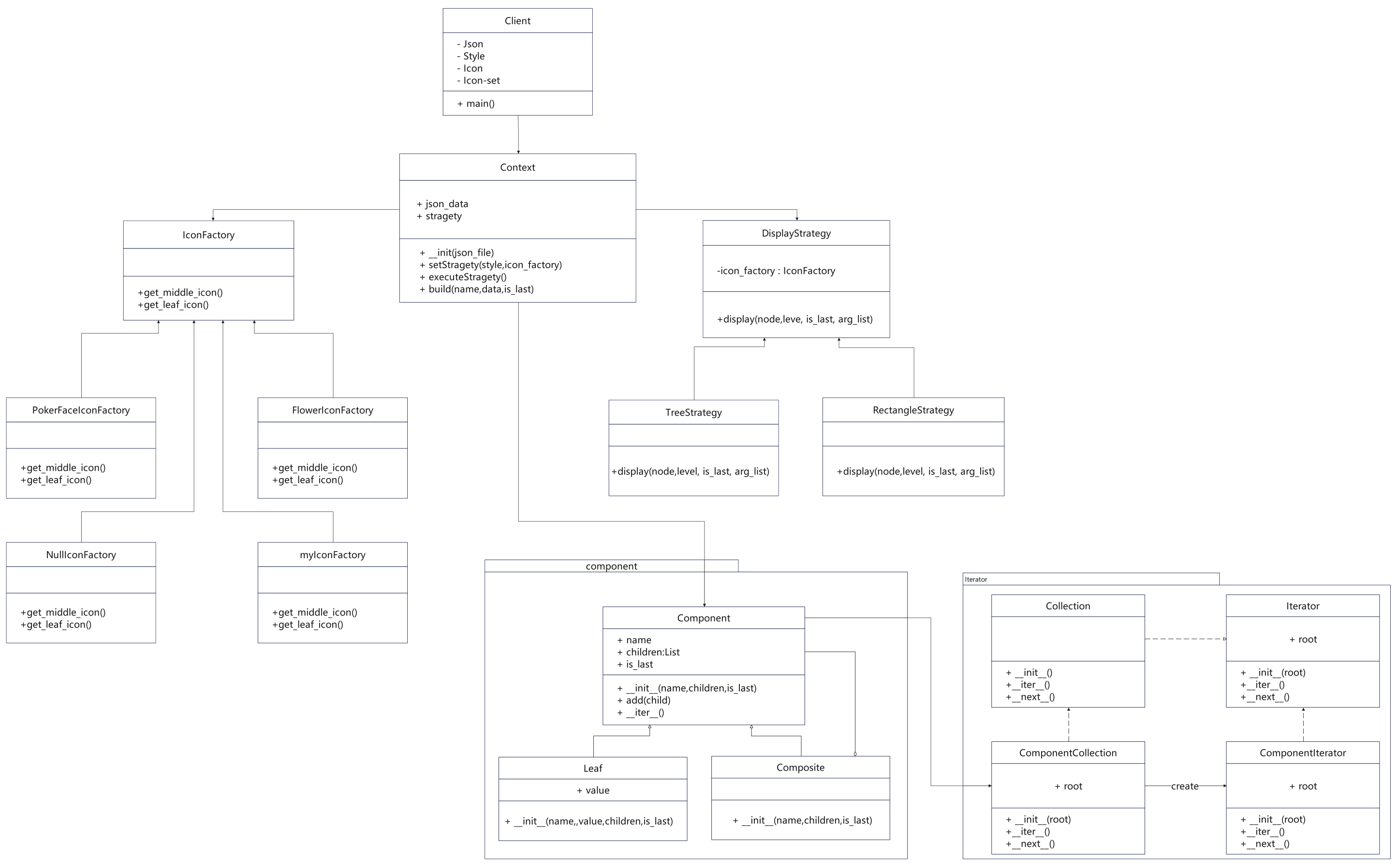
1 FJE 进阶作业要求

对已有的FJE实现进行设计重构

改用迭代器+访问者模式，或者迭代器+策略模式

2 类图



当然！以下是对每个部分的作用和所用设计模式的详细说明：

3 说明

3.1 `Collection` 类、 `ComponentCollection` 类、 `ComponentIterator` 类和 `Iterator` 类

设计模式：

- **迭代器模式（Iterator Pattern）**：通过定义 `__iter__` 和 `__next__` 方法， `Collection` 类和其子类实现了迭代器模式，使用户可以遍历集合中的元素而不需要了解集合的内部实现。
- `Iterator` 类是一个抽象基类，定义了迭代器的基本接口。
- `ComponentIterator` 类实现了迭代器接口，用于遍历组件树。

```python
class ComponentIterator(Iterator):
    def __init__(self, root):
        self.stack = [(root, 0, False, False)]  # (node, level, is_last,is_top)

    def __iter__(self):
        return self

    def __next__(self):
        if not self.stack:
            raise StopIteration

        node, level, is_last, is_top = self.stack.pop()
        if isinstance(node, Composite):
            for i, child in enumerate(reversed(node.children)):
                self.stack.append((child, level + 1, i == 0, i == len(node.children) - 1))
        return node, level, is_last, is_top
```

3.2 `Component` 类、 `Leaf` 类和 `Composite` 类

设计模式：

- **组合模式（Composite Pattern）**： `Component` 类及其子类（ `Leaf` 和 `Composite` ）实现了组合模式，使得树形结构中的叶子节点和组合节点能够统一处理。通过这种模式，树形结构中的每个节点都可以被视为 `Component` 。

```python
class Component:
    def __init__(self, name, children=None, is_last=0):
        self.name = name
        self.children = children if children is not None else []
        self.is_last = is_last

    def add(self, child):
        self.children.append(child)

    def __iter__(self):
        return iter(ComponentCollection(self))


class Leaf(Component):
    def __init__(self, name, value, is_last=0):
        super().__init__(name, children=[], is_last=is_last)
        self.value = value


class Composite(Component):
    def __init__(self, name, is_last=0):
        super().__init__(name, children=[], is_last=is_last)
```

3.3 `IconFactory` 类及其子类

设计模式：

- **工厂模式（Factory Pattern）：** `IconFactory` 类及其子类实现了工厂模式，根据不同的需求提供不同类型的图标。

3.4 `DisplayStrategy` 类及其子类

设计模式：

- **策略模式（Strategy Pattern）：** `DisplayStrategy` 类及其子类实现了策略模式，根据不同的显示策略来显示节点信息。
- `DisplayStrategy` 类是一个抽象基类，定义了显示节点的方法。
- 其子类（如 `TreeStyle` 和 `RectangleStyle` ）实现了具体的显示逻辑。

```python
class DisplayStrategy(ABC):
    def __init__(self, icon_factory):
        self.icon_factory = icon_factory

    @abstractmethod
    def display(self, node, level, is_last, arg_list):
        pass
```

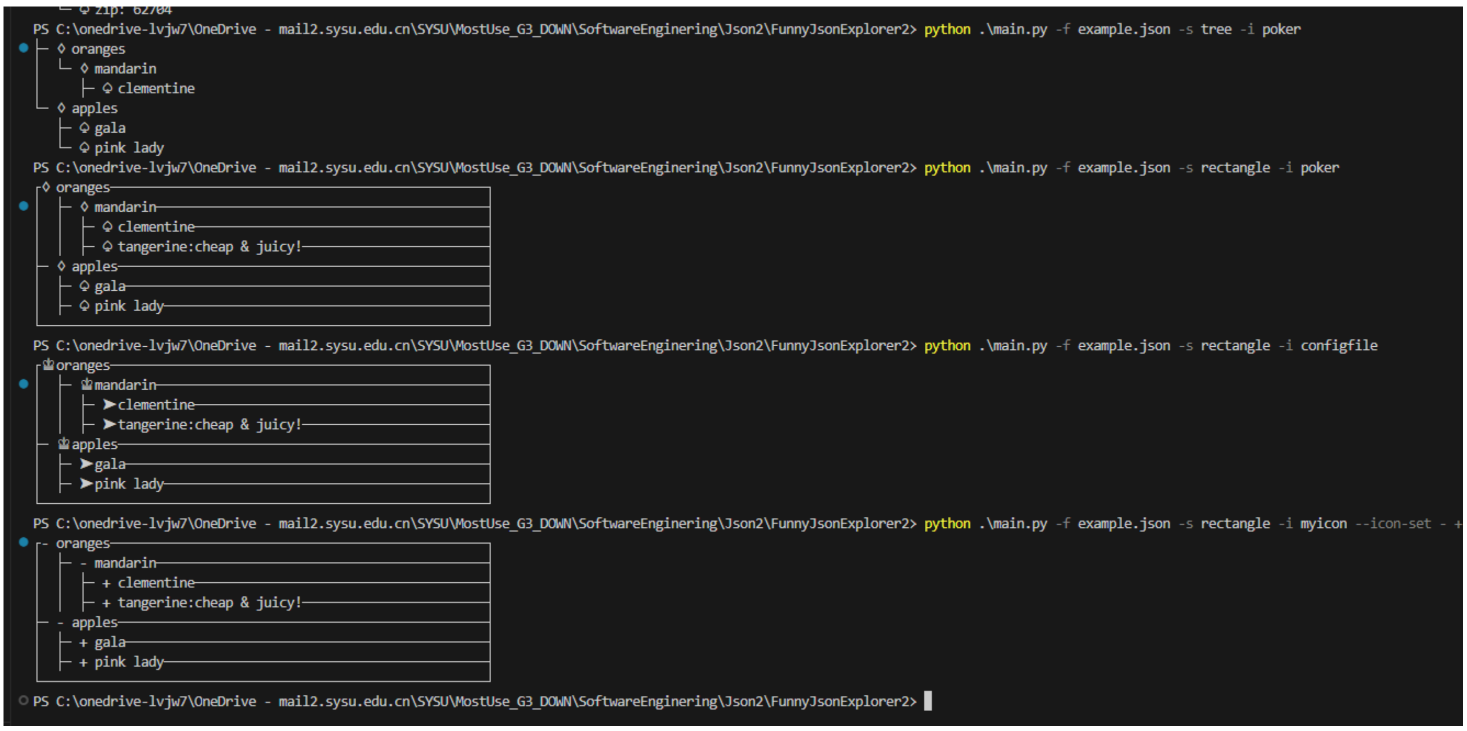
3.5 6. `Context` 类

设计模式：

- **策略模式（Strategy Pattern）：** 通过设置不同的显示策略和图标工厂， `Context` 类实现了策略模式，允许在运行时改变显示方式。
- **迭代器模式（Iterator Pattern）：** `executeStrategy` 中使用迭代器遍历component中的每一个节点。
- `Context` 类负责解析 JSON 文件并构建组件树。
- 它使用策略模式设置不同的显示策略和图标工厂，然后执行相应的显示逻辑。

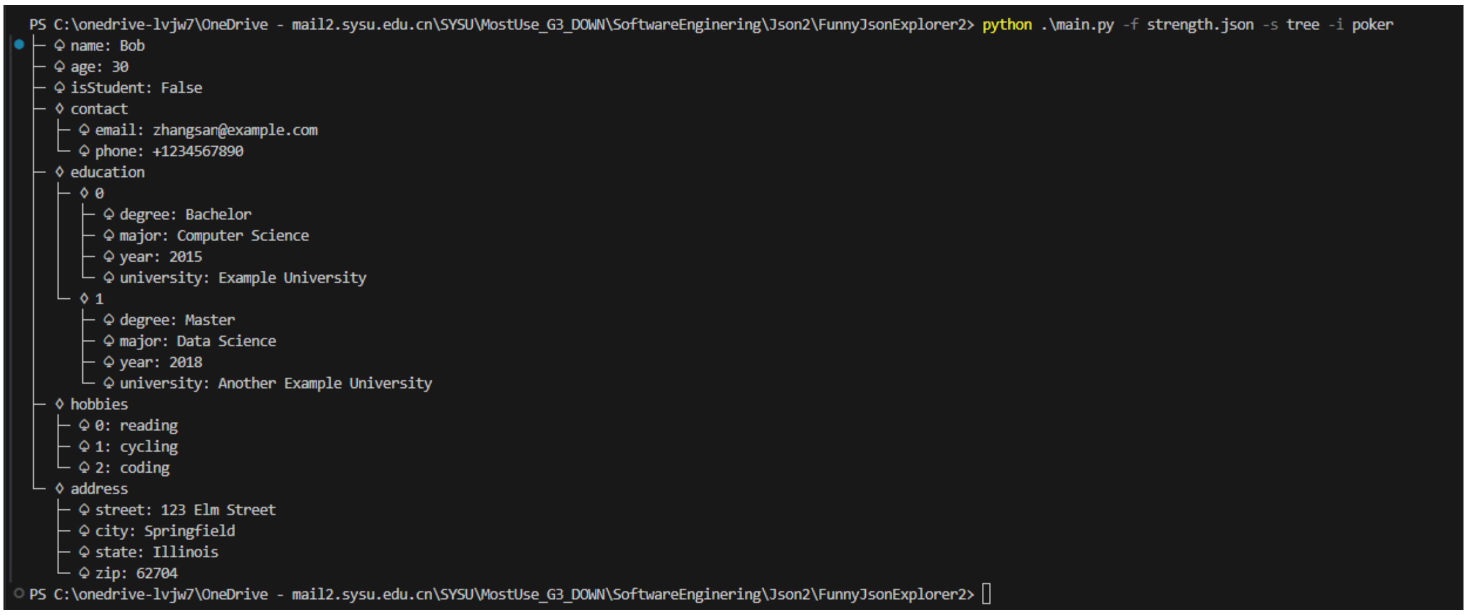
```python
class Context:
    def __init__(self, json_file):
        with open(json_file, 'r') as f:
            self.json_data = json.load(f)
        self.strategy = None

    def setStrategy(self, style, icon_factory):
        self.strategy = style(icon_factory)
        return self

    def executeStrategy(self):
        root = self.build('root', self.json_data)
        arg_list = []
        for node, level, is_last, is_top in root:  # 使用迭代器遍历
            arg_list.insert(0, is_top)
            self.strategy.display(node, level, is_last, arg_list)
        self.strategy.displayEnd()

    def build(self, name, data, is_last=False):
        if isinstance(data, dict) or isinstance(data, list):
            composite = Composite(name, is_last=is_last)
            if isinstance(data, dict):
                for i, (key, value) in enumerate(data.items()):
                    child_is_last = i == len(data) - 1
                    composite.add(self.build(key, value, is_last=child_is_last))
            elif isinstance(data, list):
                for i, item in enumerate(data):
                    child_is_last = i == len(data) - 1
                    composite.add(self.build(str(i), item, is_last=child_is_last))
            return composite
        else:
            return Leaf(name, data, is_last=is_last)
```

4 功能展示

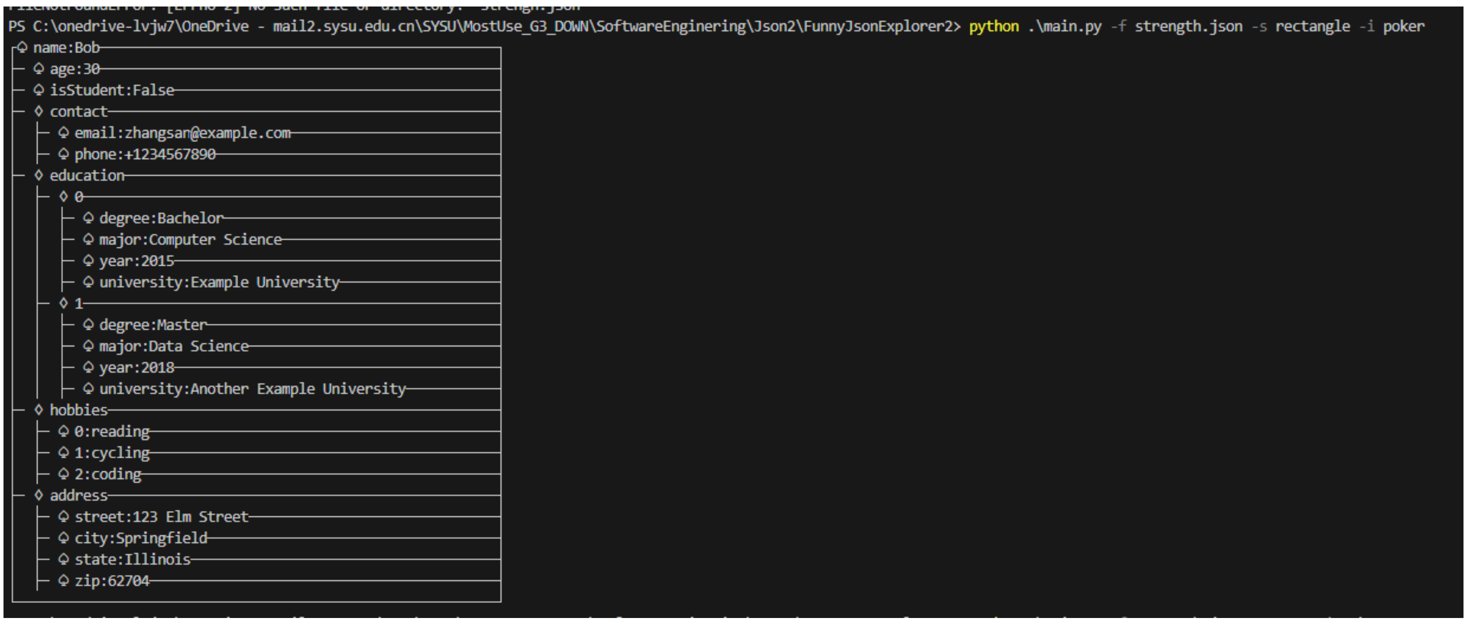


```
   └─ ♤ zip: 62704
PS C:\onedrive-lvjw7\OneDrive - mail2.sysu.edu.cn\SYSU\MostUse_G3_DOWN\SoftwareEnginering\Json2\FunnyJsonExplorer2> python .\main.py -f example.json -s tree -i poker
├─ ♢ oranges
│  └─ ♢ mandarin
│     ├─ ♤ clementine
└─ ♢ apples
   ├─ ♤ gala
   └─ ♤ pink lady
PS C:\onedrive-lvjw7\OneDrive - mail2.sysu.edu.cn\SYSU\MostUse_G3_DOWN\SoftwareEnginering\Json2\FunnyJsonExplorer2> python .\main.py -f example.json -s rectangle -i poker
┌♢ oranges───────────────────────────────────────┐
│  ├─ ♢ mandarin─────────────────────────────────┤
│  │  ├─ ♤ clementine────────────────────────────┤
│  │  ├─ ♤ tangerine:cheap & juicy!──────────────┤
├─ ♢ apples──────────────────────────────────────┤
│  ├─ ♤ gala─────────────────────────────────────┤
│  ├─ ♤ pink lady────────────────────────────────┤
└────────────────────────────────────────────────┘

PS C:\onedrive-lvjw7\OneDrive - mail2.sysu.edu.cn\SYSU\MostUse_G3_DOWN\SoftwareEnginering\Json2\FunnyJsonExplorer2> python .\main.py -f example.json -s rectangle -i configfile
┌♕oranges────────────────────────────────────────┐
│  ├─ ♕mandarin──────────────────────────────────┤
│  │  ├─ ➤clementine─────────────────────────────┤
│  │  ├─ ➤tangerine:cheap & juicy!───────────────┤
├─ ♕apples───────────────────────────────────────┤
│  ├─ ➤gala──────────────────────────────────────┤
│  ├─ ➤pink lady─────────────────────────────────┤
└────────────────────────────────────────────────┘

PS C:\onedrive-lvjw7\OneDrive - mail2.sysu.edu.cn\SYSU\MostUse_G3_DOWN\SoftwareEnginering\Json2\FunnyJsonExplorer2> python .\main.py -f example.json -s rectangle -i myicon --icon-set - +
┌- oranges───────────────────────────────────────┐
│  ├─ - mandarin─────────────────────────────────┤
│  │  ├─ + clementine────────────────────────────┤
│  │  ├─ + tangerine:cheap & juicy!──────────────┤
├─ - apples──────────────────────────────────────┤
│  ├─ + gala─────────────────────────────────────┤
│  ├─ + pink lady────────────────────────────────┤
└────────────────────────────────────────────────┘

PS C:\onedrive-lvjw7\OneDrive - mail2.sysu.edu.cn\SYSU\MostUse_G3_DOWN\SoftwareEnginering\Json2\FunnyJsonExplorer2>
```

完整性测试:



```
PS C:\onedrive-lvjw7\OneDrive - mail2.sysu.edu.cn\SYSU\MostUse_G3_DOWN\SoftwareEnginering\Json2\FunnyJsonExplorer2> python .\main.py -f strength.json -s tree -i poker
├─ ♤ name: Bob
├─ ♤ age: 30
├─ ♤ isStudent: False
├─ ♢ contact
│  ├─ ♤ email: zhangsan@example.com
│  └─ ♤ phone: +1234567890
├─ ♢ education
│  ├─ ♢ 0
│  │  ├─ ♤ degree: Bachelor
│  │  ├─ ♤ major: Computer Science
│  │  ├─ ♤ year: 2015
│  │  └─ ♤ university: Example University
│  └─ ♢ 1
│     ├─ ♤ degree: Master
│     ├─ ♤ major: Data Science
│     ├─ ♤ year: 2018
│     └─ ♤ university: Another Example University
├─ ♢ hobbies
│  ├─ ♤ 0: reading
│  ├─ ♤ 1: cycling
│  └─ ♤ 2: coding
└─ ♢ address
   ├─ ♤ street: 123 Elm Street
   ├─ ♤ city: Springfield
   ├─ ♤ state: Illinois
   └─ ♤ zip: 62704
PS C:\onedrive-lvjw7\OneDrive - mail2.sysu.edu.cn\SYSU\MostUse_G3_DOWN\SoftwareEnginering\Json2\FunnyJsonExplorer2>
```



```
PS C:\onedrive-lvjw7\OneDrive - mail2.sysu.edu.cn\SYSU\MostUse_G3_DOWN\SoftwareEnginering\Json2\FunnyJsonExplorer2> python .\main.py -f strength.json -s rectangle -i poker
┌♤ name:Bob─────────────────────────────────────────────┐
├─ ♤ age:30─────────────────────────────────────────────┤
├─ ♤ isStudent:False────────────────────────────────────┤
├─ ♢ contact────────────────────────────────────────────┤
│  ├─ ♤ email:zhangsan@example.com──────────────────────┤
│  ├─ ♤ phone:+1234567890───────────────────────────────┤
├─ ♢ education──────────────────────────────────────────┤
│  ├─ ♢ 0───────────────────────────────────────────────┤
│  │  ├─ ♤ degree:Bachelor──────────────────────────────┤
│  │  ├─ ♤ major:Computer Science───────────────────────┤
│  │  ├─ ♤ year:2015────────────────────────────────────┤
│  │  ├─ ♤ university:Example University────────────────┤
│  ├─ ♢ 1───────────────────────────────────────────────┤
│  │  ├─ ♤ degree:Master────────────────────────────────┤
│  │  ├─ ♤ major:Data Science───────────────────────────┤
│  │  ├─ ♤ year:2018────────────────────────────────────┤
│  │  ├─ ♤ university:Another Example University────────┤
├─ ♢ hobbies────────────────────────────────────────────┤
│  ├─ ♤ 0:reading───────────────────────────────────────┤
│  ├─ ♤ 1:cycling───────────────────────────────────────┤
│  ├─ ♤ 2:coding────────────────────────────────────────┤
├─ ♢ address────────────────────────────────────────────┤
│  ├─ ♤ street:123 Elm Street───────────────────────────┤
│  ├─ ♤ city:Springfield────────────────────────────────┤
│  ├─ ♤ state:Illinois──────────────────────────────────┤
│  ├─ ♤ zip:62704───────────────────────────────────────┤
└───────────────────────────────────────────────────────┘
```