Lab2 语法分析 实验报告

本次实验需要完成语法分析，并按照语法树的层次打印语法树的节点，在终止节点加语法高亮（此处并不是真的加，只是通过加后缀）。下面简述完成过程和需要应用到的知识。

左递归

实验要求中需要我们将 exp 和 cond 改成左递归写法, 因为 antlr 可以自动处理左递归和优先级。

一个文法 G，若存在 P 通过一次或者多次推导得到 Pa，则称 G 是左递归的。左递归可以分为直接左递归和间接左递归，直接左递归通过一次推导就可以看出左递归，间接左递归需要多次推导才能看出左递归。

通过改写为左递归形式，写法看起来直观了很多。

报告语法错误

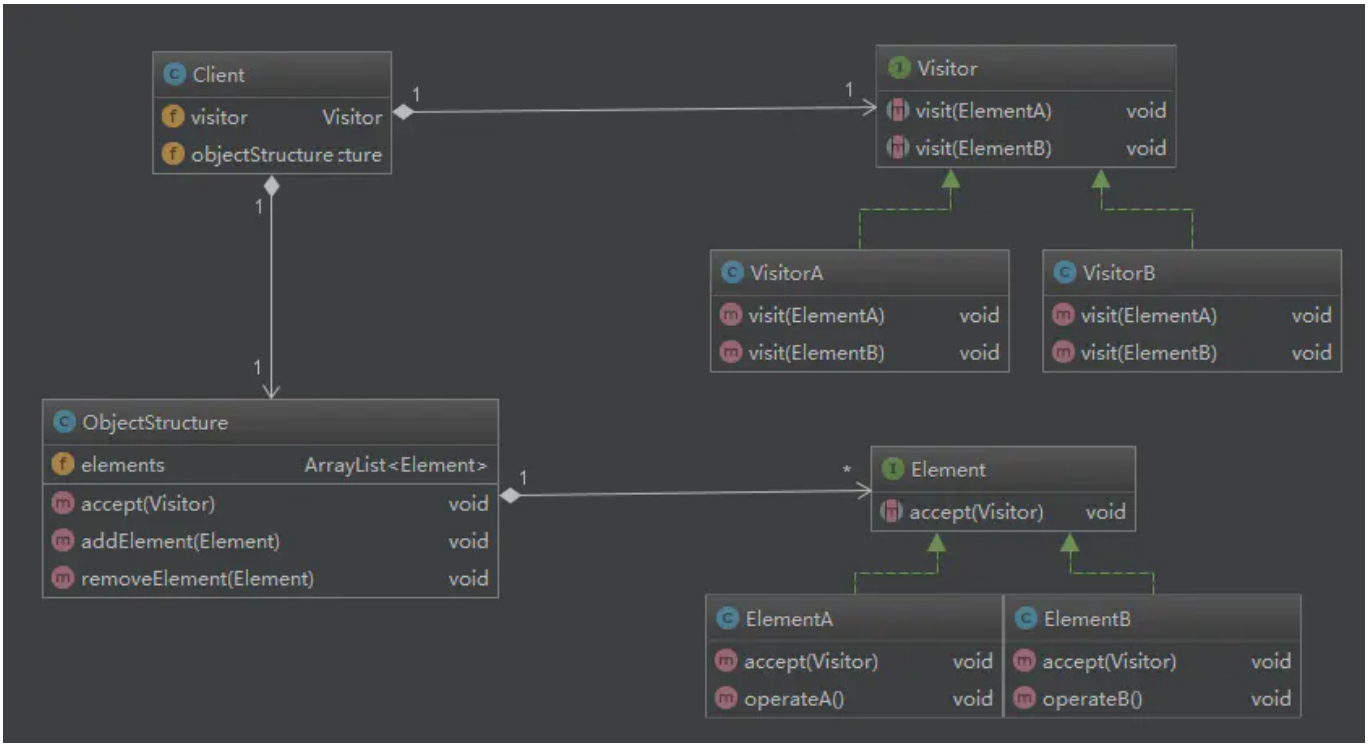
编写一个子类继承 BaseErrorListener，覆盖 syntaxError 方法，在里面输出信息即可。

当调用 parser 的 program 方法的时候会触发解析，如果有语法错误就能在此时输出信息。

同时 parser 提供了 API 来获得 error 的数量，以便于决定后续是否还要继续输出语法树。

输出语法树

使用访问者模式来完成对语法树元素的输出。



> 访问者模式：访问者模式是一种将数据操作和数据结构分离的设计模式。 Visitor：接口或者抽象类，定义了对每个 Element 访问的行为，它的参数就是被访问的元素，它的方法个数理论上与元素的个数是一样的，因此，访问者模式要求元素的类型要稳定，如果经常添加、移除元素类，必然会导致频繁地修改 Visitor 接口，如果出现这种情况，则说明不适合使用访问者模式。 ConcreteVisitor：具体的访问者，它

> 需要给出对每一个元素类访问时所产生的具体行为。 Element：元素接口或者抽象类，它定义了一个接受访问者（accept）的方法，其意义是指每一个元素都要可以被访问者访问。 ElementA、ElementB：具体的元素类，它提供接受访问的具体实现，而这个具体的实现，通常情况下是使用访问者提供的访问该元素类的方法。 ObjectStructure：定义当中所提到的对象结构，对象结构是一个抽象表述，它内部管理了元素集合，并且可以迭代这些元素提供访问者访问。

antlr 已经提供了一个默认的 visitor，我们需要自己写一个子类继承这个 visitor 类，来自定义遍历行为。

具体来说，需要覆盖 visitChildren 和 visitTerminalNode 这两个方法，分别定义对内部节点和终止节点的访问行为。

具体的做法就是先通过树高计算前缀空白的长度，然后根据格式输出语法树元素的内容，至于语法高亮，可以通过表驱动的方式来完成，在初始的时候用一个 map 记录每个终止节点元素对应的颜色。