**Deadline：2017-09-15（周六） 早上8:00**

**0、前言**

提高自己的实践能力  
**Practice makes perfect.** 没有捷径可走，就是**练习、练习再练习！** 那么，就让我们从第一个个人项目开始，修炼我们的程序员之道吧！题目要求如下。

**1、阅读**

查阅一些参考资料，特别是《构建之法》这本书的第一章至第三章的内容，并在下方作业里体现出阅读后的成果。特别是第2章中的个人软件开发流程（PSP）及单元测试。

**2、题目描述**

## 3、要求与说明

* 【编程语言】不限
* 【项目设计】分析并理解题目要求，独立完成整个项目，并将最新项目发布在Github上。
* 【项目测试】使用单元测试对项目进行测试，并使用插件查看测试分支覆盖率等指标。
* 【源代码管理】在项目实践过程中需要使用Github管理源代码，代码有进展即签入Github。签入记录不合理的项目会被助教抽查询问项目细节。
* 【博客发布】按照要求发布博客，利用在构建之法中学习到的相关内容，结合个人项目的实践经历，撰写解决项目的心路历程与收获。博客与Github项目明显不符的作业将取消作业成绩。
* 博客注册，点下面链接进行注册：

http://edu.cnblogs.com/campus/gdgy/informationsecrurity/join?id=CfDJ8J0rgDI0eRtJkfTEZKR\_e82VR7kznd95HB0PhMqfXzfz8yGj7gdIKUWJG51TRUutYPcoN-xyhJyLZJkU76ZiOpIlAs4ICGQCbfZOTnlxXBweT9yPmYHVBZRTgY5rsNSEkHeP7UP80yFdd-YydPf8C1k

## 4、博文规范

1. 在文章开头给出**Github项目地址**。（1分）
2. 在开始实现程序之前，PSP表格(参加附录1)记录下你估计将在程序的各个模块的开发上**耗费的时间**。（1分）
3. **解题思路**描述。即刚开始拿到题目后，如何思考，如何找资料的心路历程。（3分）
4. **设计实现过程**。设计包括代码如何组织，比如会有几个类，几个函数，他们之间关系如何，关键函数是否需要画出流程图？（5分）
5. **代码说明**。展示出项目关键代码，并解释思路与注释说明。（4分）
6. **测试运行**。程序必须是可运行的，展示出程序运行的截图。PS：如果有扩展需求或者更高级的需求，请秀出来，有额外加分。（3分）
7. 在你实现完程序之后，在PSP表格记录下你在程序的各个模块上**实际花费的时间**。（1分）
8. **项目小结**。结合PSP表格总结项目的成败得失，分享经验，总结教训。（2分)

**博客参考模板** <http://www.cnblogs.com/vertextao/p/7469789.html>

## 5、评分规则

本次个人项目分数由两部分组成，分别是

1. 博客 — 20分，分数组成在博文规范中。
2. 程序 — 40分（附加分20分）
   * 10分为源代码管理评分，该评分主要通过源代码管理中的commit注释信息，增量修改的内容，是否有运行说明等给分。
   * 10分为正确性评分，按要求输入参数后，程序能够正确运行。
   * 20分为功能评分，完全满足需求1-4。
   * 附加分20分，实现附加需求1&2各得10分
3. **注意事项：**

* 按时间完成并提交——正常评分
* 晚交一周以内——0分
* 晚交一周以上或不交——倒扣本次作业分数
* **抄袭——倒扣2倍本次作业分数【严禁代码与博客等一切形式的抄袭！博客园支持了对博客的查重功能，我们也有专用的代码查重系统进行代码查重。请各位同学千万不要触碰底线，勿谓言之不预也！】**

## 附录

### 1. PSP2.1表格

| **PSP2.1** | **Personal Software Process Stages** | **预估耗时（分钟）** | **实际耗时（分钟）** |
| --- | --- | --- | --- |
| Planning | 计划 |  |  |
| · Estimate | · 估计这个任务需要多少时间 |  |  |
| Development | 开发 |  |  |
| · Analysis | · 需求分析 (包括学习新技术) |  |  |
| · Design Spec | · 生成设计文档 |  |  |
| · Design Review | · 设计复审 (和同事审核设计文档) |  |  |
| · Coding Standard | · 代码规范 (为目前的开发制定合适的规范) |  |  |
| · Design | · 具体设计 |  |  |
| · Coding | · 具体编码 |  |  |
| · Code Review | · 代码复审 |  |  |
| · Test | · 测试（自我测试，修改代码，提交修改） |  |  |
| Reporting | 报告 |  |  |
| · Test Report | · 测试报告 |  |  |
| · Size Measurement | · 计算工作量 |  |  |
| · Postmortem & Process Improvement Plan | · 事后总结, 并提出过程改进计划 |  |  |
| 合计 |  |  |  |

### 2. 参考资料

* C++：[VS2015安装与C++单元测试](http://www.cnblogs.com/xiaoyongwu/p/5289964.html)
* C#: [基于VS2015的C#的单元测试](http://www.cnblogs.com/libaoquan/p/5296384.html)
* Java: [Intelli IDEA简易教程(含单元测试)](http://www.cnblogs.com/rocedu/p/6371315.html)
* Python: [基于PyCharm的Unit test](https://confluence.jetbrains.com/display/PYH/Creating+and+running+a+Python+unit+test)