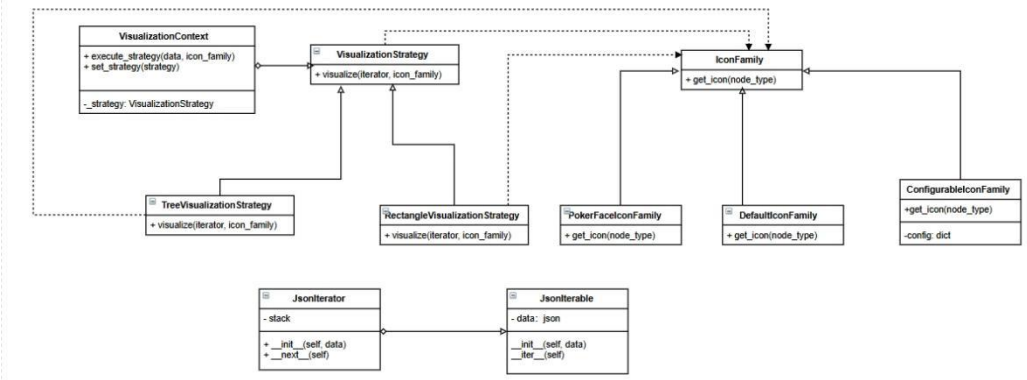
类图：上面是策略模式，下面是迭代者模式



设计说明：使用的设计模式及作用

策略模式（Strategy Pattern）：

在 strategy.py 文件中定义了 VisualizationStrategy 抽象类及其具体实现 TreeVisualizationStrategy 和 RectangleVisualizationStrategy。

在 icon_family.py 文件中定义了 IconFamily 抽象类及其具体实现 DefaultIconFamily、PokerFaceIconFamily 和 ConfigurableIconFamily。

在 context.py 文件中定义了 VisualizationContext 类，用于动态设置和执行策略。

迭代器模式（Iterator Pattern）：

JsonIterator 和 JsonIterable：通过迭代器模式，能够遍历复杂的 JSON 数据结构，而不需要了解其内部的具体实现。JsonIterator 实现了遍历逻辑，JsonIterable 提供了一个接口来生成迭代器。

项目结构：

```
funny-json-explorer/
├── config/
│   └── icon_config.json
├── fje/
│   ├── __init__.py
│   ├── iterator.py
│   ├── strategy.py
│   ├── context.py
│   ├── icon_family.py
│   └── main.py
└── setup.py
```