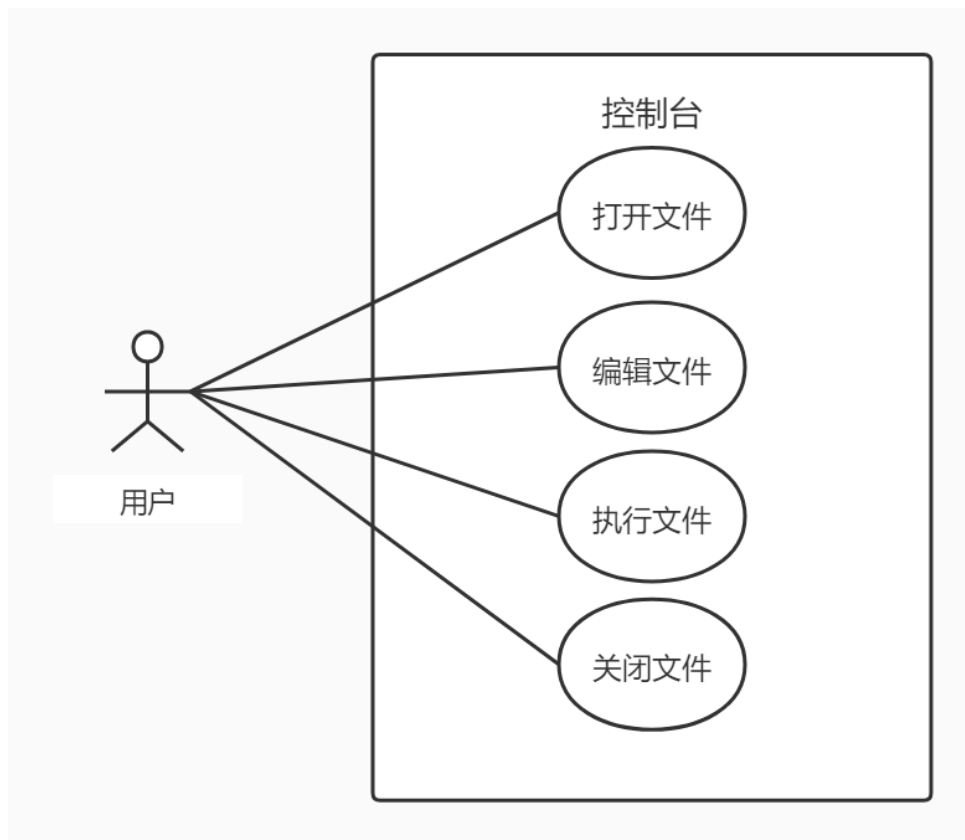


需求规格文档

小组：2017HYSE04

1 pyphy 解释器功能建模

该部分以用例图的形式展示了用户与产品之间的交互行为。对用户而言，控制台上主要有四个功能可以使用。打开文件：用户打开一个已存在的或者新的.py 文件。编辑文件：用户在控制台上修改已打开的文件。执行文件：在控制台上以本产品的解释器运行.py 文件。关闭文件：用户结束使用并退出程序。



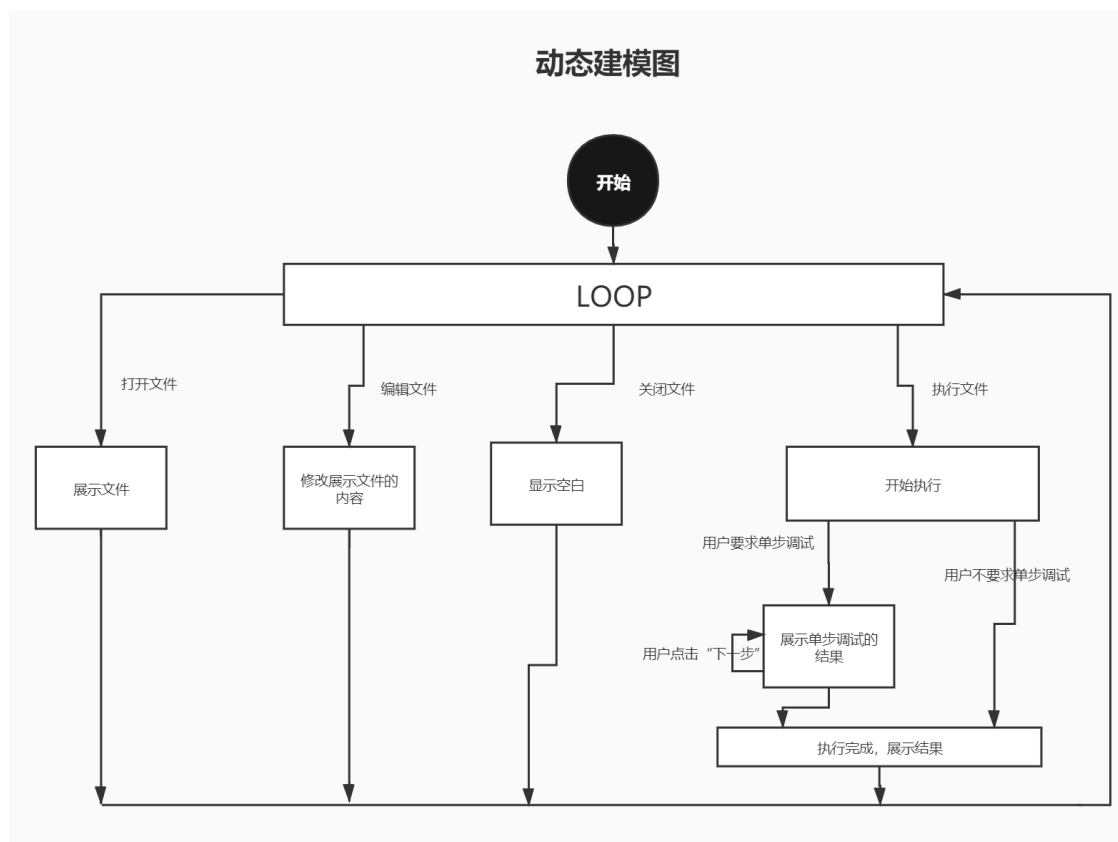
2 pyphy 解释器动态建模

本软件的状态主要有四种。

针对文件操作，会产生三种状态：1. 当用户要求打开文件，本软件展示文件的内容，并回到 LOOP 起点。2. 当用户编辑文件，本软件会展示用户编辑文件后的结果，并回到 LOOP 起点。3. 当用户关闭文件，本软件会显示空白，并回到 LOOP 起点。

针对执行操作，有两种可能的情况：1. 如果用户不要求展示中间结果，则直

接展示执行完文件的结果，并回到 LOOP 起点。2. 如果用户要求单步调试展示中间结果，则用户每点击一次“下一步”，软件的相应界面上就展示当前符号表的内容，直至文件执行完成，并回到 LOOP 起点。

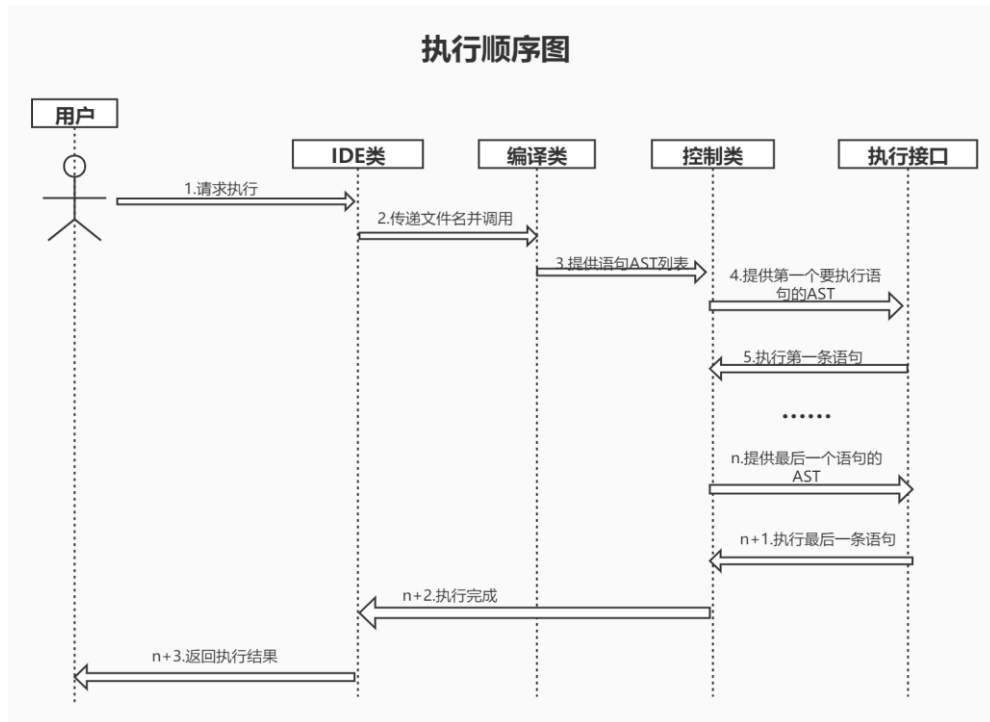


3 pyphy 解释器执行顺序图

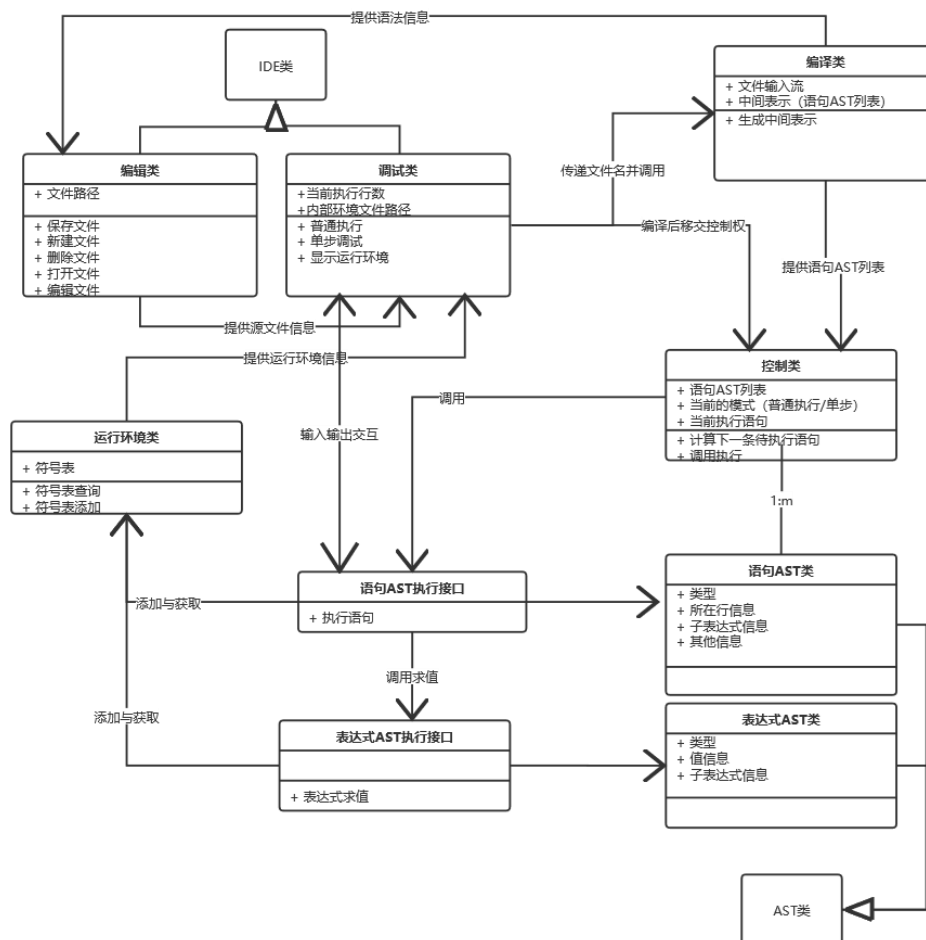
执行文件部分是本软件的核心部分。

当用户提交执行请求，IDE 类会将当前要执行的文件名传递给编译类，编译类接收到文件后，对其进行编译，得到由本文件中所生成的所有 AST 构成的一个 AST 数组。

编译类将该 AST 数组传递给控制类，控制类根据控制规则，决定将下一条要执行语句的 AST 传给执行接口，待执行接口执行完毕后，再将下一条要执行语句的 AST 传给执行接口。重复传递直至程序执行完毕。在执行过程中，如果用户要求单步调试，则每当执行完一条语句，就展示一次当前符号表里的内容。否则直



4 pyphy 解释器对象模型



以上是 **pyphy** 解释器系统的对象模型，以下使用 CRC 表说明对象模型的各项功能。

类	功能	与其他模块的交互
编辑类	对源文件的操作功能	<ol style="list-style-type: none"> 1. 文件操作于外部文件系统， 2. 可能需要获取编译类产生的语句信息
调试类	对源文件调用解释器，并在图形界面可视化运行结果与环境	<ol style="list-style-type: none"> 1. 调试类将源文件名传入并调用解释器类 2. 接受控制类的代码运行信息 3. 接受语句执行类提供的运行结果
编译类	根据源文件的文本，生成中间表示（表达式 AST 数组）	<ol style="list-style-type: none"> 1. 从测试类获得源文件名 2. 将处理好的中间表示提供给控制类
控制类	决定接下来要执行的语句	<ol style="list-style-type: none"> 1. 从编译类获取中间表示 2. 将需要执行的语句提供给执行类 3. 从执行类获取控制信息
运行环境类	对符号表进行初始化、查询、添加	<ol style="list-style-type: none"> 1. 和语句执行接口在变量的绑定上提供交互
表达式执行接口	对表达式的 AST 进行求值	<ol style="list-style-type: none"> 1. 将表达式的求值结果提供给语句执行类
语句执行接口	根据语句进行相应的执行动作	<ol style="list-style-type: none"> 1. 从表达式执行类获取语句执行结果 2. 从控制类获取需要执行的语句 3. 将语句的执行结果反馈给调试类