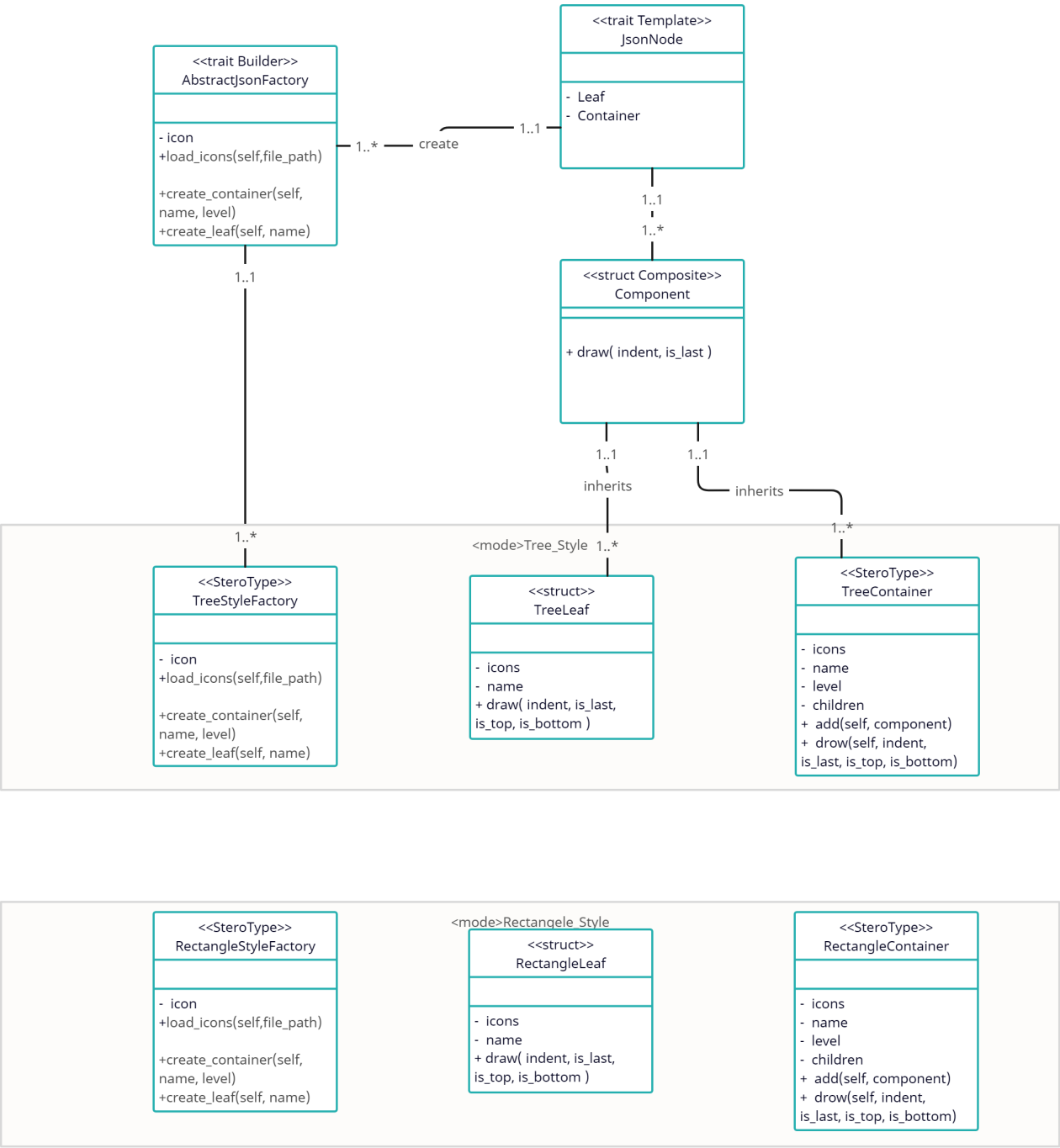


设计文档: 类图与说明

类图



说明

组件类 (Component)

- Component**: 抽象基类，定义了 `draw` 方法，所有组件（容器和叶子）都必须实现该方法。

树状风格类 (Tree Style Classes)

- **TreeContainer**: 树状风格的容器类, 继承自 `Component`。包含了一个容器的基本属性和方法, 如 `add` 和 `draw`。
- **TreeLeaf**: 树状风格的叶子类, 继承自 `Component`。实现了叶子的 `draw` 方法。

矩形风格类 (Rectangle Style Classes)

- **RectangleContainer**: 矩形风格的容器类, 继承自 `Component`。实现了矩形容器的 `draw` 方法。
- **RectangleLeaf**: 矩形风格的叶子类, 继承自 `Component`。实现了矩形叶子的 `draw` 方法。

工厂类 (Factory Classes)

- **TreeStyleFactory**: 用于创建树状风格的容器和叶子。通过 `load_icons` 方法加载图标, 并通过 `create_container` 和 `create_leaf` 方法创建相应的组件。
- **RectangleStyleFactory**: 用于创建矩形风格的容器和叶子。通过 `load_icons` 方法加载图标, 并通过 `create_container` 和 `create_leaf` 方法创建相应的组件。

功能类 (Function Classes)

- **FunnyJSONExplorer**: 包含加载JSON文件 (`load_json`), 构建树 (`build_tree`) 和显示树 (`display_tree`) 的功能。 `main` 方法用于解析命令行参数并执行相应的功能。

使用的设计模式及作用

1. 组合模式 (Composite Pattern) :

- **目的**: 组合对象成树形结构以表示“部分-整体”的层次结构。组合模式使得用户对单个对象和组合对象的使用具有一致性。
- **作用**: `Component` 类定义了统一的接口, `TreeContainer` 和 `TreeLeaf`, `RectangleContainer` 和 `RectangleLeaf` 分别实现了该接口, 使得容器和叶子可以被一致地处理。

2. 工厂模式 (Factory Pattern) :

- **目的**: 定义一个创建对象的接口, 但让子类决定实例化哪一个类。工厂方法使得类的实例化延迟到子类。
- **作用**: `TreeStyleFactory` 和 `RectangleStyleFactory` 分别负责创建树状风格和矩形风格的容器和叶子, 封装了对象的创建过程, 确保了代码的灵活性和可扩展性。

3. 策略模式 (Strategy Pattern) :

- **目的**: 定义一系列算法, 把它们一个个封装起来, 并且使它们可以互相替换。本模式使得算法可独立于使用它的客户而变化。
- **作用**: 通过选择不同的工厂 (`TreeStyleFactory` 或 `RectangleStyleFactory`), 可以动态地改变输出的风格 (树状或矩形), 符合策略模式的定义。

4. 抽象工厂模式 (Abstract Factory Pattern)

- **目的**: 抽象工厂模式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。
- **作用**: `TreeStyleFactory` 和 `RectangleStyleFactory` 这些工厂类封装了具体对象的创建过程, 确保了客户端代码不需要直接依赖于具体类, 从而实现了创建逻辑和使用逻辑的分离。

实验截图

```
(jiqui) e:\软件工程\design>python fje.py -f example.json -s tree -i icons1.json
```

```
└ oranges
  └ mandarin
    ├── clementine: None
    └ tangerine: cheap & juicy!
└ apples
  ├── gala: None
  └ pink lady: None
```

```
(jiqui) e:\软件工程\design>python fje.py -f example.json -s tree -i icons2.json
```

```
└◇ oranges
  └◇ mandarin
    ├──◇ clementine: None
    └◇ tangerine: cheap & juicy!
└◇ apples
  ├──◇ gala: None
  └◇ pink lady: None
```

```
(jiqui) e:\软件工程\design>python fje.py -f example.json -s rectangle -i icons1.json
```

└ oranges
└└ mandarin
└└└ clementine: None
└└└ tangerine: cheap & juicy!
└ apples
└└ gala: None
└└ pink lady: None

```
(jiqui) e:\软件工程\design>python fje.py -f example.json -s rectangle -i icons2.json
```

└◇ oranges
└└◇ mandarin
└└└◇ clementine: None
└└└◇ tangerine: cheap & juicy!
└◇ apples
└└◇ gala: None
└└◇ pink lady: None