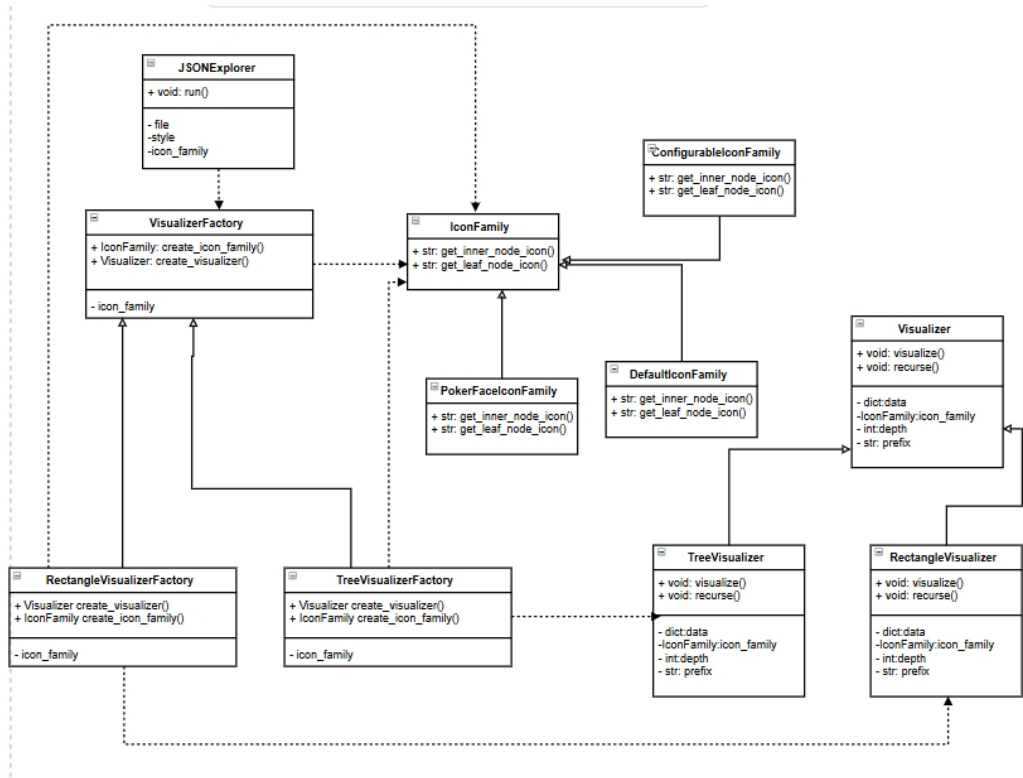


类图：



设计说明：使用的设计模式及作用

工厂方法（Factory Method）模式：

VisualizerFactory 是工厂方法模式的具体工厂类，负责创建具体的可视化器实例（**TreeVisualizer** 或 **RectangleVisualizer**）。它允许客户端代码根据需要动态选择要创建的对象，而无需关心对象的具体类型。

TreeVisualizer 和 **RectangleVisualizer** 是具体的产品类，它们实现了 **Visualizer** 接口，提供了对数据进行可视化的方法。

抽象工厂（Abstract Factory）模式：

，在 **VisualizerFactory** 中的 **create_icon_family()** 方法被视为抽象工厂模式的一种形式，因为它负责创建不同图标族的实例，而具体的图标族类（如 **PokerFaceIconFamily** 和 **DefaultIconFamily**）则实现了相同的接口（**IconFamily**）。

建造者（Builder）模式：

可视化器的构建过程被视为建造者模式。可视化器（**Visualizer**）的构建包括两个步骤：选择合适的具体可视化器类和图标族类。**VisualizerFactory** 可以被视为一个的建造者。

组合（Composition）模式：

TreeVisualizer 和 **RectangleVisualizer** 类的内部递归方法使用了组合模式。这些方法递归地处理复杂的数据结构（嵌套的字典和列表），并根据数据的层次结构逐层展开和可视化。

项目结构:

```
FunnyJSONExplorer/  
|  
├─ fje/  
|   ├─ __init__.py  
|   ├─ json_explorer.py  
|   ├─ visualizer.py  
|   ├─ icon_family.py  
|   └─ factories.py  
|  
├─ config/  
|   └─ icon_config.json  
|  
├─ sample.json  
├─ README.md  
├─ requirements.txt  
└─ setup.py
```