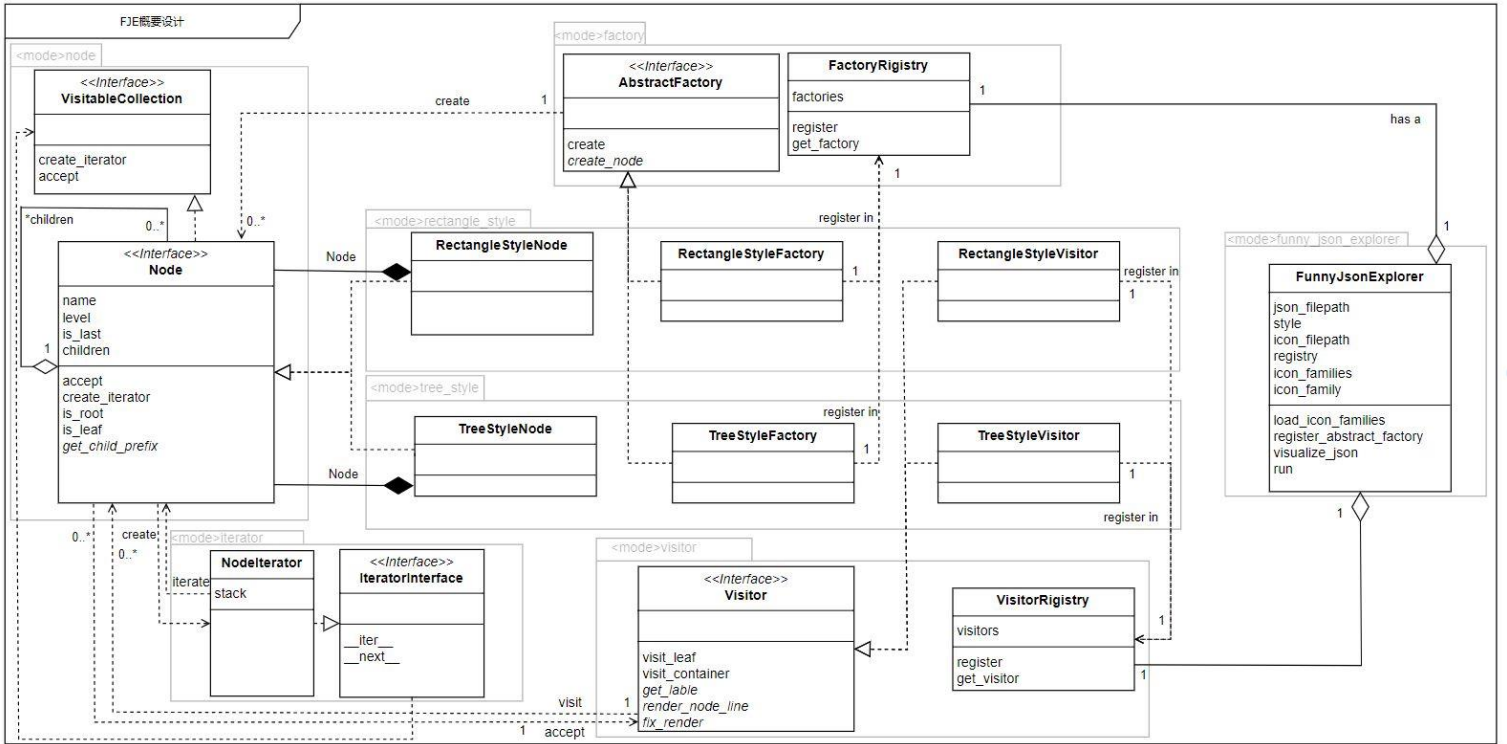


（建议使用 PDF 的书签进行跳转以提升阅读体验，图片可放大）

1. 领域模型

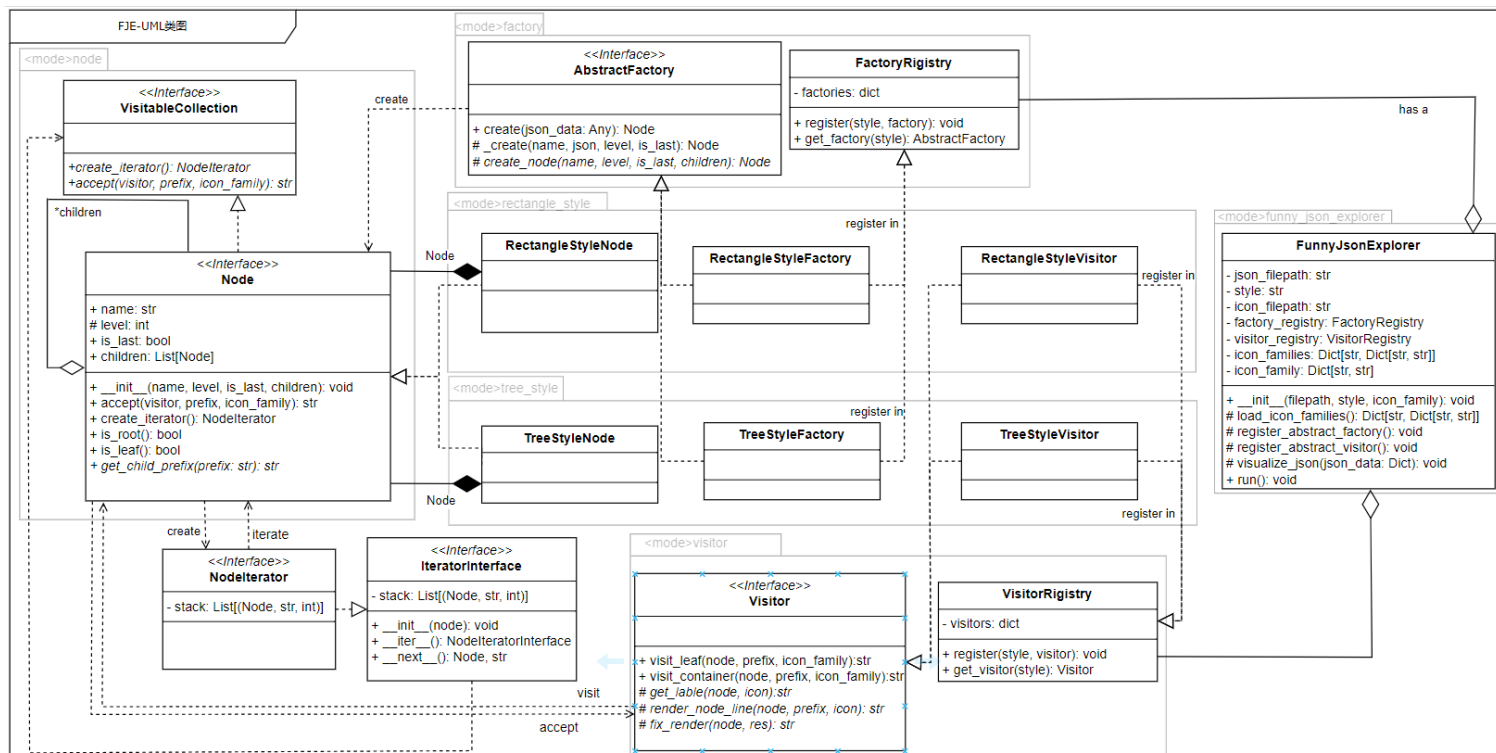
- 1、由于 UML 与开发语言无关，隐藏了开发语言的细节，所以尽管代码实现中存在相当大量的用于表示私有的或约定为受保护的变量或函数的单/双下划线前缀，但在领域模型（以及后文展示的 UML 类图）中隐去了这些用于表示访问权限的前缀。（单下划线前缀约定为 protected，双下划线前缀为 private）
- 2、参考老师的课堂示例，隐去了一些不重要的函数例如“children_iter”等，也根据惯例，隐去了构造函数；也根据惯例，隐去了子类实现的行为与父类（抽象类）的虚函数相关的函数。
- 3、由于当父类是抽象类时，子类与父类的关系既有继承关系又有实现关系，为了表现迭代器模式和访问者模式的接口的特点，我突出了实现关系。



（若看不清，请放大文档，或者详见公开可访问的 Github repo URL 对应仓库的文件夹目录“4 其他”；下图同理）

2. UML 类图

为了处理一些函数签名过长的情况，我选择将签名过长的函数的形参的数据类型略去。与上文同理隐去了单/双下划线前缀，除非隐去前缀会导致函数名重复。也根据惯例，隐去了构造函数，除非这个构造函数初始化了成员变量；也根据惯例，隐去了子类实现的行为与父类（抽象类）的虚函数相符的函数。



3. 说明：使用的设计模式及作用

3.1 迭代器模式

VisitableCollection、Node、IteratorInterface、NodeIterator 共同实现了迭代器模式。

具体而言，`VisitableCollection` 是迭代器模式的集合接口，有 `create_iterator` 函数返回迭代器（接口）；`IteratorInterface` 是迭代器接口，声明了本程序中遍历集合所需的相关操作（为了代码简洁性、可读性，未引入本程序未使用的其他操作例如 `reset`）；`Node` 是具体集合，实现了 `create_iterator` 函数，此函数返回具体迭代器 `NodeIterator`；`NodeIterator` 是具体迭代器，实现了（渲染时）树的后序深度优先搜索这一遍历方式，并且每次迭代可以返回与遍历相关的结果。

在本程序中，迭代器模式作用是隐藏了（渲染时）树的后序深度优先搜索这一遍历方式的复杂性，将这一遍历方式封装到了 `NodeIterator` 这一迭代器中；并且提升了程序的可扩展性，例如以后可以实现 `IteratorInterface` 的其他子类以实现其他遍历方式例如树的广度优先搜索。

3.2 访问者模式

VisitableCollection、Node、Visitor、TreeStyleVisitor、RectangleStyleVisitor 共同实现了访问者模式。

具体而言，`VisitableCollection` 是访问者模式的元素接口（也即 `VisitableCollection` 同时在迭代器模式和访问者模式担任相应的不同的角色）；`Node` 是访问者模式的具体元素，其中作为叶子节点（leaf）的 `Node` 是一种具体元素，作为中间节点（container）的 `Node` 是另一种具体元素；`Visitor` 是访问者模式的访问者接口，有 `visit_leaf` 和 `visit_container` 两个访问者方法，对应不同的具体元素；`TreeStyleVisitor` 和 `RectangleStyleVisitor` 是具体访问者，为两类具体元素（叶子节点、中间节点）实现渲染行为的不同版本（树形渲染版本、矩形渲染版本）。

在本程序中，访问者模式作用是树的两类具体元素（叶子节点、中间节点）提供相同操作（渲染）的变体（叶子节点渲染、中间节点渲染），并且还可以通过不同的具体访问者实现不同版本的操作（树形渲染、矩形渲染）；以及将渲染操作抽取到访问者类中，使代码更加模块化，保证了单一职责原则。

4. 运行截图（完成了题目所需功能）

详见文件夹“运行截图”，截图内容与上一次作业完全相同，这证明了我实现上文所述的相关设计模式时，仍能保证结果的正确性。

注意需要依赖库 toml 等，因此建议在 conda 命令行中激活了合适的环境再执行

我的截图的指令是利用 pycharm 这一 IDE 内置的命令行终端执行的，这样做是因为

- 1、unicode 制表符在 windows 内置命令行上的视觉效果不好、对齐不良。
- 2、考虑到老师在演示示例时，也是直接使用 IDE 内置的命令行
- 3、题目要求本程序是命令行界面小工具，而对于截图说明的要求中并未强调不可使用 IDE 内置的命令行

5. 源代码库：公开可访问的 Github repo URL

https://github.com/Xiebt3/21307352_design_pattern_2.git

（注：此网站当前为私有仓库，等到提交时间截止后才公开）