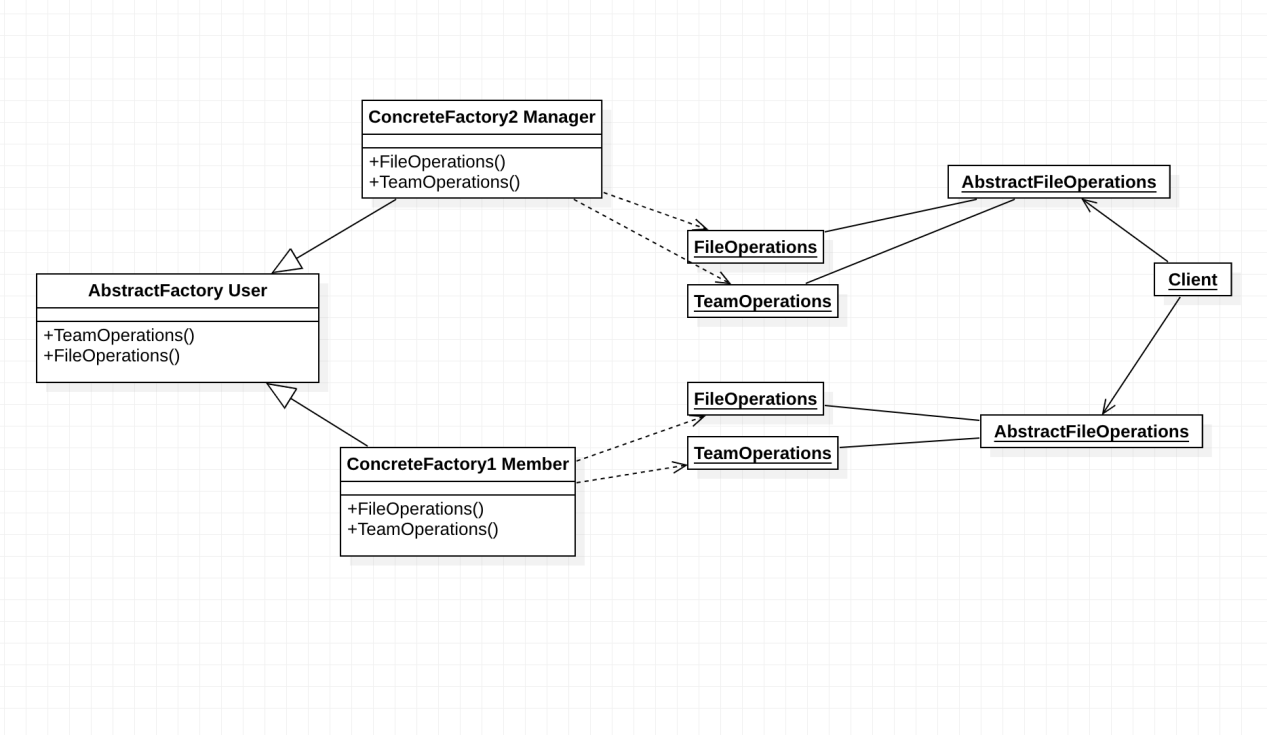
使用抽象工厂设计模式创建一组相关的对象Member和Manager；

Member和Manager中的文档操作和团队操作有不同权限；由此，Member中的文档操作和团队操作和Manager中的文档操作和团队管理就可以一一对应，这样Member和Manager就成了一组相关对象，采用统一的方式处理，在代码中创建的具体类型可以取决于调用参数。



使用中介器设计模式完成Member和Manager关于团队管理/操作的通信和交互。

Member和Manger之间并不直接产生引用，而是有一个指向中介的引用，该引用对应于中介器，所有消息均发送至中介：Member可以向中介发送请求加入团队的请求，中介可以持有已创建团队的引用，通过这些引用向Manager发送消息；Manager可以向中介发送请求邀请Member加入团队的请求，中介可以持有已注册用户的引用，通过这些引用向Member发送消息，使得通信成为双向。

