# 测试/重构的线上练习

6月18日-6月19日

代码的重构有利于更好地读懂和维护代码

1. 避免子程序过长， 繁琐的部分可以通过调用函数实现对子程序的简化

2. 对于重复的代码直接转到一个函数里面，减少代码不必要的冗余，也方便后续的统一修改

3. 去掉承担过少的职责的类和避免承担过多职责的类，类与类之间的关系最好与实际应用相贴合

4. if括号里面的条件判断语句如果过长可以考虑抽出来单独写一个返回值为布尔类型的函数进行条件判断

5. 尽量避免使用全局变量，防止出现名字的冲突与混淆

6. 没有成员方法的，不包含自定义的析构函数和拷贝构造函数，非静态成员不是指向成员的指针不需要用到类，用结构体即可

7. C++可以用断言来进行软件测试，通过判断某个子程序执行的结果是否与预期相符合来判断程序是否有问题

8. 白盒测试需要软件开发者是自己设计单元测试代码并执行，包括代码审查和继承测试等，进行单元测试时有不同的准则，包括子程序覆盖，语句覆盖，分支覆盖，条件覆盖，还有更严苛的条件、分支覆盖，组合条件覆盖

9. 黑盒测试最重要的是测试用例，测试人员无需了解程序内部的代码实现，不依赖程序内部设计，可以从用户角度出发进行测试，如题库的在线判题功能。在设计测试案例时，要全面考虑，防止遗漏。