



项目课程: 软件工程

项目名称: JSON文件可视化

专业名称: 计算机科学与技术

学生姓名: 吴臻

学生学号: 21307371

项目要求

设计文档

类图

设计模式

项目结果展示

项目要求

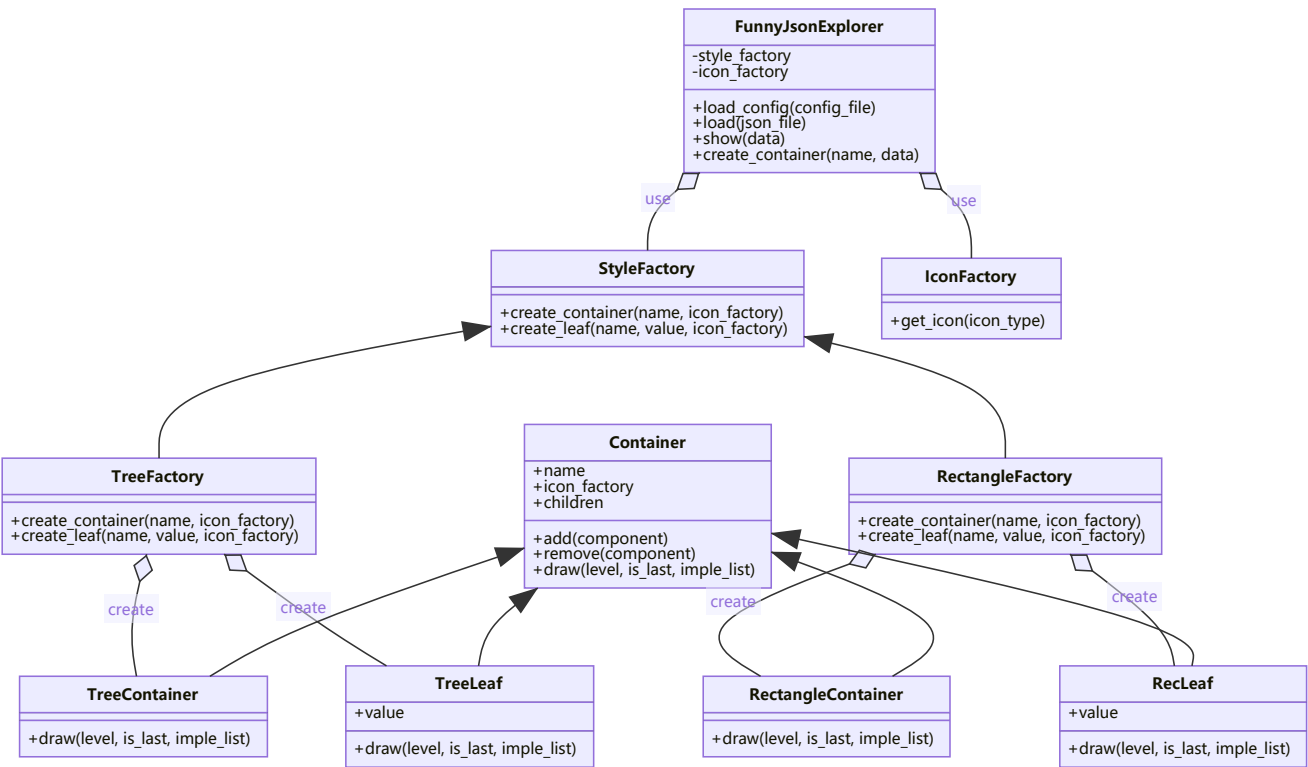
Funny JSON Explorer (FJE) ， 是一个JSON文件可视化的命令行界面小工具。

- 1. FJE可以快速切换**风格** (style) ， 包括： 树形 (tree) 、 矩形 (rectangle) ；
- 2. 也可以指定**图标族** (icon family) ， 为中间节点或叶节点指定一套icon
- 3. 通过**配置文件**(icon_config.json)， 可添加新的图标族

源代码仓库： <https://github.com/wz-wz111/Software-Engineering>

设计文档

类图



1. 在这个类图中，`FunnyJsonExplorer` 类与 `IconFactory` 和 `StyleFactory` 类之间的关系用 `o--` 表示(use)，这是一种组合关系，意味着 `FunnyJsonExplorer` 类拥有 `IconFactory` 和 `StyleFactory` 的实例，并且这些实例是 `FunnyJsonExplorer` 对象的一部分。
2. `Container` 是一个抽象基类，它定义了 `add`、`remove` 和 `draw` 方法的接口，以及 `name`、`icon_factory` 和 `children` 属性。`Tree_Leaf` 和 `Rec_Leaf` 是 `Container` 的子类，它们是容器的叶子节点。它们实现了 `draw` 方法以显示自己的内容，并且具有 `value` 属性来存储显示的数据。`TreeContainer` 和 `RectangleContainer` 也是 `Container` 的子类，它们是容器节点，可以包含其他 `Container` 或叶子节点作为子节点。它们实现了 `draw` 方法以显示容器的结构和内容。

设计模式

1. 抽象工厂 (Abstract Factory) :

`StyleFactory` 类扮演了抽象工厂的角色，它定义了两个抽象方法 `create_container` 和 `create_leaf`，这些方法由具体的工厂类实现。

- `StyleFactory` 类：定义了创建容器和叶子节点的接口。
- `RectangleFactory` 和 `TreeFactory` 类：这些类继承自 `StyleFactory` 并实现了 `create_container` 和 `create_leaf` 的具体方法，它们是抽象工厂的具体实现。

2. 建造者 (Builder) 模式:

建造者模式在 `FJE.py` 文件中的 `FunnyJsonExplorer` 类中体现。该类使用建造者模式来构建和展示 JSON 数据的结构。

- `FunnyJsonExplorer` 类在构造函数中初始化了不同的容器类和图标工厂，这些容器类和图标工厂用于构建和展示 JSON 数据。
- `create_container` 方法递归地构建容器和叶子节点，这是建造者模式中的建造过程。
- `show` 方法调用容器的 `draw` 方法来展示构建的结果。

3. 组合模式 (Composition) :

组合模式在 `Container` 类和它的子类 `Tree_Leaf`、`Rec_Leaf`、`TreeContainer`、`RectangleContainer` 中体现。组合模式允许将对象组合成树形结构，以表示“部分-整体”的层次结构。

- `Container` 类是一个抽象基类，定义了容器的基本操作，如 `add` 和 `remove` 方法，以及一个抽象的 `draw` 方法。
- `Tree_Leaf` 和 `Rec_Leaf` 类是叶子节点，它们继承自 `Container` 类，并且实现了 `draw` 方法来展示叶子节点。
- `TreeContainer` 和 `RectangleContainer` 类是容器节点，它们也继承自 `Container` 类，并且实现了 `draw` 方法来展示容器节点及其子节点。这些类通过维护一个子组件列表 (`self.children`) 来表示组合关系。

1. 只需基于 `StyleFactory` 抽象工厂，添加新的具体工厂（即添加新的py文件），在新工厂中添加新风格的容器类和叶子类（继承 `Container` 类），就可以添加新的风格
2. 通过配置文件(`icon_config.json`)，可添加新的图标族

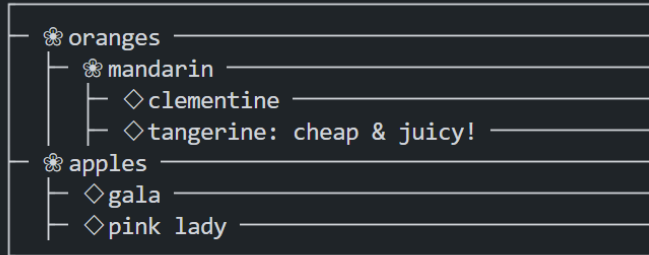
项目结果展示

运行指令：

```
1 fje -f <json file> -s <style> -i <icon family>
2
3 # icon_A可换为icon_config.json中设置的图标名字
4 fje -f example.json -s tree -i icon_A
5 fje -f example.json -s tree -i icon_B
6
7 fje -f example.json -s rectangle -i icon_A
8 fje -f example.json -s rectangle -i icon_B
```

```
PS D:\学习资料\大三下\软件工程\funny_json_explorer\src> fje -f example.json -s tree -i icon_A
├─ 🍊 oranges
│   └─ 🍊 mandarin
│       ├── ◇ clementine
│       └─ ◇ tangerine: cheap & juicy!
└─ 🍏 apples
    ├── ◇ gala
    └─ ◇ pink lady
PS D:\学习资料\大三下\软件工程\funny_json_explorer\src> fje -f example.json -s tree -i icon_B
├─ □ oranges
│   └─ □ mandarin
│       ├── ☆ clementine
│       └─ ☆ tangerine: cheap & juicy!
└─ □ apples
    ├── ☆ gala
    └─ ☆ pink lady
```

```
(base) D:\学习资料\大三下\软件工程\funny_json_explorer\src>fje -f example.json -s rectangle -i icon_A
```



```
(base) D:\学习资料\大三下\软件工程\funny_json_explorer\src>fje -f example.json -s rectangle -i icon_B
```

