提供了在树形结构下绘制节点的类 TreeContainer 和 TreeLeaf，以及相应的样式工厂类 TreeStyleFactory。
* rectangle\_style.py：提供了在矩形结构下绘制节点的类 RectangleContainer 和 RectangleLeaf，以及相应的样式工厂类 RectangleStyleFactory。
* icon\_mapping.py：定义了图标类型到符号的映射。



**Section 4 运行与结果**

在 Windows 上，打开命令提示符（cmd）或 PowerShell 并导航到项目根目录 “FunnyJsonExplorer/”,运行以下命令：

python main.py -f example.json -s tree -i dollar

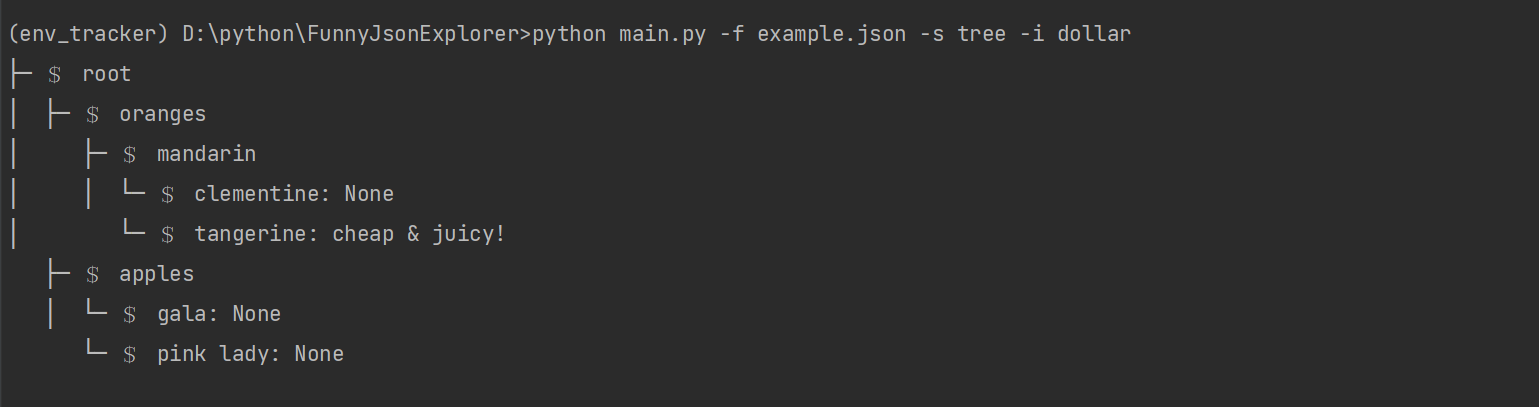
python main.py -f example.json -s rectangle -i dollar

python main.py -f example.json -s tree -i yuan

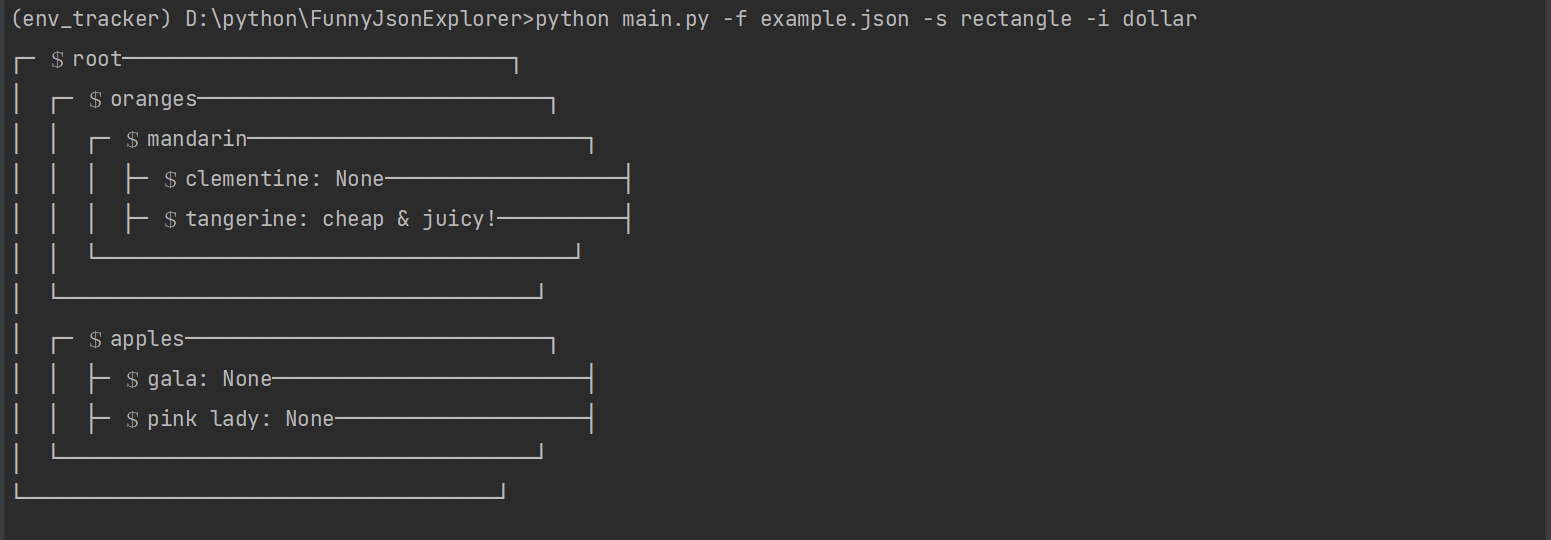
python main.py -f example.json -s rectangle -i yuan

得到结果：

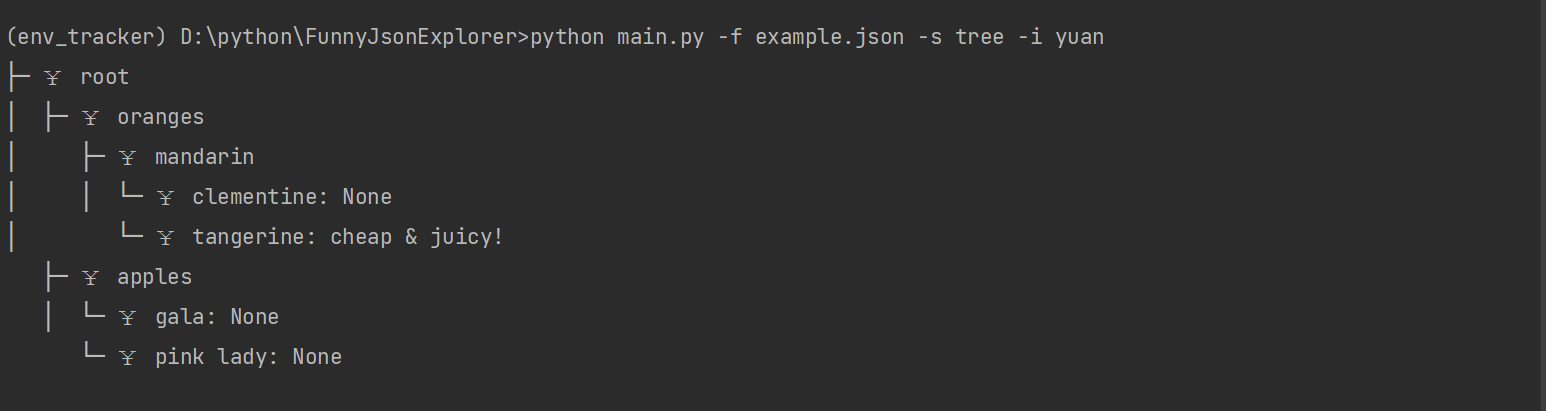
python main.py -f example.json -s tree -i dollar



python main.py -f example.json -s rectangle -i dollar



python main.py -f example.json -s tree -i yuan



python main.py -f example.json -s rectangle -i yuan

