

===

- 单元测试
- 后端测试
- 前端测试
- 集成测试
  - 测试目的
  - 测试方法
    - 功能性测试
      - 边界测试
      - 兼容性测试

## 单元测试

运行环境：Win10 操作系统，Intel Core i7-8750H @2.20GHZ 内存 8G

## 后端测试

运行环境：Win10 操作系统，Intel Core i7-8750H @2.20GHZ 内存 8G

对于我的程序，后端测试所需要做的内容主要是检验日程排列的正确性，以及对数据库进行读写的正确性。

这两部分都可以通过白盒测试来实现。

通过测试发现，当出现两个用户对数据库进行读写时，可能会出现与期望结果不一致的情况。通过在数据库操作时对其加锁来实现。

## 前端测试

运行环境：Win10 操作系统，Intel Core i7-8750H @2.20GHZ 内存 8G

前端测试的任务与后端测试有很大的不同，后端测试主要考虑了算法的正确性，而前端主要的测试目标是，对于用户的各种可能的输入（包括恶意的输入）都能给出一个易于理解和接受的结果。

## 集成测试

运行环境：Win10 操作系统，Intel Core i7-8750H @2.20GHZ 内存 8G

# 测试目的

在单元测试中，我们验证了程序各个模块的正确性。在此基础上，将所有模块按照设计要求组装成为完整系统进行测试。一些局部反应不出来的问题，在全局上可能会暴露出来。为了发现并解决程序在组装之后可能会发生的问题，需要进行集成测试。集成测试主要验证程序在预期的操作环境中，能否正确且高效的运行，并且测试程序在预期之外的操作环境和操作命令下，能否正确的执行或者给出相应的易于接受的反馈信息。我们从程序的功能性、可靠性、程序性能等方面对“日程管理系统”进行测试。

# 测试方法

## 功能性测试

为了对软件的各项功能正确性和完整性进行验证，发现并解决软件在运行过程中的功能实现问题，我们对程序进行了功能性测试，主要测试方法为黑盒测试：

## 边界测试

根据测试工作经验，大量的错误是发生在输入或输出范围的边界上，而不是发生在输入和输出范围 的内部，因此需要针对各种边界情况设计测试用例。

因此构造多个边界事件输入，其中包括

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
2121/12/31
23:59:00
999999999999
```

对于这组数据，导致在时间上对浮点运算出现精度问题，通过进行边乘除边加减的方式解决。

## 兼容性测试

程序在网页搭载，因此支持多终端访问，进行了兼容性测试。

运行环境： Win10 操作系统， Intel Core i7-8750H @2.20GHZ 内存 8G Chrome， Win10 操作系统， Intel Core i7-8750H @2.20GHZ 内存 8G Firefox， IPadOs Chrome

经测试，兼容性良好，无异常出现。