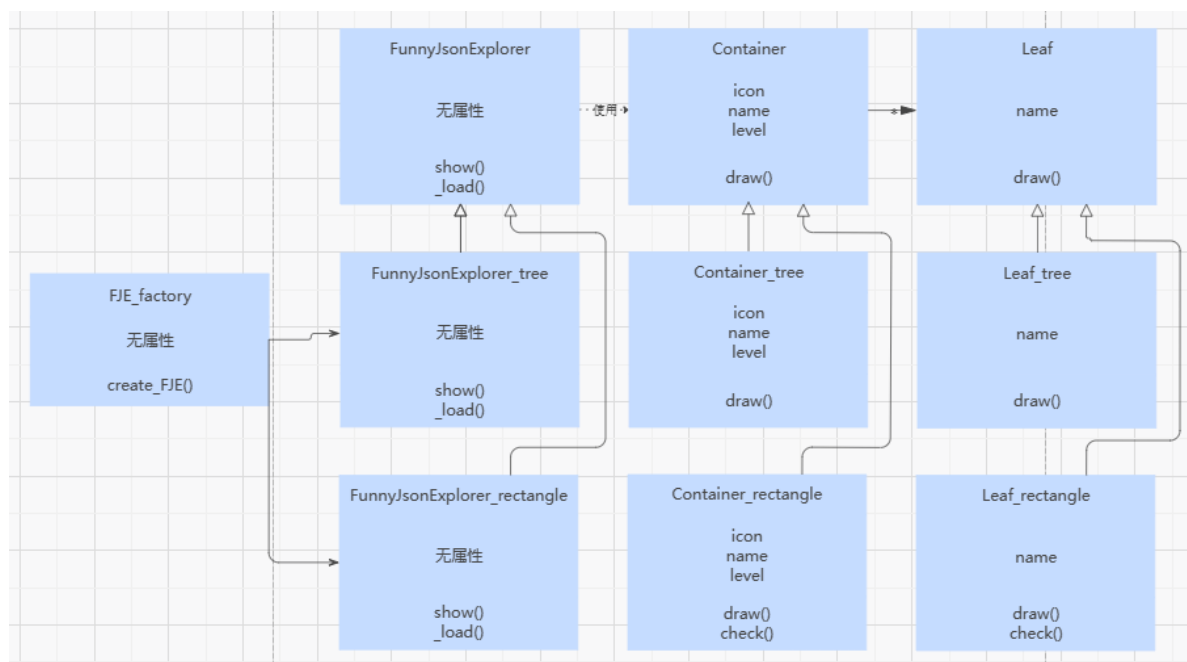


工厂方法模式

使用普通设计方法的情况下，每添加一种打印风格，就需要创建不同的打印函数，并在接收命令行的命令之后，通过条件语句来决定调用哪个打印函数。

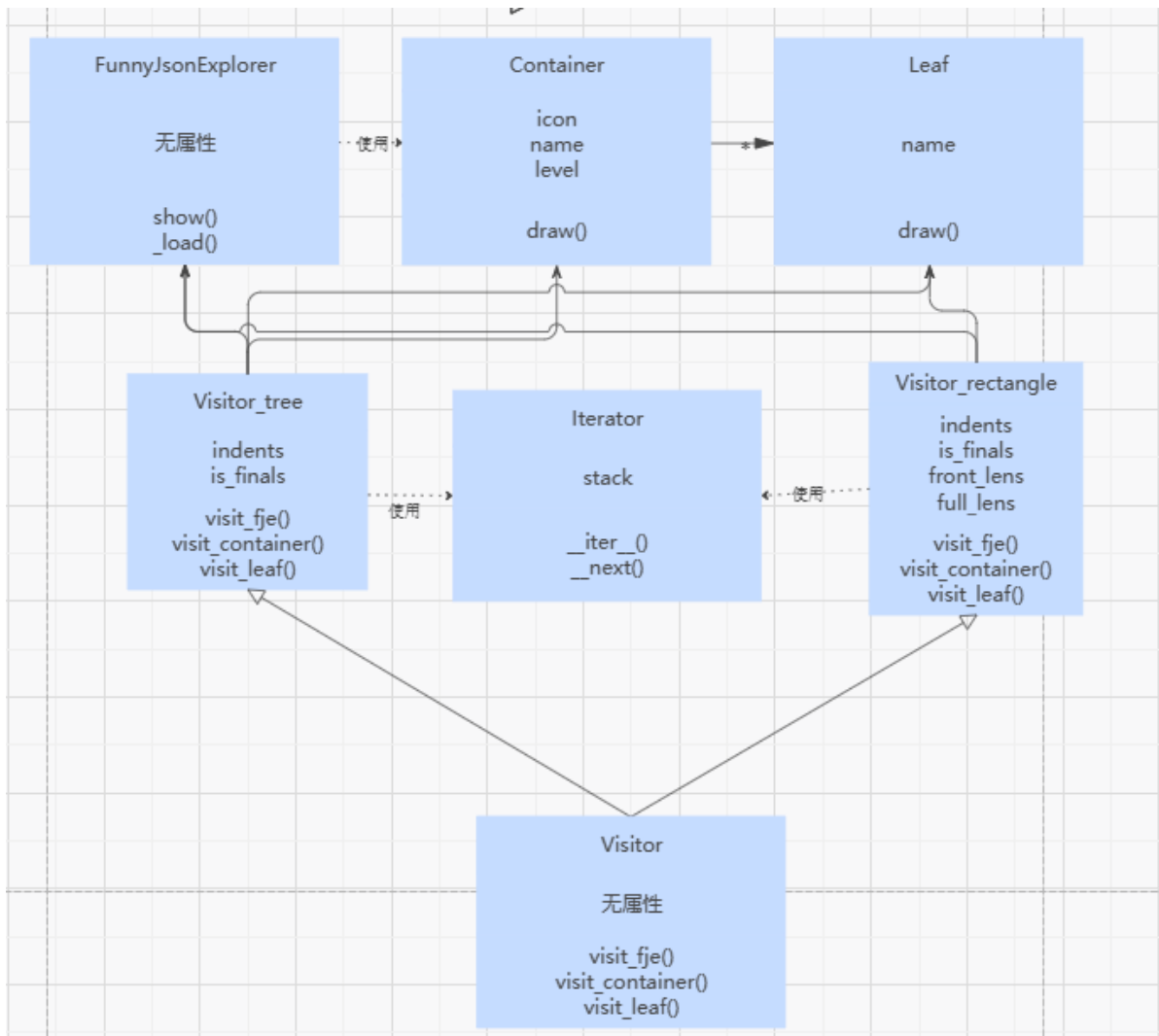
使用工厂方法模式的情况下，先创建父类，并为每个打印风格创建相应的子类；此外还创建一个工厂类，它会提供创建特定子类（特定风格）的接口。通过添加新子类的方式来实现新的打印风格。客户代码在接收命令行的命令之后，先调用工厂方法创建某风格对应的某子类，再使用这个子类处理JSON文件并打印，不需要关心这个子类是什么。所以添加新子类从而添加新风格前后，客户代码部分不需要改变。



迭代器模式+访问者模式

创建类Iterator来实现迭代器，并可以由此派生出各种子类来实现不同功能的迭代器。FJE中迭代器负责实现对JSON树形结构的遍历。

访问者负责将对类进行访问和操作的方法从类中独立出来，成为一个专门的访问者类Visitor，同样可以派生出不同的子类来实现不同的访问功能。为了辨别访问对象是那种类，使用双分派的方式，再每个被访问类中提供一个访问接口`accept()`，由`accept()`负责调用对应自身所属类的访问方法。FJE中由`show()`和`draw()`来提供上述接口，并由不同访问者类的子类来实现不同风格的打印。



使用命令：

```
fje -f test.json -s tree -i diamond
fje -f test.json -s tree -i spade
fje -f test.json -s rectangle -i diamond
fje -f test.json -s rectangle -i spade
```

运行截图（使用工厂方法）：

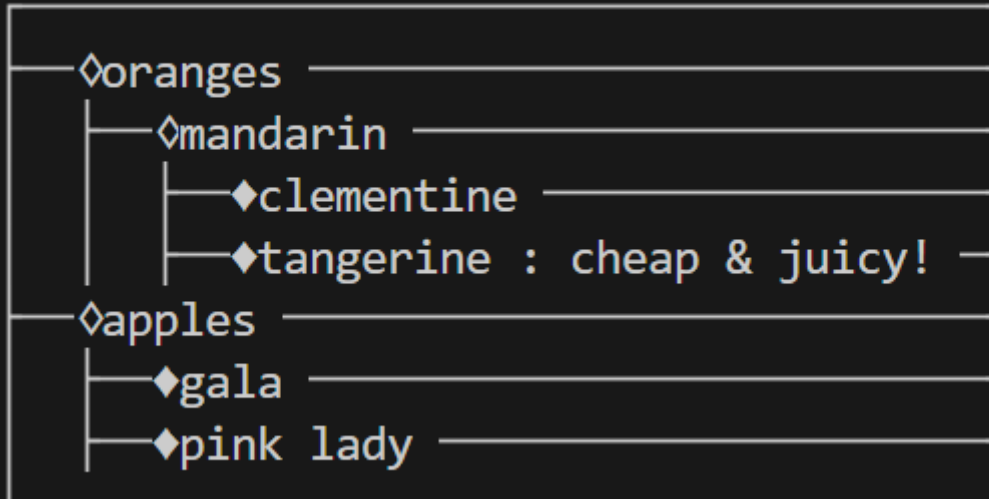
```
fje> fje -f test.json -s tree -i diamond
```

```
.
├── ◇oranges
│   ├── ◇mandarin
│   │   ├── ◇clementine
│   │   └── ◇tangerine : cheap & juicy!
└── ◇apples
    ├── ◇gala
    └── ◇pink lady
```

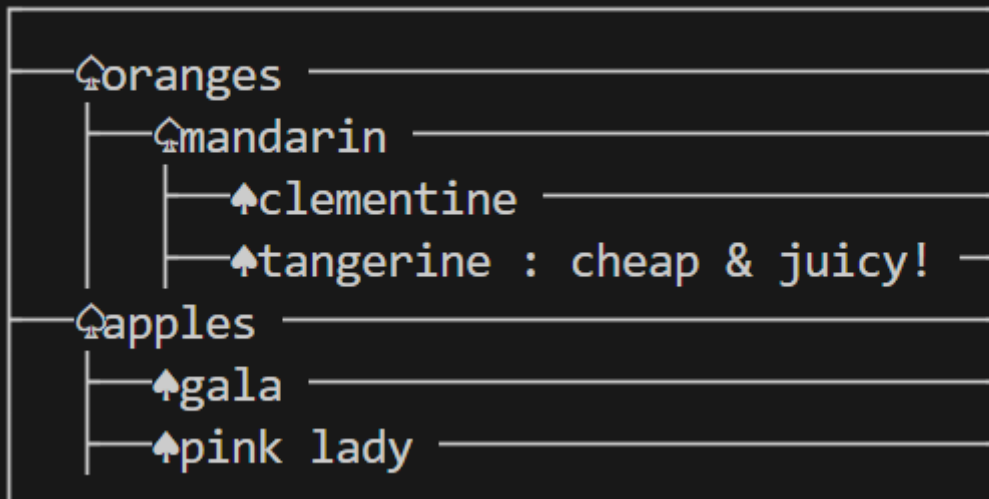
```
fje> fje -f test.json -s tree -i spade
```

```
.
├── ♠oranges
│   ├── ♠mandarin
│   │   ├── ♠clementine
│   │   └── ♠tangerine : cheap & juicy!
└── ♠apples
    ├── ♠gala
    └── ♠pink lady
```

```
fje> fje -f test.json -s rectangle -i diamond
```



```
fje> fje -f test.json -s rectangle -i spade
```



Github repo URL:

<https://github.com/zibo-lin/Funny-Json-Explorer.git>