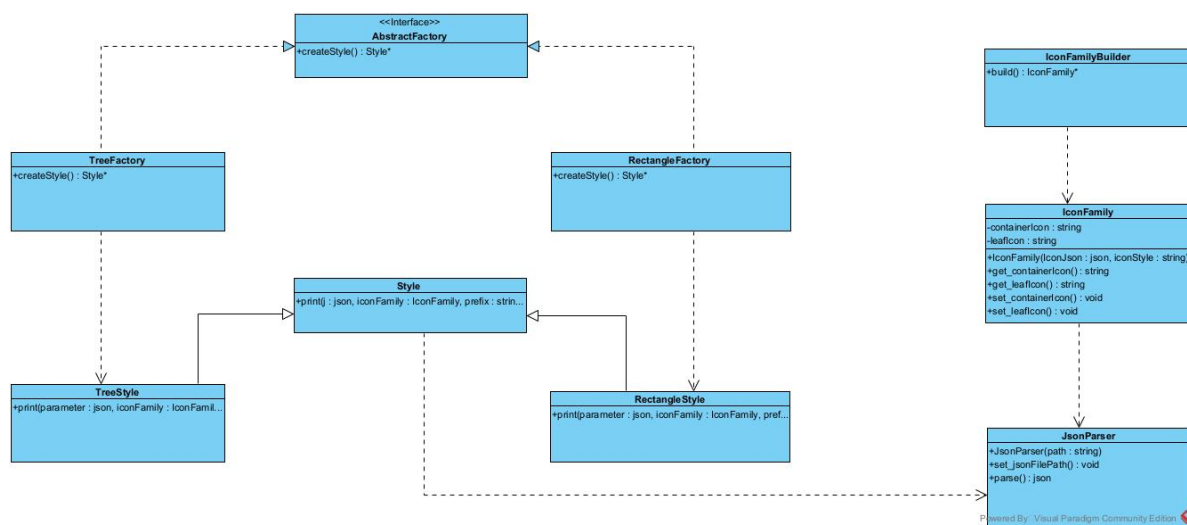


# FunnyJsonExplorer



- **AbstractFactory**: 提供接口，用于创建与特定样式相关的对象
- **TreeFactory**, **RectangleFactory**: 具体工厂，实现了 **AbstractFactory** 的接口，根据具体需求创建不同样式的对象
- **Style**: 定义抽象产品接口，具体样式类需要实现这个接口，用于打印特定格式的JSON数据
- **TreeStyle**, **RectangleStyle**: 实现 **Style** 接口，提供具体的打印方式
  - **TreeStyle** 实现了 `print` 方法，按树形样式打印JSON数据
  - **RectangleStyle** 实现了 `print` 方法，并增加了 `printBox` 和 `generatePadding` 方法，按矩形样式打印JSON数据
- **IconFamily**: 管理图标相关信息，包括中间节点图标和叶子节点图标
- **IconFamilyBuilder**: 采用建造者模式构建 **IconFamily** 对象
- **JsonParser**: 解析JSON文件，返回JSON对象。通过文件路径读取并解析JSON数据，供其他类使用

## 设计模式及作用：

- 工厂方法模式：

具体产品 **TreeStyle**, **RectangleStyle** 分别由具体工厂 **TreeFactory**, **RectangleFactory** 生产，工厂对象向上提供创建产品的接口。

- 抽象工厂模式：

抽象工厂 **AbstractFactory** 分别由两种不同的具体工厂 **TreeFactory**, **RectangleFactory** 实现，每种具体工厂只返回一种特定种类的产品。

抽象产品 **Style** 由两种不同类型的具体产品 **TreeStyle**, **RectangleStyle** 实现，提供绘制各自风格的方法。

- 建造者模式：

**IconFamilyBuilder** 将复杂的 **IconFamily** 类对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

- 组合模式：

虽然未在类图中表现，产品 `style` 的 `print` 方法将中间节点和叶子节点组合成树形模式进行处理。