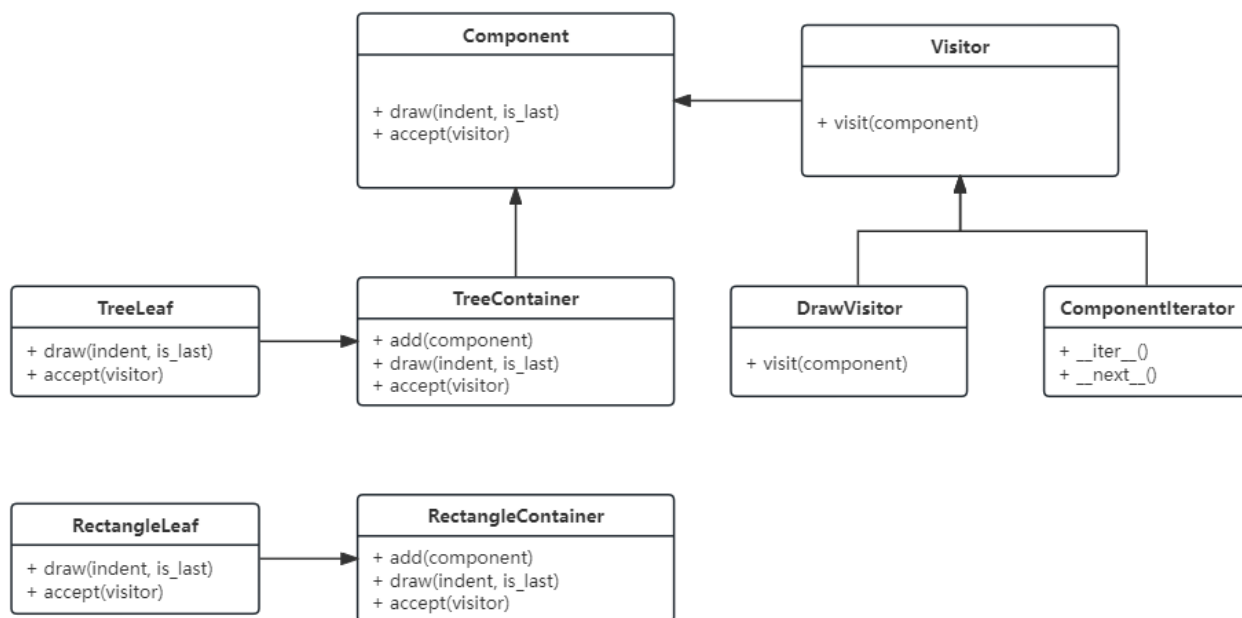


github网址: <https://github.com/lijie9/FIE-PLUS>

## 设计文档: 类图与说明

### 类图



### 设计模式

#### 1. 组件模式 (Composite Pattern)

- **Component**: 抽象基类, 定义了 `draw` 和 `accept` 方法。
- **TreeContainer** 和 **RectangleContainer**: 组合容器, 可以包含子组件。
- **TreeLeaf** 和 **RectangleLeaf**: 叶子组件, 没有子组件。

#### 2. 迭代器模式 (Iterator Pattern)

- **ComponentIterator**: 用于遍历组件树的迭代器类。

#### 3. 访问者模式 (Visitor Pattern)

- **Visitor**: 抽象访问者基类, 定义了 `visit` 方法。
- **DrawVisitor**: 具体访问者类, 实现了 `visit` 方法, 用于绘制组件。

## 详细说明

### 组件模式

组件模式允许你将对象组合成树形结构来表示“部分-整体”的层次结构。该模式使得用户对单个对象和组合对象的使用具有一致性。在代码中, `Component` 是一个抽象基类, 而 `TreeContainer` 和 `RectangleContainer` 是容器类, 能够包含其他组件。 `TreeLeaf` 和 `RectangleLeaf` 则是叶子节点。

### 迭代器模式

迭代器模式提供了一种方法顺序访问一个聚合对象中的各个元素，而不暴露其内部的表示。在代码中，`ComponentIterator` 是一个迭代器类，用于遍历组件树中的各个组件。

## 访问者模式

访问者模式封装了作用于某种数据结构中的各元素的操作，它可以在不改变各元素类的前提下定义作用于这些元素的新操作。在代码中，`Visitor` 是一个抽象基类，而 `DrawVisitor` 是一个具体访问者类，实现了绘制组件的功能。

### 实验截图

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

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