

主管
领导
审核
签字

哈尔滨工业大学 2016-2017 学年秋季学期

软件工程

试 题

题号	一	二	三	四	总分
得分					
阅卷人					

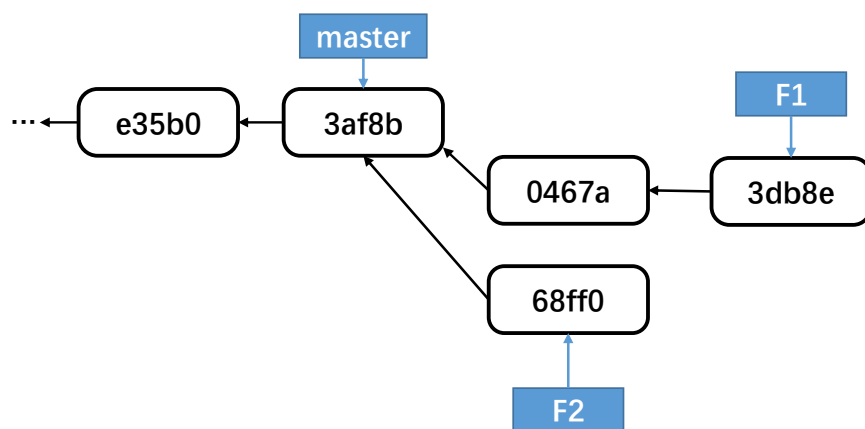
片纸鉴心 诚信不败

本试卷满分 100 分，按 30%计入总成绩。

一 团队协作开发（25 分）

某开发团队有四个成员 A、B、C、D，协作开发项目 P。A 为主程序员，负责项目主体代码的搭建和主分支（master）的维护，B、C 分别负责特性 F1 和 F2 的开发与维护，D 负责测试。项目使用 Git 进行版本管理，团队有一个私有 Git 服务器 S（项目地址为 <https://xxx/P.git>），同时还使用 GitHub 作为另一个 Git 服务器（项目地址为 <https://github/T/P.git>）。每个项目组成员在本地使用 Git 进行代码管理，通过服务器 S 进行协作，而 GitHub 上的仓库用于对外提供开源。

项目进展到 12 月 1 日，服务器 S 上的版本历史如下所示，其中白色圆角矩形表示 commit，深色矩形表示分支。各成员本地的 Git 仓库与此保持一致且均已添加了远程仓库 S（命名为 origin），但没有配置其他远程仓库。此时，GitHub 上的项目仓库里只包含 master 分支。



以下场景均针对 12 月 1 日开始的团队协作：

- (1) D 开始了对 F1 最新版本的测试，发现了两个 bug，通过 Email 报告给了 B；
- (2) B 启动了对 F1 的修改，修复了这些 bug 并在本地做了提交，推送至 S；
- (3) C 将 B 的最新修改合并至自己所负责的分支 F2 并推送至 S；
- (4) D 启动了对 F2 的测试，通过 Email 向 C 反馈了发现的 bug；
- (5) C 接到 D 的反馈，在自己的分支上修复了 bug，提交并推送至 S；
- (6) D 重新对 B 和 C 的分支进行测试，在 B 的分支上未发现 bug，在 C 的分支上发现了几

一个小 bug。他在 C 的分支上开辟了一个新分支 F2_fix，对代码做了修改来消除这几个小 bug，进而推送至服务器 S；

(7) 之后，C 自己也发现了这几个小 bug。他在本地对代码做了相应的修改和提交。在推送至服务器 S 之前，他先是获得 S 上的最新更新，发现了 D 的修改工作和新分支。他试图将 D 的分支合并到自己的分支 F2，但提示有冲突，后通过手工方式消除了该冲突，成功合并，并将所有本地更新推送至 S；

(8) 此后，D 通过 Email 向 A、B、C 分别报告了测试结果；

(9) A 将 B 的分支 F1 合并至主分支并推送至 S 以及 GitHub；

(10) A 将 C 的分支 F2 合并至主分支。由于 F2 特性目前尚不完善，不能对外开源，故仅推送至 S。

题目：

(1) (12 分) 在上图基础上，补充出上述所有步骤结束之后服务器 S 上的完整版本变迁历史。请自行行为各 commit 的 SHA 指定六位长度的十六进制编号，并确保其唯一性。图中应标明所有分支的位置。

二 软件测试 (25 分)

某函数的代码如下所示：

```
/**
 * 功能描述：          找出指定字符串在目标字符串中的位置
 * @param source      目标字符串
 * @param pattern      指定字符串
 * @return             指定字符串在目标字符串中的起始位置，找不到则输出-1
 */
public static int match (String source, String pattern) {
1   int index = -1;
2   boolean match = true;
3   for (int i = 0, len = source.length() - pattern.length(); i < len; i++) {
4       match = true;
5       for (int j = 0; j < pattern.length(); j++) {
6           if (source.charAt(i + j) != pattern.charAt(j)) {
7               match = false;
8               break;
9           }
10          if (match) {
11              index = i;
12              break;
13          }
14      }
15      return index;
16  }
```

已有的测试用例如下：

source	pattern	期望的输出
abc	d	-1

问题：

(1) (5 分) 以上测试用例对上述代码可否满足判定覆盖？如果不能，请补充最少数量的测试用例，使之可

满足判定覆盖的要求，并给出它们可满足判定覆盖的依据。

(2) (15 分) 使用基本路径法，设计一组新的测试用例，对以上表格里的测试用例加以补充，使之可覆盖全部基本路径。增加的测试用例填入右侧表格。另，在完成本题时请使用上页代码中给出的行号，且无需考虑题(1)中增加的测试用例。

绘制控制流图（图中节点编号请使用代码行号）：

计算圈复杂度：

列出基本路径：

编号	包含的代码行的序列（如 1-2-3-4-5）

请将拟补充的测试用例填入下表第 2-5 列。请根据实际情况填写，未必非要填写满所有的行。如果行数不够，请在下方自行增加新行。

(1) 测试用例编号	线 (2) source	(3) pattern	(4) 期望的输出	(5) 覆盖的基本路径编号	(6) 实际输出结果	(7) 是否一致
1						
2						
3						
4						
5						
6						

(3) (5 分) 针对目前的代码（见第 4 页），请模拟执行题(2)补充后的全部测试用例（上表），给出各测试用例的实际输出结果（填入上表第 6 列），并判断期望输出与实际输出是否一致（填入上表第 7 列，使用 Yes/No）。

若存在不一致，意味着代码中有错误。但即使都一致，也不能意味着代码 100%正确。请尝试着找出代码中存在的错误并指明如何修改。如果你认为代码中没有错误，在下方空白处简单回答“代码无错误”即可。

三 需求获取与需求分析 (25 分)

用户需要一个“儿童成长照片日历”的软件，其业务需求表述如下：宝宝刚刚出生，父母希望有一个软件，每天给孩子拍一张照片并留下文字，从而完整的纪录他/她的成长历程并以照片日历的形式展示，待孩子长大之后给她惊喜。用户介绍了以下的功能需求：

- [1] 设定宝宝名字、出生日期等
- [2] 每天拍 1 张照片加入日历，并可辅以简要的文字说明或语音说明
- [3] 也可从手机中导入已有图片，将其加入到日历中的某个日期
- [4] 每天一页展示，也可按日历形式展示，日历中每个格子的背景是当天的照片
- [5] 可修改原有的图片
- [6] 在上传照片前，可对照片做出裁剪等操作
- [7] 上传照片时，检查照片的拍摄日期，若与当天日期不合，则不允许上传
- [8] 若当天未传照片，对用户发出提醒
- [9] 可自动计算各关键日期并提前发出提醒，对这些特殊日期，允许传更多的照片
- [10] 可按关键字、日期、日期段检索照片
- [11] 可打印选定日期的单张照片、日期段内的多张照片

问题：

- (1) (10 分) 以上哪些需求是“不好”的需求（意即存在各位表述问题而导致开发者无法清晰理解用户意图，从而难以支持直接开发）？从你认为“不好”的需求中任选 2 项，分别解释其为什么“不好”。

(2) (15 分): 针对问题(1)中你所选定的 2 项“不好的”需求, 如果用户目前正在与你面对面沟通, 你作为开发者该向用户提出什么问题, 才能得到确切的需求细化/修改方案? 假设用户针对你的问题可以做出非常明确的回答:(请自行假设用户的回答是什么样子, 且用户的回答应该是理性的、符合实际的), 基于用户的回答对这两项需求进行修改, 使之变为“好”的需求。

问题: 如何展示用户的照片? 如果当天上传了很多的照片应该如何展示?

问题: 关键日期是机器自动计算得到还是由用户自主设定的? 是否允许用户向这些关键日期上传多张照片?

问题: 用户在上传照片时, 是按照照片拍摄的时间自动归到日历的某一天, 还是将照片归到用户指定的某天?

密
封
线

四 OO 分析与设计 (25 分)

你要开发一个日历应用, 帮助用户管理自己的日程安排。这是一个 Web 应用, 支持多个用户使用。

每个用户可以建立 1 或多个日历, 每个日历可用于管理该用户不同类型的日程安排。例如, 张三希望自己的第一个日历记录自己的学习相关活动、第二个日历记录自己的生活相关活动。

用户向自己拥有的某个日历中添加活动, 每个活动具有特定的名称(例如“软件工程期末考

试”)、简介 (0-200 字)、起始日期和起始时间 (例如 2016-12-24 13:00)、结束日期和结束时间 (例如 2016-12-24 15:00)、地点 (例如致知楼 33)。活动也可以不指定具体的起始/结束时间而只有起始/结束日期 (例如从 2016-12-24 到 2016-12-26)。

除了添加活动, 用户还可以删除已有活动, 修改已有活动、平移已有活动 (例如将某活动延迟 3 天进行、将某活动提前 1 小时进行)。

用户希望能在特定日历上执行查询操作、查重操作。“查询”是指: 给定某关键词, 查到该日历中哪些活动包含该关键词并列出这些活动; “查重”操作是指: 给定某日期, 看哪些活动之间存在重叠 (例如: 活动 1 的时间范围是 2016-12-24 13:00-15:00, 活动 2 的时间范围是 2016-12-24 14:30-18:30, 那么它们就有重叠, 重叠范围是当日 14:30-15:00), 该操作的结果是“活动对”及重叠日期/时间范围 (例如: <活动 1, 活动 2, 2016-12-24, 14:30, 15:00>)。

更复杂的操作是跨日历的查询和查重操作, 意即: 在用户拥有的多个日历上进行全局查询 (例如, 满足条件的某些活动来自于日历 1, 某些活动来自于日历 2) 和全局查重 (例如, 日历 1 中的某个活动和日历 2 中的某个活动存在重叠)。

下一个操作是日历的合并, 即: 给定同一用户拥有的两个日历 A 和 B, 将 B 中包含的全部活动合并进入 A, 然后删除 B。

最后一个操作是计算用户每天的“忙碌度”。给定特定的日期- (例如 2016-12-24), 在用户的所有日历上计算该日期的忙碌度。“忙碌度”定义为: 该天的全部活动的累积小时数 \div 24。可以看出, 如果某天内的多个活动之间存在重叠, 忙碌度可能会大于 1。

问题:

(1) (11 分) 基于以上需求陈述, 采用 OO 思想, 识别实体类, 设计每个实体类的属性、方法, 以 UML 领域类图的形式展现分析结果。类图中应给出类的名字、各属性的名字和数据类型、各方法的名字、返回值类型、参数名, 以及各实体类之间的关系 (关联、继承、组合、聚合、依赖)。为了便于理解, 图中所有内容都使用中文命名。

(2) (8 分) 有四个操作: 特定日历上的查重、跨日历的查重、日历的合并、计算用户某天的忙碌度。从中任选 1 个, 给出其伪代码设计。伪代码要符合题目(1)中的 OO 设计思路。

(3) (6 分) 在该日历应用的第二个迭代开发周期内, 用户提出“不允许有重叠的活动”这一新需求, 意即: 当用户向某一日历添加活动时, 首先检测新活动是否与该用户的所有日历中的所有已有活动有重叠, 若有, 则不允许添加。请基于你在(1)中的 OO 设计方案, 如何对其进行最小程度的修改, 使其支持这个新需求。列出所有修改, 阐述每一条修改分别影响哪个 (些) 类中的哪个 (些) 组成部分 (如属性、操作等)。