样卷 2 参考答案

一、by 吕鑫

1.

(a) 可以参考 ppt-02 第 18 页来写 ,因为不确定答案 ,所以大家自己看就好。

(b) 优点：

1. 使用了多个分支，便于多个版本或不同模块间的共同开发。
2. commit 次数较多，较好的体现了项目的开发进度
3. 使用了 pull request，更符合多人开发大型项目的做法

缺点：

1. commit 注释中英文混叠，没有一个标准
2. 有用的 commit 注释较短，另有 Bug Fix Again!等无用注释
3. 分支相对比较混乱（并不确定）
4. 11 月 5 号-11 月 18 号之间 commit 次数只有 5-7 次，之后 1 天 commit 十几次，不符合开发规范。（不确定是否属于版本管理）

2.

（注，以下仅供参考，而且按照 ppt 上写的比较简略，设计的也比较粗糙，如果时间充足可以按照迭代计划开发表那样规范的来写）

Product Backlog

ID	Backlog Item	Estimate(day)
1	前端设计出管理员输入图书馆信息界面	3
2	后台构建存储图书馆信息的 model（类）	3

3	后台处理前端管理员输入信息的请求	4
4	后台根据存储的图书馆信息去访问服务接口	4
5	前端构建用户访问图书馆服务界面	4
6	前端根据用户的请求，调用后台来访问图书馆开放接口，渲染出页面	3
7	后台用过接口查询合作图书馆的图书信息	4
8	前端在原来查询图书信息的基础上，增加查询合作图书馆的选项，调用后台接口查询图书信息	3
9	前端设计出用户使用申请借阅的界面	5
10	后端建立借阅申请的类，包括申请时间等信息	2
11	前端可以与后端交互得到用户借阅和申请结果	2
12	后端为管理员设计处理借阅申请的接口	5
13	前端设计管理员处理申请的界面	4
14	前端根据申请结果来对用户进行提醒	3

以 10 天一个 Sprint，迭代计划开发表如下

Sprint1

ID	Author	Estimate(day)
1	A, B	3
2	C	3
3	C	4
4	C	4

5	A, B	4

Sprint2

ID	Author	Estimate(day)
6	A, B	3
7	C	4
8	A, B	3
9	A, B	5
10	C	2

Sprint3

ID	Author	Estimate(day)
11	A, B	2
12	C	5
13	A, B	4
14	A, B	3

二、by 翔班

a)、课件 7 第 79、80 页

（完整性：应覆盖所有功能和约束；每个需求都包括了必要的信息）

分析：该用例图的分析完整性不足，至少应增加以下信息：

用户浏览水果商店时，排序所使用的“用户喜好”的来源（是否从数据库中获取信息）。

对于非注册用户和没有购买经历的注册用户，浏览的水果商店时的排序方式。

新用户注册时，对用户数据库的更新。

对用户“挑选水果”这一流程更清晰的表述，如“点击水果”和“加入购物车”应以怎样的顺序进行，分别给用户怎样的反馈等。

……

（一致性：需求不应彼此冲突，即对同一系统功能不应出现不同的描述或相互矛盾的约束）

分析：该用例图的分析一致性不佳，以下部分可能存在矛盾：

非注册用户点击水果时，更新用户数据库时没有可用的用户信息，会出现矛盾。

“点击水果”和“加入购物车”都可能用来描述用户挑选水果这一功能，可能存在矛盾。

……

b)、

序列图没有在我们的讲课中提到，有兴趣的同学可以上网查阅资料。

c)、课件 7 第 35~40 页

该系统需要满足的非功能需求可能有：

①、系统应具备较高的可靠性。软件每运行 1000 次，在运行中崩溃的现象出现的次数不超过 5 次。

②、系统应具备健壮性。若用户购买水果时，软件异常停止，用户的购物车和订单信息可以在重启软件后恢复。

③、App 上切换不同页面所需的响应时间在 0.1s 以下。

④、商店中的水果信息应实时更新，每两次更新间隔的时间不应超过 0.5 小时。

⑤、在一个 30MBbps 网速的网络连接上，系统可支持 100 个并发用户同时购买水果。

⑥、数据中心需支持 700GB 的数据存储，并且数据容量每年按 30%的速度递增。每年需要保留 2 个数据备份。

⑦、系统应具备高可用性。后台系统每年度正常运行的时间应占全年时间的 99%以上，应至少有两台备份服务器。

⑧、系统应具备易用性。95%从来没有用过该 App 的用户应该在不超过 5 分钟的熟悉和适应后，就能正确应用系统购买水果。95%的情况下，使用该 App 购买过水果的用户应该在平均 3 分钟内、最多 5 分钟内，完成一次购买水果的操作。

三、代码质量 by OnionYST

1. 代码注释

这个主要考虑如何写出高质量的代码注释，需要注意的是：代码注释的目的、特征。

> 很明显左边的不如右边的。

- 代码注释的目的

```C++
size_t picNum = _picVec.size();  // no need to explain what picNum is
```

首先要有一个共识：代码注释是辅助好代码的，不是辅助烂代码的。有些事情可以由代码的写法来实现，比如变量含义之类，这些就没必要通过注释来说明。代码注释不要 describe，也不要 replace。

> 变量名字一看就是 username，password，还要注释来再说一遍做什么
>
> 对 param 有什么要求、约束没写，return 什么东西没写，except 要抛什么异常也没写，这注释写了等于没写，指着代码行数算工钱吗

需要说明的是 **【为什么这样写 code】**，code 为什么这样组织，特别的注意事项，边界条件等。

> 相比之下，右边这份解释了函数的使用功能（而非含义），给出了变量 sql 的约束说明，对返回值的含义有说明，解释了 exception 的触发条件。对于一个调用者，这样的接口说明才算注释。（当然，@return 那里能换个行就更好了）

- 代码注释的特征

代码注释要简洁明了，也要注意全面，让读者能很快掌握这部分 code。

> 这个结合着来说就行

2. 单元测试

```python
import unittest
from findobj import find_object


class FindObjectTestCase(unittest.TestCase):

    def test_find_module(self):
        import os
        self.assertEqual(find_object('os'), os)

    def test_find_module_module(self):
        import os.path
        self.assertEqual(find_object('os.path'), os.path)

    def test_find_module_module_object(self):
        import os.path
        self.assertEqual(find_object('os.path.split'), os.path.split)

    def test_find_empty_name(self):
        self.assertRaises(ImportError, find_object, '')


if __name__ == '__main__':
    unittest.main()
```

（这道单元测试是去年原封不动的作业）

我也不知道怎么说，但上面这个代码 work，就这样吧……

样卷2

1. **四、by 则腿**

    1. 1.
        - B的设计让程序的逻辑关系更清晰
        - B的设计满足了开闭原则.增加新的类型只需要增加新的代码,而不需要修改已有的实现.

    2. B使用组合代替继承,让逻辑关系更简单.A需要$m \cdot n$个类,B只需要$m + n$个类,实现更简单,避免了类爆炸,提高了代码的可复用性.

    2. 1.

```Java
setChanged();
notifyObserver();
```

1.

```Java
this.observable = observable;
this.observable.addObserver(this);
```

1.

```Java
WeatherData weatherData = (WeatherData)obs;
this.temperature = weatherData.getTemperature();
this.humidity = weatherData.getHumidity();
this.pressure = weatherData.getPressure();
```

五、软件测试

黑盒测试是最基本的测试之一，这个 SATM 去年是至少当了两年作业然后放出来当考试。第一问要求设计 5 个测试用例，我写完了然后发现后面 2-4 问竟然全是对 1 的补充说明，所以希望各位好好审题……至少先全看一眼……

2 要求设计规范，有数据操作输出，走的是基本路径和扩展路径的路子。

3 要求覆盖正常输入和异常输入，4 要求分区和边界值，是一个意思，黑盒测例要强要全面。

这种实际场景（非数学函数）的测试设计是很综合的题目，特点是要写很多很多很多字，所以我最后不到半个小时基本在一路狂飙。而且这种东西没有标准答案，在此只能提供一些思路（脑洞）。如果遇到这样的题，祝各位好（xie）运（wan）……

- 登录功能：

我插了银行卡然后输入密码

我插了银行卡然后输入错误的密码

我插了银行卡然后拔了卡

我插了银行卡然后输了一半密码然后拔了卡

我插了半张银行卡然后使劲疯狂按密码

我插了一张饭卡

我插了一块酒精棉

- 存款功能

我在存款信封口放了一张红票子

我在存款信封口放了一张 21474837 张红票子

我在存款信封口放了半张红票子

我在存款信封口放了一张红票子然后拔了出来

我在存款信封口放了一张红票子然后把银行卡拔了出来

我在存款信封口什么都没放

- 取款功能

我输入取钱 100 块

我输入取钱 0 块

我输入取钱 B1 B2 B3 块

我输入取钱 100 块，然后拔了银行卡

我把现金交付口用海绵堵了起来，然后输入取钱 100 块

我输入取钱 100 块（老子卡里就五毛）