高级软件工程复习题

2021版

罗铁坚

2021/12/10

目录

[一、 概念解释 4](#_Toc90715005)

[**⯀** 三类问题 4](#_Toc90715006)

[**⯀** 建模和模型 4](#_Toc90715007)

[**⯀** 设计与重构 4](#_Toc90715008)

[**⯀** 代码和界面 5](#_Toc90715009)

[**⯀** 测试 5](#_Toc90715010)

[**⯀** 两种基本途径 5](#_Toc90715011)

[二、 选择题 6](#_Toc90715012)

[**⯀** 软件需求和应用场景 6](#_Toc90715013)

[⯀ 业务建模和数据模型 15](#_Toc90715014)

[**⯀** 软件架构和应用框架 19](#_Toc90715015)

[⯀ 设计模式和代码重构 20](#_Toc90715016)

[**⯀** 接口设计和前端编程 48](#_Toc90715017)

[**⯀** 软件验证与自动测试 50](#_Toc90715018)

[⯀ 性能评价和设计优化 56](#_Toc90715019)

[⯀ 集成部署和系统运营 56](#_Toc90715020)

[⯀ 三大问题 59](#_Toc90715021)

[三、 判断题 61](#_Toc90715022)

[**⯀** 三类问题 61](#_Toc90715023)

[**⯀** 需求 62](#_Toc90715024)

[**⯀** 建模和模型 63](#_Toc90715025)

[**⯀** 设计与重构 63](#_Toc90715026)

[**⯀** 代码和界面 64](#_Toc90715027)

[**⯀** 测试 65](#_Toc90715028)

[**⯀** 两种基本途径 67](#_Toc90715029)

[四、 设计题 67](#_Toc90715030)

[**⯀** 需求分析 67](#_Toc90715031)

[**⯀** 设计模式 68](#_Toc90715032)

[**⯀** 程序与界面设计 71](#_Toc90715033)

[**⯀** 测试 79](#_Toc90715034)

[五、 简答题 80](#_Toc90715035)

[**⯀** 软件理论 80](#_Toc90715036)

[**⯀** 需求 80](#_Toc90715037)

[**⯀** 建模与模型 80](#_Toc90715038)

[**⯀** 设计与重构 82](#_Toc90715039)

[**⯀** 代码和界面 82](#_Toc90715040)

[**⯀** 框架与开发 89](#_Toc90715041)

[**⯀** 测试 93](#_Toc90715042)

[**⯀** 两种途径 94](#_Toc90715043)

# 概念解释

## ⯀ 三类问题

**1.** 什么是软件？软件工程的使命和任务是什么？

2.简答一个好的理论要具备的三个要素**。**

3. 软件三大定律是什么？

4.请简述RUP与UML的含义，并指明两个的不同点。

5. 请简要描述TDD与BDD的开发理念，并说明二者的区别。

6、软件生命周期划分为哪几个阶段?

7.软件的基础设施包括哪些？

8.如何理解软件设计原则中的低内聚高耦合原则？

9. 简单概括，软件工程的的三大根本问题、九项重要任务和两种研发方式。

10. SMART 用户故事

## ⯀ 建模和模型

11.如何建立业务建模的用例视图？

12、简述UML实际建模过程

***13. 敏捷开发方法 与Scrum方法***

14. 基于计划-文档开发方法（Plan-and-Document based Development）

15. 持续集成及开发

16. MVC（软件作为服务的开发框架）

17.什么是螺旋软件开发模型？它有什么特点？

## ⯀ 设计与重构

18、什么是结构分析方法?该方法使用什么描述工具?

19、结构化程序设计的主要内容是什么？它有什么优缺点？

20. 请列出五种设计模式。

21. TDD and 红-绿-重构

***22. DRY (Don’t Repeat Yourself) 无重复代码***

23. 软件开发中的重构是什么？为什么需要重构？有哪些基本原则？

24. 代码味道及类内方法 SOFA原则

（说明S、O、F、A分别代表什么？违法该原则的代码的不好特征、重构和修复的方法）

25. ***类间关系的SOLID原则（说明每个原则的意义、违法该原则的代码的不好特征、重构和修复的方法）***

（1）单一责任原则

（2）开闭原则

（3）里氏替换原则

（4）依赖注入原则

（5）迪米特法则

## ⯀ 代码和界面

26. 文档对象模型（DOM）和jQuery

27. JavaScript函数特点

28. 简述HTML作用与特点

29. Ruby语言的面向对象特点是什么？

30. ruby访问控制public、private和protected

31.Ruby属于什么类型的语言？它有怎样的特性？有哪些优点？

32. CSS 规则构成

33. ruby中的数组的特点，给出如何使用它的方法的例子。

34. Ruby 哈希（Hash）特点，给出使用它的方法例子。

35. Ruby 迭代器：each 和 collect，给出使用迭代器解决问题的例子（语句）

36．解释迭代器背后的的通用机制 yield，给出例子。

## ⯀ 测试

***37.请说出至少五种你知道的测试方法***

38. A/B 测试是什么？（4 分）测试的目的是什么？（3 分）测试步骤是什么？

39. TDD是什么?

***40. FIRST 测试原则***

41. Git 工作流程

## ⯀ 两种基本途径

42.简述软件工程的两种研发方式

43.人类发明软件的动机是什么？软件主要解决什么类型问题？它的优点和局限性是什么？

44. 用软件解决业务问题的基本条件和途径是什么？

45. 为什么说人工智能或机器人是软件的一种能力。

46. 你认为设计和实现软件必须解决什么关键问题？

47. 软件的需求分析、设计实现和运营服务等方面的问题都必须回答下面的四类问题是什么？

1. 一个具体领域或某项业务的问题是什么？ 有没有已经解决这个业务问题的软件系统?
2. 开发解决业务问题的软件的必要性和可行性？回答软件交付和运营后是否产生价值？
3. 寻找业务问题与软件需求之间的关系。潜在的用户在什么应用场景中使用，并获得什么好处，效率或效果体现在哪？，有没有具体的量化指标评价这些效果或效率？
4. 采用什么“语言”表达需求，使得领域专家和软件设计人员都能相互理解和流畅沟通。
5. 什么是一个软件架构？什么是合理的软件架构？如何为特定的应用设计可演化高质量的架构。
6. 对比分析Python Django 与 Ruby on Rails 的优点和缺点

50. 应用间调用什么接口方式实现调用功能或交换数据？，应用接口（如REST等）设计的优缺点？

51. 人在参与解决业务问题中需要什么关键信息？在什么设备或终点上呈现这些人需要的决策信息？采用什么表现形式和用什么语言来实现这些信息导航或推送？

52. 机器与机器的接口与人机接口的区别？

53. 信息产生价值的必要条件和充分条件是什么？

54. 在信息表达中，什么是有效信号和干扰噪音？

55. 前端编程的模型和语言有哪些？它们的特点是什么？

56. 什么是一个软件成功的标准？

57. 如何进行软件的验收和功能测试？

58. 如何设计测试代码，举例说明你熟悉的自动测试代码工具使用方法、优点和不足。

59. 什么安全？计算机中的软件安全是什么？

60. 应用安全的主要威胁和风险，如何进行应用安全设计？

61. 软件性能的主要指标是什么？

62. 否如何测试软件性能是否满足设计要求？

63. 如何进行应用的性能优化？

# 选择题

## ⯀ 软件需求和应用场景

1. 下面哪些用户故事描述是满足的 SMART 原则要求的?

A. 用户能用影片名查找电影信息。   
B. 作为一个用户，我想看前 10 个按价格从低到高排序的影片列表，以便我买

到最便宜的电影票。

C. Rotten Potatoes APP 应该有好的用户界面。

D. Rotten Potatoes APP 应该有好的用户界面和快的响应时间。

***2.下面哪个是对，就用户故事而言?***

(i) 它们应该描述该应软件是如何使用的。

(ii) 它们应该有业务价值。

(iii) 它们不必要是可以测试的

(iv) 它们应该在敏捷软件开发生命周期的各阶段都能实施或实现。

1. i only
2. i and ii
3. i and iv
4. i, iii, and iv

3. 相对强调过程的场景描述而言，声明性场景描述特点是:

A. 有更复杂语句和步骤。   
B. 不是 DRY。   
C. 更段、易理解和维护。

D. 关注在低层步骤上，这些步骤需要初始化设置和执行测试。

4. 对隐性和显性需求而言，下面哪些说法是对的？

A. 无论隐性和显性需求，你都不能写用户故事。   
B. 隐性需求更简洁，而显性需求更啰嗦。

C. 隐性需求是显性需求的逻辑结果，通常对应于综合性测试。   
D. 显性需求通常可以用场景化过程性语言描述，而隐性需求通常用声明性语言描述。

5. 需求分析最终结果是产生

A. 项目开发计划

B. 可行性分析报告

C. 需求规格说明书

D. 设计说明书

6. 需求分析中，开发人员要从用户那里解决的最重要的问题是

A. 让软件做什么

B. 要给软件供哪些信息

C. 需求软件工作效率怎样

D. 让软件具有何种结构

7. 需求规格说明书的内容不应包括对（）的描述。

A. 主要功能

B. 算法的详細过程

C. 用户界面的运行环境

D. 软件性能

8. 软件需求分析阶段的工作，可以分成 4 个方面：需求获取，需求分析，编写需求规格

说明书以及（）

A. 用户

B. 需求评审

C. 总结

D. 都不正确

9. 在原型法中，开发人员根据（）的需求不断修改原型，直到满足客户要求为止

A. 用户

B. 开发人员

C. 系统分析员

D. 程序员

10. 需求验证应该从下述几个方面进行验证：

A. 可靠性、可用性、易用性、重用性

B. 可维护性、可移植性、可重用性、可丬则试性

C. 一致性、现实性、完整性、有效性

D. 功能性、非功能性

11. 以下哪一项不是restful的特点。（ ）

A. 通常使用json或xml进行数据交换

B. 是无状态的，不用考虑上下文

C. 将网络上的所有事务抽象成资源

D. 不具有自解释性

12. 下面哪项不属于软件开发的测试方法。（ ）

A. A/B测试

B. 单元测试

C. 集成测试

D. 代码测试

13. 使用ruby进行web开发时，下面哪项不是可选框架。（ ）

A. Ruby on Rails

B. Sinatra

C. Camping

D.Springboot

14. 若一个web应用采用MVC架构进行设计，则对于不同http的url进行区分和处理的模块是（ ）。

A. M

B. V

C. C

D. MVC合作完成

|  |  |
| --- | --- |
| 15. 敏捷开发者崇尚的价值点是 | |
|  | A. 过程和工具胜于个人与团队交流 |
|  | B. 可工作的软件系统胜于全面的文档 |
|  | C. 合同谈判胜于与客户协作 |
|  | D. 遵循计划胜于响应需求变化 |
|  |  |
| 16. 瀑布软件开模型与螺旋软件开发模型的区别？ | |
|  | A. 瀑布模型包括了大量的计划文档和时间跨度大的可工作软件版本，而螺旋模 |

型小量的计划文档和时间跨度小的可工作软件版本。

|  |  |
| --- | --- |
|  | B. 瀑布模型从开始就写全所有的需求文档，而螺旋模型按迭代阶段写需求文 |

档。

|  |  |
| --- | --- |
|  | C. 瀑布模型有教长的迭代周期，而螺旋模型有较短和快的迭代周期。  D. 瀑布模型的在每个阶段完成后进行保障测试，最后验证阶段包括了验收测 |

试；而螺旋模型的保障测试在每 2 个月内进行。

17. 敏捷开发者崇尚的价值点是

A. 过程和工具胜于个人与团队交流

B. 可工作的软件系统胜于全面的文档

C . 合同谈判胜于与客户协作

1. 遵循计划胜于响应需求变化

18. 瀑布软件开模型与螺旋软件开发模型的区别？

A. 瀑布模型包括了大量的计划文档和时间跨度大的可工作软件版本，而螺旋模型小量的计划文档和时间跨度小的可工作软件版本。

B. 瀑布模型从开始就写全所有的需求文档，而螺旋模型按迭代阶段写需求文档。

C. 瀑布模型有教长的迭代周期，而螺旋模型有较短和快的迭代周期。

D. 瀑布模型的在每个阶段完成后进行保障测试，最后验证阶段包括了验收测

19. 需求分析最终结果是产生（）

A、项目开发计划

B、可行性分析报告

C、需求规格说明书

D、设计说明书

20.在软件工程中高质量的文档标准是，完整性，一致性和（）

A． 统一性

B． 安全性

C． 无二义性

D． 组合性

21.面向对象中，对象之间的协作是通过()来实现的。

A.依赖

B.泛化

C.消息

D.关联

22. 软件产品的质量是由()阶段的工作决定的。()

A.需求分析

B.软件设计

C.软件构建

D.软件测试

|  |  |  |  |
| --- | --- | --- | --- |
| 23. 软件需求分析阶段的工作，可以分成 4 个方面：需求获取，需求分析，编写需求规格说明书以及\_\_\_\_\_\_\_\_\_\_。 | | | |
| A. 用户 | B. 需求评审 | C. 总结 | D. 都不正确 |

24. 需求分析的方法包括

A.结构化分析技术、面向对象的分析技术、原型开发技术

B.结构化分析技术、面向对象的分析技术

C.面向对象的分析技术、原型开发技术

D.原型开发技术、结构化分析技术

25. 下列哪个选项不是软件工程面临的根本问题？   
A． 构思有价值的软件系统或产品，寻找检验其质量的测试用例和量化方法。   
B． 建立一套应对领域问题及其需求变化的软件架构和代码模块的理论方法。   
C． 设计和验证可持续发展的软件生产和部署运营的效益模型。   
D． 快速完成代码书写，快速适应市场而不注重代码的架构和后续修改问题。

26.单元测试是在软件开发过程中的哪个阶段完成的？（）

A . 需求分析

B . 概要设计

C . 实现/详细设计

D . 使用和维护

27. 在E-R模型中，包含以下基本成分:

A. 数据、对象、实体

B. 控制、联系、对象

C. 实体、联系、属性

D. 实体、属性、操作

28.下列哪项不是软件需求阶段所要完成的任务？

A. 需求获取   
B. 需求完善

C. 需求定义   
D. 需求规约

29.下述情况分别最适合采取哪种需求发现的方式分别是什么？   
① 为解决生活中遇到的麻烦事而开发的软件   
② 有较多繁琐环节的社区医保系统的开发   
③ 某小型团体组织开发其内部人员管理系统   
④ 某大型连锁集团开发集团人员管理系统   
⑤ 某专业化软件外包公司接手烂尾的软件开发项目   
A. ①-自悟；②-观察；③-交流；④-小组会；⑤-提炼   
B. ①-观察；②-自悟；③-小组会；④-交流；⑤-提炼   
C. ①-自悟；②-交流；③-观察；④-提炼；⑤-小组会   
D. ①-提炼；②-自悟；③-交流；④-观察；⑤-小组会

30. 确定测试计划是在（A）阶段制定的。

A. 总体设计

B. 详细设计

C. 编码

D. 测试

31. 需求验证应该从下述几个方面进行验证：   
A. 可靠性、可用性、易用性、重用性   
B. 可维护性、可移植性、可重用性、可丬则试性   
C. 一致性、现实性、完整性、有效性   
D. 功能性、非功能性

32. 软件生命周期中花费最多的阶段是   
A、软件编码。   
B、详细设计。   
C、需求分析。   
D、软件测试和维护。

33. （ ）活动数量较大，占整个维护活动的 50%。   
A、完善性维护。   
B、预防性维护。   
C、校正性维护。   
D、适应性维护。

34. 下面哪个测试类型不属于性能测试？（ ）

A. 负载测试

B. 压力测试

C. 稳定性测试

D. 兼容性测试

35. 为了提高软件的可维护性，在编码阶段应注意（ ）   
A.保存测试用例和数据 B.提高模块的独立性   
C.文档的副作用 D.养成好的程序设计风格

36. 从结构化的瀑布模型看，在它的生命周期中的八个阶段中，下面的几个选项中哪个环节出错，对软件的影响最大( )   
A、详细设计阶段 B、概要设计阶段   
C、 需求分析阶段 D、 测试和运行阶段

37. 需求分析最终结果是产生( )。

A. 项目开发计划

B. 需求规格说明书

C. 设计说明书

D. 可行性分析报告

38.需求验证应该从下述几个方面进行验证：

A. 可靠性、可用性、易用性、重用性

B. 可维护性、可移植性、可重用性、可丬则试性

C. 一致性、现实性、完整性、有效性

D. 功能性、非功能性

39. 在面向对象方法中，信息隐蔽是通过对象的( )来实现的。

A. 分类性

B. 继承性

C. 封装性

D. 共享性

40. 以下哪项工作不适宜在需求分析阶段完成？（ ）

A. 挖掘真实需求

B. 提出解决方案

C. 筛选、验证方案

D. 完成程序建模

41.软件需求分析阶段的工作，可以分成 4 个方面：需求获取，需求分析，编写需求规说

明书以及（）

A 用户 B 需求评审 C 总结 D 都不正确

42. 关于 Ruby语言下列说法错误的是（ ）。

A. 在 Ruby中 Everything is object, 例如 a=1, 整型变量 a也是对象

B. 内置了正则表达式引擎，可以方便地处理字符串

C. 函数返回参数必须要 return

D. Ruby是解释型语言，其性能比 Java差

43. 软件的复杂性是（ A ），它引起人员通信困难、开发费用超支、开发时间超时等问题。

A. 固有的

B. 人为的

C. 可消除的

D. 不可降低的

44. 在结构化分析方法中，（ C ）表达系统内部数据运动的图形化技术。

A. 数据字典

B. 实体关系图

C. 数据流图

D. 状态转换图

45. Linux系统下，使用（B）命令修改文件或目录的权限。

A. chown

B. chmod

C. mv

D. ln

46．git fetch + git merge的命令组合可与（D）命令实现相同的效果。

A. git push

B. git checkout

C. git commit

D. git pull

47. Ruby语言中，a='ABC'，b='DE'，下面操作（D）不能得到'ABCDE'。

A. a.concat(b)

B. a + b

C. a << b

D. "#{a} #{b}"

48. Ruby on Rails中，使用（A）命令来初始化一个需要重新部署的数据库。

A. rake db:schema:load

B. rails db:migrate

C. rails server

D. rails db:setup

49.需求分析是（）

A. 软件开发工作的基础

B. 软件生存周期的开始

C. 由系统分析员单独完成的

D. 由用户自己独立完成的

50. 甲方要求乙方完成一个疫情期间能用于线上开会的在线视频软件，而且必须要有一定

的安全性，即公司局域网外无法连入该软件，因为甲方可能用于商谈公司高层机密。乙交

付后甲方发现在家也能用这个软件。请问乙方在哪个环节出了问题（）

|  |  |  |
| --- | --- | --- |
| A.软件架构设计 B.安全问题与防御设计 |  | C.用户体验接口设计 D.需求分析 |
|  |  |  |

51. 需求分析是软件生存周期中的一个重要阶段，应该在（）进行。

A.维护阶段

B.软件开发全过程

C.软件定义阶段

D.软件运行阶段

52. 软件工程中三大类问题不包括（ ）。

A. 发明有价值的软件系统并找到检验其质量的理论方法

B. 探究对应领域问题软件需求变化的可动态升级模块和架构的理论方法

C. 测算软件设计实现和运营的效益模型的理论方法

D. 研究发明 可 自动化测试软件系统的理论方法

53.软件生产过程中，需求信息由（ ）给出。

A、程序员 B、项目管理者 C、软件分析设计人员 D、软件用户

54. 软件工程的形成不包含对以下哪一项知识的系统化归纳提炼（ ）。

A.软件的创造构思

B.软件的设计代码

C.软件的市场营销

D.软件的组织运营

55. 软件需求分析的任务不包括（ ）。

A、问题分析

B、信息域分析

C、结构化程序设计

D、确定逻辑模型

56. Ruby中alias关键字的作用是什么。（ ）

A．定义别名

B.定义函数

C．定义模块

D.不是ruby的关键字

## ⯀ 业务建模和数据模型

|  |  |  |
| --- | --- | --- |
| 57. 下面的例子中哪个是属于表示层? | | |
|  | A. Apache HTTP 服务器, Ruby and Rails, MySQL 数据库 | |
|  | B. Ruby and Rails, Django, Symfony, Catalyst | |
|  | C. MySQL、PostgreSQL 、Oracle 数据库 | |
|  | D. Microsoft IIS, Apache HTTP 服务器, lighttpd 服务器 | |
| 58. 一个 HTTP 请求必须都包含哪两项 | |
|  | A. CRUD 操作, 数据库 |
|  | B. header, cookie |
|  | C. URL, wildcard (e.g. :id) |

59. 下面的哪个场景不应该用 HTTP GET 来现。

A. 在一个相片分享的网站上，用户点击一个指定的相册。   
B. 用户点击提交按钮发送一封邮件。   
C. 用户在下拉框中选择搜索查询条件。   
D. 用户在影评网 RottenPotatoes 上查看修改个人信息。

60. 某开发团队在合作开发某一项目时，成员 A 认为某一功能由成员 B 负责的模块实现比在自己的模块中实现简单，而成员 B 则认为该功能应该由成员 A 的模块进行实现，针对上述情况，你认为他们在软件工程九个重要任务中哪个任务没有合理完成？

A、软件需求和应用场景   
B、集成部署和系统运营   
C、软件架构和应用框架   
D、业务建模和数据模型

61. 下列哪个图示强调的是系统式的建模？

A、活动图   
B、用例图   
C、类图   
D、时序图

62. 下面哪一种不是属于面向对象设计原则。（ ）。

A．开闭原则

B．里氏替换原则

C．以来倒置原则

D．元编程设计原则

63. 下列关于观察者模式（Observer Pattern）的叙述，错误的为：   
A. 定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的

对象都得到通知并被自动更新。

B. 一个对象状态改变给其他对象通知的问题，而且要考虑到易用和低耦合，保证高度的协

作。

C. 一个对象（目标对象）的状态发生改变，所有的依赖对象（观察者对象）都将得到通知，

进行广播通知。

D. 当一个对象被修改时，不会自动通知依赖它的对象，需要自行进行调用。

***64. 下⾯哪些任务不适⽤快速迭代开发的软件⽅式？ (i) 卫星控制系统 (ii) ⼿术机器 ⼈ (iii) 模拟⽕箭控制的⽹络游戏 (iv) 电脑的操作系统***

A: i 和 ii   
B: i, ii 和 iii   
C: i, ii 和 iv   
D: 全部

65.数据耦合、公共耦合、标记耦合、控制耦合的耦合性从低到高的顺序是（ ）

A.数据、公共、标记、控制 B.数据、标记、控制、公共

C.控制、数据、标记、公共 D.控制、数据、公共、标记

66. 具有风险分析的软件生命周期模型是（  C   ）。

A.瀑布模型

B.喷泉模型

C.螺旋模型

D.增量模型

67. 软件工程的基本要素包括方法、工具和（ A ）。

A. 过程

B. 软件系统

C. 硬件环境

D. 人员

68.在 SD 方法中全面指导模块划分的最重要的原则是( )

A.程序模块化 B.模块高内聚

C.模块低耦合 D.模块独立性

69. 发现软件系统中的Bug后，在报告和跟踪Bug时，以下哪种方式最不合适？（ ）

A. 在代码托管平台（如GitHub）上提交issue，带Bug修复后由开发者关闭issue。

B. 使用Bug跟踪工具（如Jira）管理Bug，并及时在工具中更新Bug修复状态。

C. 使用邮件通知开发者，并讨论解决方式。

D. 使用自动化测试工具，自动地将Bug信息分配给对应开发者。

70.软件设计中划分模块的一个准则是（ ）

A、低内聚低耦合

B、低内聚高耦合

C、高内聚低耦合

D、高内聚高耦合

71.数据耦合、公共耦合、标记耦合、控制耦合的耦合性从低到高的顺序是（ ）

A、数据、公共、标记、控制

B、数据、标记、控制、公共

C、控制、数据、标记、公共

D、控制、数据、公共、标记

72. 以下哪种方式最不适合在需求频繁变更的项目采用？（ ）

A. 改进需求变更流程，提高需求管理的规范性。

B. 快速进行原型设计，并做好需求确认，降低需求变更成本。

C. 采用灵活的软件架构，提高可扩展性。

D. 采用瀑布开发模型，让开发过程有序、开发任务合理分配。

73. 下面哪一种提高软件生产效率的方法比较能体现面向服务架构的软件? （）

A. 通过简洁达到清晰（Clarity via conciseness）；

B. 代码合成（Synthesis）；

C. 可重用（Reuse）；

D. 自动化和工具（Automation and Tools）

74. 接口设计的主要任务是（）

A. 定义软件的主要结构元素及其之间的关系

B. 确定软件涉及的文件系统的结构及数据库的表结构

C. 描述软件及外部环境之间的交互关系，软件内模块之间的调用关系

D. 确定软件各个模块内部的算法和数据结构

75. 软件系统开发中使用MVC模式开发的优点，不包括（）

A. 有利于代码重用

B提高系统的运行效率

C提高系统的开发效率

D提高系统的可维护性

76. 下面中，哪个不是Ruby中的循环语句之一？（ ）

A. redo

B. until

C. unless

D. yield

77. 内聚是衡量模块内部聚合能力的量度。内聚越高，说明模块内各成分彼此联系的程度

A 越松散

B 彼此相等

C 越紧密

D 没有联系

78.软件设计中划分模块的准则之一是（）

A 高内聚低耦合

B 高耦合低内聚

C 高内聚高耦合

D 低耦合低内聚

79. Linux环境下如何编译一个.c文件。

A. touch test.c

B. gvim test.c

C. gcc -o test.c

D.gcc -c test.c -o test.o

80. 模块的内聚性最高的是（）

A.逻辑内聚

B.时间内聚

C.偶然内聚

D.功能内聚

81. “系统中每一个元素必须有着单一的、清晰的、明确的职责”是（C）原则的表述？（2分）

A. REST

B. SAAS

C. DRY

D. SOA

82. 下列有关 ruby 中 require,load,include,extend 说法错误的是

A:require 在加载一个文件是不需要加扩展名，如 require ’test’

B:load 跟 require 类似，但只能加载一次

C:extend 引入的方法会成为类方法

D:include 引入的变量会成为实例变量

83. 下列模块独立性最强的是(A)

A、非直接耦合

B、数据耦合

C、公共耦合

D、内容耦合

## ⯀ 软件架构和应用框架

84. 下面哪些是表达了 BDD 的不足。

i. 与客户交互沟通的成本。

ii. 客户满意不等于该软件有一个好的架构。

iii. 编写功能代码前，先写测试代码。

iv.

A. i and ii   
B. iii and iv   
C. ii and iii   
D. i, ii, iii, and iv

85. 在面向对象的高层设计体系结构中， (C) 是用来描述用户一般使用软件习惯的部分

A. 人机交互界面

B. 问题论域

C. 系统交互界面

D. 数据管理

86. 软件第二定律不包含以下哪个要素

A 时间

B 角色

C 作什么

D 怎样做

87. 以下哪项不属于软件工程九大任务

A 软件需求和应用场景

B 软件迭代和持续集成

C 软件架构和应用框架

D 软件验证和自动测试

88. 关于代码重构，以下说法错误的是

A 可能发现新的bug

B 可能会引起功能的改变（增加或减少）

C 增强代码可读性和可维护性

D 会改变既有代码的设计

89. 下面关于BDD的说法不正确的是（B）

A. 有相关的工具支撑。

B. 先测试后代码

C. 改变团队的工作模式，侧重多方面的协作。

D. BDD的技术基础是软件测试框架和应用软件框架。

90. 一个服务直接访问另外一个服务的数据时可能出现不稳定的情况一般是以下哪一

种情况的特点之一：

|  |  |  |
| --- | --- | --- |
|  |  | A. Rails 应用框架 |
|  | B. 面向服务的架构 |
|  | C. 面向对象编程 |
| D. 敏捷开发过程 |

91. 假设一个 Web APP 网站要开发一个移动 APP 版软件，若原有网站的架构设计采用了 MVC 架构，改版工作量大的地方是：

|  |  |  |
| --- | --- | --- |
|  |  | A. models |
|  | B. views |
|  | C. controllers |
| D. schema |
|  |  |  |

## ⯀ 设计模式和代码重构

***92. “系统中的每一个知识（功能或特性）必须有单一的、无二义和明确的表示。”是对下面的哪条原则的表达？***

A. REST

B. SAAS

C. SOA

D. DRY

93. Model-View-Controller (MVC) 架构/设计模式有什么好处？

A. 支持多用户存取和更新模型数据，对每个用户提供各自的视图。

B. 确保模型与视图有一对一的映射。

C. 为了调试目的，提供模型和控制器的调试窗口。

D. 像其他设计模式一样，遵循这类架构可以得到更简洁的代码。

94. 下面哪一种提高软件生产效率的方法比较能体现面向服务架构的软件?

A. 通过简洁达到清晰（Clarity via conciseness）

B. 代码合成（Synthesis）

C. 可重用（Reuse）

D. 自动化和工具（Automation and Tools）

***95. 假设影评网站 RottenPotatoes.com 要增加'filming location' 属性值到 movie 模型中。修改 MVC 哪个部分的工作量大?***

A. Movie 模型

B. 显示影片信息的 view 页面

C. 在控制器中增加新的 movie 实例

D. 以上所有都不是

96. UML是软件开发中的一个重要工具，它主要应用于哪种软件开发方法（C）

A．基于瀑布模型的结构化方法

B．基于需求动态定义的原型化方法

C．基于对象的面向对象的方法

D. 基于数据的数据流开发方法

97. 可行性分析研究的目的是（A）

A.项目值得开发否

B.争取项目

C.开发项目

D.功能内聚

98. 下面代码中第 4 行做什么?

1 class PostsController < ActionController::Base   
2 def add\_post   
3 # ...code to create new blog post...   
4 flash[:notice] = "Post added successfully"   
5 redirect\_to posts\_url   
6 end   
7 end

(i) 写消息到日志文件。   
 (ii) 这个方法执行后在 view 中提示。   
 (iii) 保存信息直到下一个请到来。

A. (ii) only

B. (iii) only

C. (i) and (iii)

D. (ii) and (iii)

99.下面的（D）不是良好编码的原则。

A．在开始编码之前建立单元测试

B. 建立一种有助于理解的直观布局

C．确保注释与代码完全一致

D．保持变量名简短以便代码紧凑

***100. 如果在 controller 方法中设置了一个实例变量，该值可以被什么模块读取。***  A. views 不能读取。

B. Views 可以用它，但仅限于这次请求处理。

C. Views 可以用它，仅限于这次请求和下次请求。

D. Views 可以用它，包括这次请求以后的所有请求。

101. 请阅读理解下面的代码



|  |
| --- |
| 下面哪个方式可以正确地把一个用户的信息加入到数据库中？ |

A. (i) and (ii)

B. (i) and (iii)

C. (ii) and (iii)

D. (i), (ii), and (iii)

102. 在课程管理系统中增加一个查询条件。除了按原有的 credit 列排序外，还增加一个域：search terms 在勾选框里。目的是让相应的 controller 的方法在params[:search\_terms]里读到这个选择。参数格式为：

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | params = {:credit => 3, :search\_terms => "software\_engineering" } |     请问下面哪种方式可以实现这样的效果? |

1. <input type="text" params="search\_terms"/>
2. <input type="text” name="search\_terms"/>
3. <input type="text" id="search\_terms"/>
4. <input type="text" name="params[search\_terms]"/>

103. RESTful 路由：假设 index 方法接收到带有上一个请求中显示的参数哈希值的表单提交，那么下列哪种方法将被视为处理返回的文本搜索过的电影列表结果的最简单方法？

A. 不渲染任何视图;而重定向到 movies index 页面，URI 如下/index?search\_terms=world&rating[PG]=1

B. 渲染当前的 movies index 页，使用匹配 “world”的结果, 过滤条件为release date 和 rating "PG", 但仍然使用统一个 URI /index

C. 建立新的 controller 方法 MoviesController#search, 建立新的 route match '/search/:ratings/:search\_terms', 'MoviesController#search' 然后重定向到/search/PG/world .

D. 渲染 XML 响应表示 movies 匹配了“world” 而且 ratings 过滤, “G”, “PG”, and “PG­13”

104. 下面哪个说法是错误的？

A. 使用更多设计模式的软件不一定性能更好.   
B. 精心设计的软件可以发展到模式成为反模式的程度。   
C. 过早地尝试应用设计模式可能与应用它们太晚一样糟糕。

D. 大多数设计模式特定于编程语言的特定子集。

105.面向对象的开发方法中，（ C ）将是面向对象技术领域内占主导地位的标准建模语言。

A．Booch 方法

B．Coad 方法

C．UML 语言

D．OMT 方法

106. 关于 P＆D 维护的哪一项声明（如果有的话）是假的？

A. 维护成本通常超过 P＆D 的开发成本   
B. 与 P＆D 变更请求相当的 Agile 是用户故事; 等价的变更请求成本估算是

积分; P＆D 版本就像迭代一样

|  |  |
| --- | --- |
| C. | Agile 生命周期类似于 P＆D 维护生命周期：增强工作软件产品，与客户合作与合同谈判，不断响应变化 |

D. 以上所有都属实

107. 软件项目可能过度超出预算或延迟计划的最佳预测因素是：

|  |  |
| --- | --- |
| A. 项目的范围或规模非常大  B. 项目使用 P＆D 方法而不是 Agile | |
| C. | 项目使用 Agile 方法而不是 P＆D |

D. 项目的测试/ QA 由一个单独的团队而不是开发人员完成

108. 鉴于小型软件项目与课堂上提供的大型软件项目的成功率，我们所涵盖的哪些技

术将为我们提供从许多小型项目构建大型项目的最佳机会？

1. 在计划和文档生命周期的设计阶段开发的软件体系结构   
   B. 模型 - 视图 - 控制器设计模式   
   C. 外墙设计模式   
   D. 遵循面向服务的体系结构

109. 关于类遵守单一责任原则（SRP）说法正确的是？

A. 一般而言，我们预计会看到不良凝聚力得分与差的 SOFA 指标之间的相关性

|  |  |
| --- | --- |
| B. 低内聚是分离类的可能指标 | |
| C. | 如果一个类尊重 SRP，它的方法可能会尊重 SOFA |

D. 如果一个类的方法尊重 SOFA，那么该类可能会尊重 SRP

110. 不恰当的亲密设计气味，有时在单元测试中表现为 Mock Trainwreck，通常表示违反以下哪种 SOLID 原则：

A. Single Responsibility   
B. Open/Closed   
C. Liskov Substitution   
D. Injection of Dependencies   
E. Demeter

111. 您希望创建相关对象的族，以便可以互换地配置应用程序。 哪种设计模式最适合使用？

|  |  |
| --- | --- |
| A. 抽象工厂模式  B. 组合模式 | |
| C. | 观察者模式 |

D. 模板方法模式

|  |  |
| --- | --- |
| 112. 根据开闭原则, 应该为 开放但是( | )时应该关闭. |

A. class; extension; modification   
B. method; modification; polymorphism   
C. class; delegation; polymorphism   
D. method; stubbing; overriding

113. 以下哪一个不属于软件工程的九项任务之一 (D)

A. 业务建模和数据模型

B. 接口设计和前端编程

C. 性能评价和设计优化

D. 软件产品移交用户

114. 以下哪一个不属于 CI/CD 的内容 (D)

A. 持续集成

B. 持续交付

C. 持续部署

D. 自动化测试

115. 以下哪一个不属于 A/B 测试的优点 (B)

A. 可以提高产品设计和运营水平

B. 实现成本较低

C. 可以降低新产品或新特性的发布风险

D. 在少量数据中测试依然有效

116、 需求分析阶段的任务是确定（D）

A．软件开发方法

B.软件开发工具

C.软件开发费

D.软件系统的功能

117. 下面关于 JavaScript 中函数的说法正确的是（多选）

|  |  |
| --- | --- |
| A. 它可以是匿名的  B. 它是该语言中的一级对象  C. 它可以与其他功能同时执行  D. 它可以是 JavaScript 对象中属性的值 | |
| E. | 当没有参数调用时，括号是可选的（与 Ruby 函数一样） |

118. 什么情况下可以只在客户端验证全部输入信息（而在服务端不做）（单选）

|  |  |
| --- | --- |
| A. 确认浏览器 javascript 启用  B. 确认浏览器 javascript 启用，且表单通过安全链接提交 | |
| C. | 服务器必须总是执行验证，即使客户端有 javascript 验证。 |

119. 执行后台请求并根据响应更新适当位置的 html 页面的能力依靠的是（单选）：

A. 浏览器内置 javascript 翻译器

B. 浏览器使 DOM 对 Javascript 可用

C. 一个对 XmlHttpRequest 的 Javascript 绑定

D. 浏览器中对 XML 的解析

E. 浏览器对 JSON 的解析

120. 以下哪一个不属于软件工程学科的三类根本问题？

A. Ensure Quality

B. Good Architecture

C. Value for User

D. Appropriate Document

121. 以下哪一个不属于A/B 测试解决的问题？

A. 降低网站跳出率

B. 基于现有的访客流量获得更高的收益回报

C. 做出低风险的改动

D. 为增加软件功能过渡

122. 软件测试是为了（ B ）而执行程序的过程。

A. 纠正错误

B. 发现错误

C. 避免错误

D. 证明正确

123. 以下哪一个不能用作包管理工具 (D)

A.NPM

B. Gem

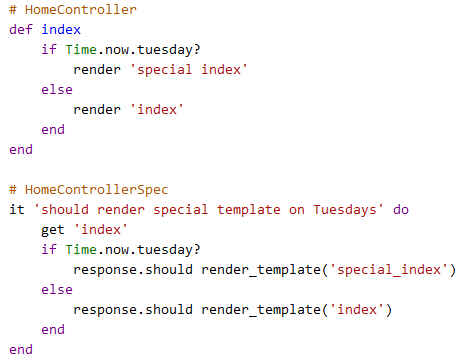
C. PIP

124. 最终莎莉意识到， 她的网站的用户可能希望创建一个黑名单。 用户不希望和他们的黑名单人员匹配。 然而， 她意识到此功能不是很重要， 于是决定在几个月的后面实现它。 最后她写了功能测试。 莎莉显然没有遵循 FIRST 原则的哪一个？

A.快   
B.独立   
C.可重复   
D.自我检查   
E.及时

125. 莎莉希望她的网站在每个月的第一个星期二有一个特别的布局，她有以下的控制

器和测试代码:



明显没有遵循的 FIRST 原则是什么?   
A. Fast   
B. Independent   
C. Repeatable   
D. Self-checking   
E. Timely

126. 当探索一段遗留代码时，下列那些技术有可能会有帮助?

|  |  |
| --- | --- |
| i. | 创建一个应用的草稿分支 |

ii. 观看用户如何使用该软件   
iii. 运行测试集合（如果有的话）   
iv. 检查非正式的设计文档（例如 wikis，提交日志等）   
A. (iii) and (iv) only   
B. (i) and (ii) only   
C. (i), (ii), and (iv) only   
D. (i), (ii), (iii), and (iv)

127. 在对遗留代码进行高级别的探索时，下列哪个表现你可以认为是正确的?

A. 探索指的是在合理的期限内去修正代码中一些小的不够“漂亮的”问题 B. 当你发现了类之间重要的集合时，你将能轻松地为它们建立桩。 C. 你应该创建说明现在应用是如何工作的测试，即使这些测试是用于测试bug 和一些丑陋的特性。   
D. 如果设计文档是可获得的，你可以用他们来建立当前代码工作的“基准情况” 。

128. 关于重构，下列陈述正确的是( )

A. 改进代码结构是主要目标

B. 重构可以改善软件的质量、性能

B. 测试对重构很重要

D. 重构总是减少代码

129. 根据开闭原则， \_\_\_\_应该\_\_\_\_开放，但是封闭反对 \_\_\_\_。

A. 类; 扩展; 修改

B. 方法; 修改;多态性

C. 类; 委派;多态性

D. 方法;存根; 方法的重写

130. 开发语言都会演化出应用框架，关于框架下面说法错误的是

A.开发效率会更高

B.可扩展性和可维护性会更高

C.通过框架能深刻理解逻辑

D.轻量级和重量级框架开发效率没有太大区别

D的话软件必须要有用户这个要素，所有潜在价值与使用过程绑定

131. 下列哪个不是准备修正遗留代码的挑战？

A． 它可能能够在生成环境中运行，但是在开发环境中难以

运行

B． 在测试集合中，可能有失败或过时的测试。

C． 用户可能并不清楚什么样的改进是最重要的。

D． 可能存在没有被用户用例引发或者测试覆盖的bug

132. 对比下面相同代码的两个版本，下面陈述正确的是:

# version 1: # version 2:   
def foo(x,y) def foo(x,y)   
 if x if x   
 if y check\_y\_z(y)   
 z() else   
 end check\_y\_w(y)   
 else end   
 if y end   
 w()   
 end def check\_y\_z(val) ; z() if val ; end   
 end def check\_y\_w(val) ; w() if val ; end   
end

A. 版本 1 有更高的圈复杂度。

B. 版本 1 有更多的测试接缝。（通过所谓的接缝(Seam)，可以将具有逻辑的部分与承上启下的代码分离开，然后将单元测试注入到接缝中，形成对有逻辑代码的保护网）

C. 版本 1 有更低的 ABC 得分

D. 版本 1 不能通过 100% C2 覆盖测试。

133. 关于重构，下面哪个陈述是正确的？

A. 重构提高测试覆盖率   
B. 在重构期间，永远不应该有失败的测试   
C. 改进代码结构是基本目标   
D. 重构总是减少代码量

134. 关于方法级的重构，\_\_\_\_\_代码味道是可能存在的在其他三个代码都存在的情况下.

A. 长方法   
B. 方法做了不止一件事   
C. 方法有很多参数   
D. 方法在抽象层次间来回跳跃

135. 软件设计中划分模块的一个准则是（）

A.低内聚低耦合

B.低内聚高耦合

C.高内聚低耦合

D.高内聚高耦合

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 136. 假设一个 Web APP 网站要开发一个移动 APP 版软件，若原有网站的架构设计采用了MVC 架构，改版工作量大的地方是\_\_\_\_\_\_\_\_\_\_。 | | | | | |
| A. models |  | B. views |  | C. controllers | D. schema |
|  |  |  |  |  |  |

137. 下面那个说法不属于设计准则

A.提高模块的内聚，降低模块间的耦合

B.降低模块接口的复杂程度   
C.模块大小要适中

D.模块要用重用性

138. 下面说法正确的是

A.模块的作用与在模块的控制域之内

B.模块的控制域在模块的作用域之内

C.模块的作用域和模块的控制域有时相同

D.模块的作用域和模块的控制域都是一种层次结构

139. 若要保证开发出来的软件布局优化，而且能尽早发现设计上的错误，应采用

A.自顶向下的开发方法

B.自底向上的开发方法

C.渐增式的开发方法

D.非渐增式的开发方法

140. 模块的初始化工作和模块处理意外故障均属于

A.时间内聚

B.顺序内聚

C.功能内聚

D.逻辑内聚

141. 下列哪种属于发布-订阅的架构风格？

A、观察者模式   
B、策略模式   
C、工厂模式   
D、代理模式

142. Ruby 的三种典型变量的写法不包括下列哪一项

A. &a   
B. @a   
C. @@a   
D. a

143. 下列哪个写法是 a+x\*y 的显示 send 调用方法

A． a.+(x.\*(y))   
B． a.send(:x, +.send(:x, y))   
C.a.send(:+, x.send(:x, y))   
D.x.send(:\*, y.send(:+, a))

144. 根据 DIP 原则，下列哪种写法更好

A． @vips = User.find\_vips   
B． # in controller

@vips = User.find\_vips

C． @vips = User.where('group="VIP"')   
D． 上述写法效果相同

145. 下⾯匹配正则表达式 /^\d{1,3}\.\w{1,3}\.\d{1,3}\.\w{1,3}$/ 的是

A: "192.168.1.1"   
B: "192.abcd.1.1"   
C: "192.abc.1.abcd"C: "192.abc.1.abcd"   
D: "192.abc.1."

146. .Ruby 采⽤诗歌模式，在没有歧义的前提下可以尽可能的简化表达，省略没有必要 的括号，下⾯调⽤函数 plus 的⽅法中，错误的是   
def plus (a,b)   
a+b   
end   
 A: plus(3,5)

B: plus 3, b=5

C: plus b=3, c=5

D: plus a:3, b:5

147. 下⾯⼀段代码违背了哪⼀项原则   
class Report   
def output   
formatter =   
case @format   
when :html   
HtmlFormatter.new(self)   
when :pdf   
HtmlFormatter.new(self)   
end   
end   
end

A: 开闭原则   
B: ⾥⽒替换原则   
C: 接⼝隔离原则 C: 接⼝隔离原则   
D: 依赖注⼊原则

148. 假如我们构建了⼀个选课系统，我们希望新增 ⼀个功能，让学⽣可以对⼀⻔课 程进⾏评价、打分，其它学⽣也可以看到，那么整个系统中改动最多的部分是

A: 路由   
B: 数据库   
C: 控制器   
D: 视图\⻔⾯

149. 本课程中涉及的 sota 的全称为？

A、state-of-the-art   
B、School of the Arts   
C、Statewide Organization of Third-world Artists   
D、以上都有可能

150. 软件设计中划分模块的原则是（）

A、高内聚低耦合   
B、低内聚高耦合   
C、低内聚低耦合   
D、高内聚高耦合

151. 问：在用 rails 开发一个项目时，要与页面布局中的链接进行交互式测试，测试文件一般应放在 test 的哪个文件夹中？

A. controllers

B. fixtures

C.channels

D. integration

152.下列模式中，属于行为模式的是（）。

A . 工厂模式   
B . 观察者   
C . 组合模式   
D . 以上都是

153.下面那些 UML 图描述系统行为（）

A.用例图   
B.类图   
C.对象图

154.在 UML 的类图中，描述整体与部分关系的有 （）。

A.泛化关系   
B.聚合关系   
C.依赖关系

155 软件设计中划分模块的一个准则是()。

A.低内聚低耦合

B.低内聚高耦合

C.高内聚低耦合

D.高内聚高耦合

156 .PDL 是软件开发过程中用于( )阶段的描述工具。   
A.需求分析 B.概要设计 C.详细设计 D.编程

157. 下列语言中哪个语言是低级语言( )   
A. PASCAL B.VISUAL BASIC C. C++ D.汇编语言

158.以下哪几个是软件的设计原则？

A、模块化

B、抽象

C、封装

D、松耦合与高内聚

E、接口和实现分离

F、充分性和完整性

159.以下设计模式中哪些属于创建型模式？   
A、抽象工厂模式 B、生成器模式 C、工厂模式 D、原型模式 E、单例模式 F、适配器模式（结构性模式）G、修饰模式（结构型模式）H、外观模式（结构型模式）I、状态模式（行为型模式）

160、如何比较两个文件或当前文件和以前版本的修订？   
A、git diff B、git compare C、git clone D、git checkout

161、下面哪一个不属于 MVC 模式中的对象？   
A、Model B、View C、Collection D、Controller

162、Servlet 在容器中经历的阶段，按顺序为：   
A、服务、加载、初始化、卸载、销毁   
B、加载、初始化、服务、销毁、卸载   
C、初始化、服务、销毁、加载、卸载   
D、服务、卸载、加载、初始化、销毁

163. ruby 是什么模式的编程语言？   
A 基于逻辑的编程语言 B.面向对象编程语言 C 函数式语言 D 多范型语言

164.FIRST 原则中的 I 代表什么？   
A informative B imperative C implemented D isolated

165.为了尽可能分离可能发生变化的代码，编程时应当针对\_\_\_\_并且尽可能使用\_\_\_\_。 A 接口/组合与委派 B 实现/继承 C 接口/继承 D 实现/组合与委派

166.对观察者模式，以下叙述不正确的是（）。   
A 必须找出所有希望获得通知的对象。   
B 所有的观察者对象有相同的接口。   
C 如果观察者的类型相同，目标就可以轻易地通知它们。   
D 在大多数情况下，观察者负责了解自己观察的是什么，目标需要知道有哪些观察者依赖自己。

167.下列哪一个 git 命令用于将所有文件提交至暂存区？（）   
A git add .   
B git push   
C git commit   
D git manage

168、以下哪一条不是敏捷开发者崇尚的价值观？（）   
A 流程和工具高于个体和互动   
B 工作的软件高于详尽的文档   
C 客户合作高于合同谈判   
D 响应变化高于遵循计划

169、在 Ruby 中，类的构造函数是（）方法。   
A create   
B initialize   
C new   
D class   
***170、以下程序输出的结果是（）。***

|  |
| --- |
| myStr = String.new("THIS IS TEST")  foo = myStr.downcase  puts "#{foo}" |

A THIS IS TEST   
B THISISTEST   
C this is test   
D This Is Test

171.按照\_\_\_\_可以将软件生存周期过程分为基本过程、支持过程和组织过程   
A. 软件开发活动的层次关系   
B. 软件开发工作的主体   
C. 软件开发项目的结构   
D. 软件开发任务的重要程度

172.对于变换设计的基本步骤，下列顺序正确的是？   
① 确定输入、变换、输出三部分之间的边界   
② 系统模块结构图顶层和第一层的设计   
③ 复审并精化系统需求模型   
④ 自顶向下，逐步求精   
A.①②③④   
B.②①③④   
C.③①②④   
D.①③②④

173.敏捷开发者崇尚的价值点是   
A. 过程和工具胜于个人与团队交流   
B. 可工作的软件系统胜于全面的文档   
C. 合同谈判胜于与客户协作   
D. 遵循计划胜于响应需求变化

174. 一个服务直接访问另外一个服务的数据时可能出现不稳定的情况一般是以下哪一种情况的特点之一：   
A. Rails 应用框架   
B. 面向服务的架构   
C. 面向对象编程   
D. 敏捷开发过程

175. 下面哪一种提高软件生产效率的方法比较能体现面向服务架构的软件?   
A. 通过简洁达到清晰（Clarity via conciseness）   
B. 代码合成（Synthesis）   
C. 可重用（Reuse）

D. 自动化和工具（Automation and Tools）

176. 下面哪个说法是对的（根据 Ruby 语法要求）?

i. 局部变量以 $开头

ii. 实例变量以 @开头

iii. 类变量以 @@开头

A. i & iii

B. ii & iii

C. i & ii

D. i, ii & iii

177.以下哪个不是软件工程的根本问题   
A、构思创造发明有价值的软件系统或产品和找到检验其质量的理论和方法。   
B、寻找项目开发各项资源最优化分配和平衡效益成本的理论和方法。   
C、探讨测算软件设计实现和部署运营的效益模型的理论和方法。   
D、探寻应对领域问题和软件需求变化的可动态升级软件模块和架构的理论和方法。

178. 以下哪项不是 Ruby 的重要贡献   
A、一切都是对象。   
B、每个操作都是调用某些对象的方法并返回一些值。   
C、变量不需要写类型。   
D、所有编程都是元编程。   
179、瀑布模型存在问题是（ ）   
A、缺乏灵活性。   
B、用户容易参与开发。   
C、用户与开发者易沟通。   
D、适用可变需求。

180. 瀑布模型的存在问题是（ ）   
A．用户容易参与开发 B．缺乏灵活性   
C．用户与开发者易沟通 D．适用可变需求

181. 以下哪一项不是面向对象的特征（）   
（A）多态性（B）继承性（C）封装性（D）过程调用

182.一个好的架构应该是（）

A 能不依赖于某个技术的实现

B 是针对某一类业务而设计的   
C 可以得到更简洁的代码   
D 不需要用专业的业务术语来描述

183.solid 原则说法正确的是（）

A l 代表接口隔离原则   
B 每一个类中可以包含多种相似功能符合 solid 原则   
C 单一职责原则是指每个类应该只有一个职责   
D 高凝聚是开放封闭原则的内容

184.MVC 架构是什么的简称（）

A Model Vision Controller   
B Manage View Condense   
C Manage Vision Condense   
D Model View Controller

185.架构负责解决的问题有 （）   
①代码如何组织   
②运行时“模块”如何交互   
③编写测试代码   
A, ①   
B, ③   
C, ①②   
D, ②③

186.关于重构的说法正确的是（）

A 只能在测试之后进行重构   
B 重构可以提高代码的可读性，提高代码质量   
C 重构不能找出 Bug   
D 重构总是可以减少代码量

187.瀑布模型的存在问题是（ ）

A．用户容易参与开发 B．缺乏灵活性

C．用户与开发者易沟通 D．适用可变需求

188.开发软件所需高成本和产品的低质量之间有着尖锐的矛盾，这种现象称做

( )

A.软件工程 B.软件周期

C.软件危机 D.软件产生

189. Jackson 图中，模块框之间若有直线连接，表示它们之间存在( )。

A. 调用关系 B. 组成关系 C. 链接关系 D. 顺序执行关系

190. 软件详细设计的主要任务是确定每个模块的( )。

A. 功能 B. 外部接口 C. 算法和使用的数据结构 D. 编程

191．为了提高软件的可维护性，在编码阶段应注意（ ）

A.保存测试用例和数据 B.提高模块的独立性

C.文档的副作用 D.养成好的程序设计风格

192.软件重构关注的是什么（）   
A.软件体系结构 B。模块细节 C.软件性能

193.Github 哪一年上线？

A 1991 年 B 2002 年 C 2005 年 D 2008 年

194.Ruby 提供了多种字符串的表示方法，下面哪一种是错误的？

A str=['在线学习乐园']

B str="\n{www.itzcn.com}"

C str=%/\t 免费教学和视频\n/

D str="窗内网

195. 假设有两个 HTTP 请求的 URI 是一样，但调用的方法(GET 与 POST)不同。对于Rails 路由而言，下面哪个说法是对的。

A. 两个请求可以对应不同控制器的方法，但不是必须的。

B. 两个请求必须对应不同控制器的方法。

C. 两个请求必须对应相同控制器的方法。

D. Rails 将给该请求抛出一个错误信息

196.系统中的每一个知识（功能或特性）必须有单一的、无二义和明确的表示。”是对下

面的哪条原则的表达？

A. REST

B. SAAS

C. SOA

D. DRY

197.开发软件所需高成本和产品的低质量之间有着尖锐的矛盾，这种现象称作（ ）

A、软件工程

B、软件周期

C、软件危机

D、软件产生

198. 软件是（）

A.处理对象和处理规则的描述

B 程序

C 程序及其文档

D 计算机系统

199. 在软件工程中，高质量的文档具备完整性、一致性和（）

A 统一性

B 无二义性

C 安全性

D 组合性

200. 在软件生存周期中，（ ）阶段必须要回答的问题是“要解决的问题是做什么？”。

A 详细设计 B 可行性分析和项目开发计划 C 概要设计 D 软件测试

201. 下列哪项不属于 SOLID 原则？（）

A 单一职责原则 B 开放封闭原则 C 里氏替换原则 D 依赖拒绝原则

202. 下列选项中属于产品的内部属性的是（）。

A 模块耦合度 B 软件可靠性 C 软件有效性 D 软件可维护性

203.下面不属于软件工程的 3 个要素是()

A 工具 B 过程 C 方法 D 环境

204. 以下哪个选项是在 Linux 系统下安装 Django

A．python setup.py install

B．pip install Django

C．apt-get update && apt-get install -y --no-install-recommends

D．"build-essential=12.4ubuntu1

205.在命令控制台产看 Ruby 的版本应该使用哪个语句？

A．ruby -h

B．ruby -version

C．ruby -v

D．ruby -s

206. 调用什么方法可以去除用户输入的换行符

A．chomp

B．trim

C．to s

D．to i

207. 下面 ruby 使用方法错误的是

A．ruby hi.rb

B．ruby –e 'print "hi" '

C．ruby –h hi.rb

D．ruby –c hi.rb

208. 评价一个软件工程的关键词中，错误的是

A．创新性

B．记忆性

C．分析性

D．排他性

209. 设计模式通常不包括()

A.创建型模式

B.结构型模式

C.行为型模式

D.操作型模式

210. Ruby 语言属于一种( )的语言。

A.面向对象

B.面向过程

C.函数式

D.面向服务

211. Django 作为一种 Web 框架，是用哪种语言编写的( )

A.Python

B.Java

C.C++

D.Ruby

212. ruby 变量前有一个@符号，意为：

A 实例变量 B 类变量 C 局部变量 D 全局变量

213. 下列哪个 docker 命令可以列出正在运行中的镜像？

A docker images ls B docker ps C docker build . D docker exec -it

214.命令 put :a.class 得到？（ruby）

A Symbol B Integer C String D Object

215. ruby 的函数式语言中，.map 的作用是：

A 遍历 B 改写 C 筛选

216. 下面关于 rake 的说法错误的是：

A rake 相当于 C 的 make

B rake 利用了 ruby 语言的 DSL 性质，完全用 ruby 语言书写

C make 不依赖于 shell

D make 只更新修改过的文件，减少了大量的重复劳动

217. 下面关于 DevOps 描述错误的是：()

A.DevOps 突出重视了软件开发人员和客户之间的沟通合作。

B.DevOps 的核心是角色的分工，而不是组织架构变化。

C.DevOps 涉及持续集成、持续部署、持续测试等流程。

D.DevOps 一个好处就是会改善公司组织文化、提高员工的参与感。

***218. 下面哪些用户故事的叙述是满足的 SMART 原则要求的：()***

A.用户可以翻译外语。

B.作为一个用户，我可以通过输入待翻译句子，选择要翻译的目标语言以及点击翻译按钮来

获取翻译后的句子，以便我可以快速知道外语的意思，方便学习外语。

C.APP 应该有好的用户界面。

D.APP 应该有好的用户界面和快的响应旪间。

219.下面哪个选项不是 ruby 的最重要支柱：)

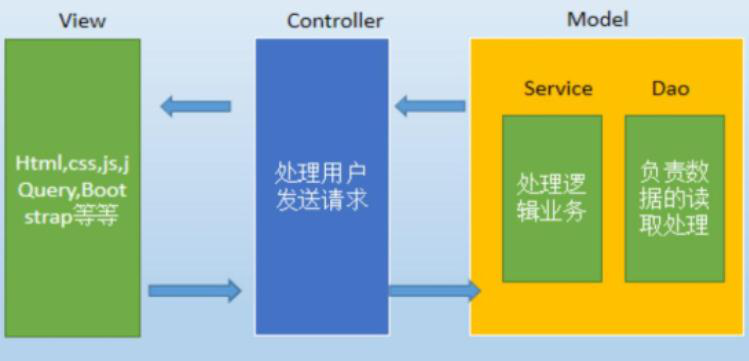
A.一切都是对象。

B.所有编程都是元编程。

C.每一个操作都是调用某些对象的方法，并且返回值。

D.具有内存管理机制。

220.根据下图以及自己对 MVC 模式的理解，当用户点击网页时，正确的代码调用顺序应该是（）？



A.View>Controller>Model

B.Controller>Model>Controller>View

C.Model>Controller>View

Controller>View>Controller>Model

221.软件工程的出现主要是由于（ )

A.程序设计方法学的影响

B.其它工程科学的影响

C.软件危机的出现

D.计算机的发展

222. 软件开发常使用的两种基本方法是结构化和原型化方法，在实际的应用中，它们之间

的关系表现为（）

A.相互排斥

B 相互补充

C 独立使用

D.交替使用

223.软件开发的结构化生命周期方法将软件生命周期划分成（）

A.计划阶段.开发阶段.运行阶段

B.计划阶段.编程阶段.测试阶段

C.总体设计.详细设计.编程调试

D.需求分析.功能定义.系统设计

224. 完整的软件结构通常用\_\_\_\_图来表示。（）

A．IPO 图 B.HIPO 图 C.PAD 图 D.DFD 图

225. 一个好的软件架构需要满足（D）

A.不依赖技术实现

B.用领域知识描述

C.解耦合

D.以上都需要满足

226. 一个好的科学理论需要满足（）

A.具有好的描述能力

B.有预见性

C.能提供精准调控的依据

D.以上都需要满足

227.下列过于代码重构的说法正确的是\_\_。

A. 对于过长的方法（long method），可以用 Extract Method（提炼函数）进行重构。

B. 代码重构能够提高测试覆盖率。

C. 在重构期间，永远不应该有失败的测试。

D. 重构总是可以减少代码量。

228. 螺旋模型共有四个阶段，这些阶段的顺序应该是（ ）

i. 确定本次迭代的目标和约束

ii. 开发和验证本次迭代原型

iii. 计划下一次迭代

iiii. 评估各种备选方案并加以确认，化解风险

A i. iiii. ii. iii.

B i. ii. iii. iiii.

C i. ii. iiii. iii.

D ii. i. iiii. iii.

229. 在观察者模式中，表述错误的是（ ）

A.观察者角色的更新是被动的

B.被观察者可以通知观察者进行更新

C.观察者可以改变被观察者的状态，再由被观察者通知所有观察者依据被观察者的状态进行

D.以上表述全部错误

230. 以下哪个 Ruby 表达式无法匹配字符串中的 Email()

A、"My email is 12345@qq.com." =~ /\w+([-+.]\w+)\*@\w+\.\w+([-.]\w+)\*/

+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*\w+([-.]\ w+)\*/

B "My email is 12345@qq.com." =~ /\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*\w+([-.]\ w+)\*/

C、"My email is 12345@qq.com." =~ /\w+([-+.]\w+)+@\w+([-.]\w+)+\.\w+([-.]\w+)+/

D、"My email is 12345@qq.com." =~ /\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*/

231. 以下哪种不属于设计模式中的 Structural()

A、Composite B、Proxy C、Adapter D、Observer

232. 以下哪种编程语言不是面向对象的()

A、Scala B、Ruby C、Eiffel D、Clojure

233.使得在多个类中能够定义同一个操作或属性名，并在每一个类中有不同的实现的一种

方法是（）

A.继承 B.多态性 C.约束 D.接口

234. 假设一个 Web APP 网站要开发一个移动 APP 版软件，若原有网站的架构设计采用了

MVC 架构，改版工作量大的地方是：（）

A. models

B. views

C. controllers

D. Schema

235. 下列关于 ruby 类变量的说法哪些是正确的：（）

(a) 局部变量以$开头

(b) 实例变量以@开头

(c) 类变量以@@开头

A. (a) (b) (c)

B. (b) (c)

C. (a) (c)

D. (a) (b)

236. 软件⼯程⽅法学包含的三个要素为：⽅法、（）和过程。（C）

A.设计

B.概念

C.⼯具

D.定义

237. 在进行软件测试时，首先应当进行以下哪项测试，然后再进行组装测试，最后再进行有效性测试（）（2分）

A.单元测试

B.系统测试

C.集成测试

D.确认测试

238. 软件设计一般分为总体设计和详细设计，它们之间的关系是 ()。（2分）

A．全局和局部

B．抽象和具体

C．总体和层次

D．功能和结构

239. 模块本身以及所有直接或间接从属于它的模块的集合，称为该模块的 (B ) （2分）

A．作用域

B．控制域

C．定义域

D．限制域

240. Git 远程下载一个项目和它的整个代码历史( )

A.git init

B.git clone

C.git add

D.git checkout

241. Ruby 中引入需要频繁更新的外部文件( )

A.extend

B.include

C. load

D.require

242. 在 SOLID 理论原则中，什么是对扩展开放，对修改关闭。（）

A.SRP

B.OCP

C.LSP

D.ISP

243. 下面哪种提高软件效率的方法比较能体现面向服务架构的软件（）

A.通过简洁达到清晰

B.代码合成

C.可重用

D.自动化和工具

244. 要创建应用程序的 development 和 test 数据库，应使用下面命令（）

A. rails db:create

B. rakes db:create

C. rails new db

D. rake db:create

245. 什么是 bundler？

a）Rails 生成支架 User name:string email:string

b）自成 bundles of gems用于版本控制。

c）编辑 Gemfile，然后运行“捆绑安装”或仅运行“捆绑”。

d）Ruby 版本管理器。 安装和管理多个版本的 Ruby。

246. 调用（D）方法可以去掉用户输入的换行符。

A.trim

B.to\_s

C.to\_i

D.chomp

247. 以下哪个不属于 Ruby 的重要支柱（贡献）？（）

A.一切都是对象。

B.每一个操作都是调用某些对象的方法，并且通过返回一个值。

C.所有的的编程都是元编程。

D.拥有功能强大和完善的标准类库。

248. Ruby 中所有类的基类为

A, Object

B, BasicObject

C, BasicClass

D, ClassObject

249.以下对于 ruby 的说法正确的是 D

A, 所有的编程都是元编程

B, 一切都是对象

C, 每一个对象的操作都是调用对象的某些方法并且返回值

D, 以上都正确

250. 以下说法正确的是

①10 对 4 取余可用 10.modulo(4)表示

②ruby 不需要分号

③SOLID 五项原则中 O 表示 Open-minded

A, ①

B, ①②

C, ③

D, ②③

251. 以下说法正确的是

①使用 attr\_accessor 可以节约代码量

②ruby 中可以用；在同一行写多条语句

③BDD 不可以与 TDD 结合使用

A, ①

B, ①②

C, ③

D, ②③

252. 以下字符串输出方法正确的是

|  |  |
| --- | --- |
| A, name=”Yasuo” |  |

print name

B, print “Yasuo”

C, name=”Yasuo”

print name + “666”

D, 以上都正确

253. 软件生命周期中时间最长的是( )。

A. 总体设计 B. 需求分析 C. 软件测试 D. 软件维护

254. 如何初始化 lita bot 的功能文件夹。

A.lita new bot

B.lita create bot

C.lita bundle bot

D.lita handler bot

255. 如何在本地运行 lita 以测试自己的功能。

A.bundle exec lita

B.lita run

C.web: bundle exec lita

D.bundle

256. 在面向对象的设计中，我们应该遵循的设计准则除了模块化、抽象、低耦合、高内聚

外，还有()。

A.隐藏复杂性

B. 信息隐蔽

C. 经常类的复用

D. 类的开发

257. 软件维护困难的最主要原因是()

A. 费用低

B.人员少

C. 开发方法的缺陷

D.得不到用户支持

## ⯀ 接口设计和前端编程

258. 下面的描述中不属于 Ruby 特性的是\_\_\_\_\_\_\_\_\_\_。

A．Ruby 是一种功能强大的面向对象的脚本语言。

B．Ruby 遵守 GPL 协议并且是免费的。

C．Ruby 具有自动内存管理机制。

D．Ruby 是基于 MVC 架构的。

259. 下面的描述中不属于 Rails 特性的是\_\_\_\_\_\_\_\_\_\_。

A．Rails 是一个使用 Ruby 语言写的开源网络应用框架。

B．Rails 具有强大的反射机制与后设编程。

C．“不要重复自己”和“约定胜于配置”是 Rails 的设计原则。

D．Rails 具有实时映射技术和模板编程技术。

260.Ruby 提供了多种字符串的表示方法，下面哪一种是错误的？\_\_\_\_\_\_\_\_\_\_

A．str=['在线学习乐园']

B．str="\n{www.itzcn.com}"

C．str=%/\t 免费教学和视频\n/

D．str="窗内网"

261.在命令控制台要查看 Ruby 的版本应该使用命令\_\_\_\_\_\_\_\_\_\_。

A．ruby –v

B．ruby –version

C．ruby –h

D．ruby /?

262.下列不属于 RubyGems 提供命令的是\_\_\_\_\_\_\_\_\_\_。

A．gem list

B．gem install rails

C．gem –v

D．gem –update rails

263.下面关于 Ruby 编译器的使用方法，错误的是\_\_\_\_\_\_\_\_\_\_。

A．ruby hi.rb

B．ruby –e 'print "hi" '

C．ruby –h hi.rb

D．ruby –c hi.rb

264. 调用\_\_\_\_\_\_\_\_\_\_方法可以去除用户输入的换行符。

A．chomp

B．trim

C．to\_s

D．to\_i

265. 假设有两个 HTTP 请求的 URI 是一样，但调用的方法(GET 与 POST)不同。对于Rails 路由而言，下面哪个说法是对的。

A. 两个请求可以对应不同控制器的方法，但不是必须的。

B. 两个请求必须对应不同控制器的方法。

C. 两个请求必须对应相同控制器的方法。

D. Rails 将给该请求抛出一个错误信息

266. 下面哪个说法是对的（根据 Ruby 语法要求）?

i. 局部变量以 $开头

ii. 实例变量以 @开头

iii. 类变量以 @@开头

A. i & iii

B. ii & iii

C. i & ii

D. i, ii & iii

267. 如果执行下面代码的结果是什么?

a = SavingsAccount.new(100)

b = SavingsAccount.new(50)   
c = SavingsAccount.new(75)

[a,b,c].sort

A. 工作, 因为存款余额是数字，可以比较。   
B. 不工作，但传入比较方法后也许可以。   
C. 不工作，但如果在 SavingsAccount 定义了<=>方法后也许可以。   
D. 不工作， SavingsAccount 不是 Ruby 基本类型不可以比较。

***268. 当 Ruby 表达式 foo + bar 求值执行时，下面哪个说法对？***

|  |  |
| --- | --- |
|  | A. bar 作为一个参数传给 foo 的方法 + 。  B. foo 作为一个参数传给 bar 的方法 + 。  C. 出错，因为'+' 只定义在字符串和数字对象上. |
|  | D. foo 和 bar 相加。 |

269. OmniAuth 定义了一些应用程序必须提供的 RESTful 端点，以处理与各种第三方的

身份验证。 要添加新的身份验证提供程序，请创建与该提供程序一起使用的gem。 关于 OmniAuth，以下说法错误的是？

A. OmniAuth 本身符合 OCP   
B. 使用 OmniAuth 帮助您的应用程序遵循 OCP（关于第三方身份验证）

C. OmniAuth 是模板模式的一个例子

D. OmniAuth 是 Strategy pattern 的一个例子

270. 下面的描述中不属于 Rails 特性的是\_\_\_\_\_\_\_\_\_\_。

A．Rails 是一个使用 Ruby 语言写的开源网络应用框架。

B．Rails 具有强大的反射机制与后设编程。

C．“不要重复自己”和“约定胜于配置”是 Rails 的设计原则。

D．Rails 具有实时映射技术和模板编程技术。

271.针对接口编程，不要针对实现编程,是( )的表述

A.开闭原则

B.接口隔离原则

C.里氏代换原则

D.依赖倒转原则

## ⯀ 软件验证与自动测试

272. 为什么很多软件项目失败?

|  |  |
| --- | --- |
| (i)  (ii)  (iii)  (iv) | 软件不是用户所要的。  延迟交付。  软件超出预算。  软件演化后容易维护和升级 |

A. i and ii   
B. iii and iv   
C. i, ii, and iii   
D. i, ii, iii, and iv

|  |  |
| --- | --- |
| 273. 如果一个项目延期后，为什么说增加人手不是一个好主意。因为： | |
| A. 程序员太贵 |
| B. 不能用大团队来承建 SaaS 软件 |
| C. 交流成本会下降 |
| D. 让新程序员上手需要较多时间 |

274. 以下哪一项保证你有详尽的单元测试一段代码？

A. 100% C0 (statement) 覆盖率   
B. 100% C1 (branch) 覆盖率   
C. 100% C2 (path) 覆盖率   
D. 以上都不是

275. 为了测试在代码中引入了故意错误，以查看测试是否中断。这是哪种测试?

突变试验   
A. 突变试验   
B. DU 测试

C. 黑盒测试   
D. 模糊测试

276. 请按测试的代码量和复杂程度有低到高排序?

A. 单元测试，功能或模块测试，集成测试

B. 集成测试，功能或模块测试，单元测试

C. 单元测试，集成测试，功能或模块测试

D. 功能或模块测试，单元测试，集成测试

277. 关于 TDD 哪一项是错误的?

A. 视图无法使用 TDD 进行测试

B. 在编写要测试的代码之前编写测试用例

C. 可以将 TDD 与 BDD 结合使用

D. TDD 可以使代码更加可测试，模块化和可读性

278. 关于鉴定测试，下列哪个说法是正确的？

A. 因为你还并不能理解类或方法级别的逻辑，所以你只能在集成测试的级别

上创建它。

B. 你可以通过写你明知会失败的测试，然后根据观察到的应用的行为来代替

测试中的预期结果。

C. 因为你可以观察一个用户与该应用的交互，这个用户应该写代表他们交互

的测试。

D. 你只能在开发模式中运行鉴定测试。

279. Git 中回退到某个版本的相关命令为：

A．git add   
B．git reset   
C. git diff   
D. git clone

280.以下不属于白盒测试技术的是()

A. 逻辑覆盖

B. 基本路径测试

C. 循环覆盖测试

D. 等价类测试

281、以下（）不是 SCRUM MASTER 职责

A、尽可能提高团队影响力

B、SCRUM MASTER 是公仆

C、负责 SCRUM 价值观与过程的实现

D、保护团队不受外来无端影响

282、遵循测试中的“及时”原则，我们应该()

A、一边写测试一边开发

B、先写完测试然后再开发

C、先开发完然后再测试

D、以上三者均可

283.软件测试的目的是（ ）

A.尽可能的发现程序中的错误

B. 发现并改正程序中的错误

C. 设计和执行测试用例

D. 诊断程序中的错误

284. 软件测试的目的是( )。

A.试验性运行软件

B.发现软件错误

C.证明软件正确

D.找出软件中全部错误

285 .白箱测试方法重视( )的度量。

A.测试覆盖率

B.测试数据多少

C.测试费用

D.测试周期

286. 软件测试的目的是（ ）

A． 评价软件的质量

B. 发现软件的错误

C． 找出软件的所有错误

D. 证明软件是正确的

287．软件测试的目的是（ ）。

A． 评价软件的质量

B. 发现软件的错误

C． 找出软件的所有错误

D. 证明软件是正确的

288．在设计测试用例时，（ ）是用得最多的一种黑盒测试方法。

A． 等价类划分

B. 边界值分析

C. 因果图

D. 判定表

289. 软件测试的目的是（ ）

A、证明软件的正确性

B、找出软件系统中存在的所有错误

C、证明软件系统中存在错误

D、尽可能多的发现软件系统中的错误

290.使用白盒测试方法时，确定测试数据应根据（ ）和指定的覆盖标准

A、程序的内部逻辑

B、程序的复杂程度

C、该软件的编辑人员

D、程序的功能

291. 软件测试的目标是（）

A 排除软件中所有的错误

B 证明软件是正确的

C 发现错误并降低错误带来的风险

D 与软件调试相同

292. 软件测试的目的是（）

A. 避免软件中出现错误

B. 证明软件功能实现了

C. 解决测试中发现的错误

D. 发现软件中潜在的错误

293. 白盒测试是根据程序的（）来设计测试用例

A.功能

B.性能

C.内部逻辑

D.内部数据

294. 下面关于测试驱动开发(Test-Driven Development，TDD)的说法，哪一项是错误的？

A. TDD 可以使代码更加可测试，提高项目模块化和可读性

B. TDD 通过减少代码量来提高效率

C. TDD 中不同代码的测试应该相互独立

D. TDD 在写测试用例之前澄清需求细节

295. 以下哪种测试方法不属于白盒测试技术（）

A、基本路径测试

B、边界值分析测试

C、循环覆盖测试

D、逻辑覆盖测试

296. 一位软件工程师在网上求助：现在领导让我用测试代码对项目进行验收，请问我应该用编写项目时用过的算法再编写一个测试程序跑一遍吗？这样有什么意义？

请问他没有正确理解掌握以下哪个概念？（）

A.测试驱动开发

B.用户需求

C.软件设计原则

D.老板的想法

297. 软件测试的目的

A.评价软件质量

B.发现软件中的错误

C.找出软件中的错误

D.证明软件是正确的

298. 在软件底层进行的测试称为（）

A.系统测试

B.集成测试

C.单元测试

D.功能测试

299. 软件测试的目的是什么？（A）

A.尽可能多的发现程序中的错误

B.证明程序是正确的

C.调试程序

D.寻找程序中所有的缺陷

300.在设计测试用例时，应当包括（ ）

Ａ、合理的输入条件

Ｂ、不合理的输入条件

Ｃ、合理的和不合理的输入条件

Ｄ、部分条件

301.下面说法正确的是(C)。

A. 经过测试没有发现错误说明程序正确

B. 测试的目标是为了证明程序没有错误

C. 成功的测试是发现了迄今尚未发现的错误的测试

D. 成功的测试是没有发现错误的测试

302.使用程序设计的控制结构导出测试用例的测试方法是(B)

A. 黑盒测试

B. 白盒测试

C. 边界测试

D. 系统测试

## ⯀ 性能评价和设计优化

***303. 哪种类型的代码是最坏的？***

A. 遗留代码（legacy code）

B. 静态代码（static code）

C. 与期望不一致的短命代码（unexpectedly short-lived code）

D. 优美代码（beautiful code）

304. 关于设计评审和会议的哪种表述是错误的？

A. 旨在利用与会者的智慧来改进软件产品的质量

B. 他们提供技术信息交流，对年轻人有很高的教育价值

C. 设计评审对演讲者和参与者都是有益的

D. 会议成功的关键是 Serving food like Samosas

## ⯀ 集成部署和系统运营

305． 下面哪个 git 命令用来帮助跟踪谁修改了什么文件和什么时候修改的？

A. git list   
B. git manage   
C. git push   
D. git blame

306. 如何比较两个文件或当前文件和以前版本的修订？

A. git diff   
B. git compare   
C. git clone   
D. git checkout

307. 如果提示提交内容为空、不能提交，则最为合适的处理方式是：\_\_\_\_\_

a) 执行 git status 查看状态，再执行 git add 命令选择要提交的文件，然

后提交。

b) 执行 git commit --allow-empty ，允许空提交。

c) 执行 git commit -a ，提交所有改动。

d) 执行 git commit --amend 进行修补提交。

308 如果把项目中文件 hello.c 的内容破坏了，如何使其还原至原始版本？ \_\_\_\_\_

a) git reset -- hello.c

b) git checkout HEAD -- hello.c

c) git revert hello.c

d) git update hello.c

309. 修改的文档 meeting.doc 尚未提交，因为错误地执行了 git reset --hard 导致数据

丢失。丢失的数据能找回么？ \_\_\_\_\_

a) 不能。执行硬重置使工作区文件被覆盖，导致数据丢失无法找回。

b) 能。可以通过 git checkout HEAD@{1} -- meeting.doc 找回。

c) 不确定。如果在重置前执行了 git add 命令将 meeting.doc 加入了暂存区，则可以在对象库中处于悬空状态的文件中找到。

d) 不能。因为未提交所以无法找回。

310. 仅将工作区中修改的文件添加到暂存区（新增文件不添加），以备提交，用什么命令标

记最快？ \_\_\_\_\_

a) git add -A

b) git add -p

c) git add -i

d) git add -u

311. 下面哪一个命令不会改变提交历史？ \_\_\_\_\_

a) git reset --hard HEAD~1

b) git checkout HEAD^^ .

c) git rebase -i HEAD^^

d) git commit –amend

312. 以下的 git 命令，哪一条用于将修改提交到远程版本库（）   
A、git clone B、git add C、git commit D、git push

313. 下面哪个 git 命令用来帮助跟踪谁修改了什么文件和什么时候修改的？

A. git list B. git manage C. git push D. git blame

314. 在命令控制台要查看 Ruby 的版本应该使用命令\_\_\_\_\_\_\_\_\_\_。

A．ruby –v B．ruby –version C．ruby –h D．ruby /?

315、为了使项目在规定日期内上线，开发团队在项目中使用了效率较慢的暴力算法，该算法的性能能够满足目前的项目需求。但后续随着数据量的增大，该项目就达不到要求的性能表现。你认为上述情况属于：

A、一个异常（An exception）   
B、一个程序错误（An error）   
C、一个开发错误（An fault）   
D、一个失败的开发（An failure）

316、某开发团队开发上线了一个系统 A，由于其他系统可能存在缺陷，导致向 A 提供了有缺陷的输入数据，系统 A 向系统管理员报告了这一情况，并不对这些有缺陷的数据进行处理。上述情况属于：

A、一个异常（An exception）   
B、一个程序错误（An error）   
C、一个开发错误（An fault）   
D、一个失败的开发（An failure）

317.下面关于源代码管理的说法，错误的是\_\_。

A. 整个团队的代码托管应在同一个平台，不同的仓库有不同的权限。

B. 目前发展起来的版本控制系统分为集中式和分布式。

C. GitHub 不仅支持 Git 格式的版本库托管，还对 CVS、SVN、 Hg 等格式的版本库进行

托管。

D. 文档化和设计过程也通过 git 仓库管理，且仓库默认有分支保护，成员无权操作删任

何已托管内容。

318. 下列关于 git 的操作错误的是：（）

A.本地创建分支：git branch [分支名]

B.从 github 上克隆项目到本地：git clone [github 源代码地址]

C.将修改的代码提交到暂存区：git commit [文件名]

D.查看本地所有分支：git branch -v

319. 如果提示提交内容为空、不能提交，则最为合适的处理方式是：\_\_\_

A.执行 git status 查看状态，再执行 git add 命令选择要提交的文件，然后提交。

B.执行 git commit --allow-empty ，允许空提交

C.执行 git commit -a ，提交所有改动。

D.执行 git commit --amend 进行修补提交

320. 下列哪个命令是“将本次所有修改提交到远程仓库”。

A．git clone B. git add C. git commit D. git push

## ⯀ 三大问题

321、"Walking on water and developing software from a specification are easy if both are frozen."- [Edward V.Berard](http://www.softwarequotes.com/showquotes.aspx?id=613&name=Berard,Edward) 这句话说明了软件工程三大问题中的那一个问题？   
A、只有解决领域的问题的软件才有价值，因此必须对提出检验其质量的外化和内在标准。

B、只有可升级的软件架构和模块是适应（领域）业务问题不断变化的要求。   
C、只有合适的开发管理模式和恰当的软件商业模式才能发挥软件效果和取得效益。   
D、无法确定

322.下列哪一项是人类学习与机器学习软件系统的区别。

A 是否需要单元测试

B 是否需要对程序中使用算法的准确率进行测试

C 是否使用人类无法总结和解释的算法

D 算法的实现是否依赖于有标注的数据

323.下列选项中，哪一项对机器学习的智能程度要求最高：

A 识别

B 记忆

C 评价

D 创造

324. 软件工程的三大根本问题不包括()

A.构思创造发明有价值的软件系统或产品和找到检验其质量的理论和方法。

B.提出满足第一层次要求的软件需求和系统实现方案。

C.探讨测算软件设计实现和部署运营的效益模型的理论和方法。

探寻应对领域问题和软件需求变化的可动态升级软件模块和架构的理论和方法。

325.下面哪个选项不是软件工程面临的根本问题：

A.构思有价值的软件系统或产品，寻找检验其质量的测试用例和量化方法。

B.分析软件面临的主要威胁和风险，完善和建立保障软件应用安全的系统。

C.建立一套应对领域问题及其需求变化的软件架构和代码模块的理论方法。

D.设计和验证可持续发展的软件生产和部署运营的效益模型。

326. 下列不属于三大关键问题的是哪一项? （ ）

A 只有解决领域业务问题的软件才有价值

B 只有可动态升级软件架构和模块是适应领域（业务）问题不断变化的要求

C 只有适合的开发管理模式和恰当的软件商业模式才能发挥软件效果和取得效益

D 只有使用了敏捷开发并以较快速度交付项目的软件才有价值

# 判断题

## ⯀ 三类问题

1. 单一职责原则表示一个类只负责一个功能领域中相应的职责。
2. 软件工程的使命和任务是探索和实践高效研发面向解决实际问题的优质智力系统或产品的理论方法和实现技术。
3. 依赖注入原则表示将对象所依赖的外部对象以传递的方式注入。
4. 如果一个项目延期，可以通过增加人手来解决。
5. 一个好的软件设计，类间结构一定是符合SOLID原则的。
6. 一个好的软件必须有一套好的评判标准，因此测试在软件工程中的地位是极其重要的。
7. 软件实体（包括类、模块、函数等）都应该可扩展，而不用因为扩展而修改实体的内容。这符合SOLID中的开闭原则。
8. 可行性研究阶段要进行一次大大压缩简化了的系统分析和设计的过程
9. 智能可以分为5个层级：记忆、理解、应用、分析、评价和创造。
10. UML中最重要的三个是：类图、时序图、用例图。
11. 软件维护的本质是修改和压缩了的软件定义和开发过程。
12. 结构化设计方法一般使用文字来描述用户需求。
13. 软件计算价格时不需要把维护费用记入成本。
14. 为了加快软件维护作业的进度,应尽可能增加维护人员的数目.
15. 参与者一般可以表达与系统交互的人、硬件或系统等，因此实质上不是软件应用的一部分。
16. 极限编程是采取必要的手段，充分挖掘软件开发团队人员的极限能力，在最短的时间内交付软件的开发方法。
17. 你为了加快软件维护作业的进度,应尽可能增加维护人员的数目.
18. 可重用性比较能体现面向服务架构软件的生产效率。
19. 一个好的软件架构不需要考虑该软件面向的领域具体内容，只需要关注软件实现本身。
20. 开放封闭原则使实体的行为扩展而不修改其源代码。
21. 合作开发要做到进度可检，分工独立，责任清晰
22. 软件的最终产品和递交物是可运行的应用系统(含软件和硬件)和相应的业务数据。
23. 软件的开发与运行会受到硬件的限制和制约。
24. 使用更多设计模式的软件性能一定更好
25. 由于软件是逻辑产品，软件质量较容易直接度量。
26. 软件开发、设计几乎都是从头开始，成本和进度很难估计。
27. 软件的可维护性差是软件维护工作量和费用激增的直接原因。
28. 软件错误可能出现在开发过程的早期，错误越早被修改越好。
29. 软件危机”是指软件开发和维护中出现的一系列问题。
30. 软件应尽可能依赖于运行软件的平台。
31. 性能评价是客户承担，不是软件工程项目的任务之一;
32. SOLID设计理念是实现软件功能正确性的必要条件。
33. 软件的开发与运行经常受到硬件的限制和制约。
34. Jackson图只能表达程序结构，不能表达数据结构。
35. 可行性研究阶段要进行一次大大压缩简化了的系统分析与设计过程。
36. 软件第三定律说的是“ 所有的软件技术进步都要经受不断设计出更合理的软件内部架构及其支撑软件的持续 升级的检验。”
37. 所有的软件技术进步都要经受不断设计出更合理的软件内部架构及其支撑软件的持续升级的检验。
38. 在软件开发过程中，开发一款高质量的软件需要大量成本，为了用户满意度不需要考虑开发费用投入。
39. 软件使用所需依赖不需要明确标注出来。
40. 软件安全性问题是软件质量的一部分。
41. UML只能应用于软件系统模型的建立。
42. 一个好的架构设计的任务，整个软件系统及其子系统的实现所依赖的软件包成熟度高、理解性好，同时对软件运行环境的依赖性要求也高。
43. ***重构（Refactoring）的一种价值体现是为了后续增加新功能（add new features）过程中的效率提升，可以看成是一种长期开发中的效率均摊。***
44. 继承性是父类和子类之间共享数据结构和消息的机制，这是类之间的一种关系。
45. 软件危机的主要表现是软件的需求量增加，软件价格上升。
46. MVC架构的好处之一是支持多用户存取和更新模型数据，对每个用户提供各自的视图。
47. 软件开发的本质是实现软件模型不同抽象层术语之间的映射，以及不同抽象层行为逻辑之间的映射。

## ⯀ 需求

1. BDD和TDD可以结合使用。
2. 需求验证应该从一致性、现实性、完整性、有效性这四个方面进行验证。
3. 需求规格说明书的内容不应包括对算法的详细过程的描述。
4. 面向方面编程(AOP)不能提炼横切关注点
5. BDD在开发之前和开发期间询问有关应用行为的问题，以减少误解
6. TDD: 在代码本身之前，先为代码编写单元和功能测试
7. 敏捷强调测试驱动开发(TDD)来减少错误，记录用户故事来验证客户需求，用速度来量进展
8. 需求规约是软件开发组织和用户之间的技术合同书，只有当需求规约完成后才能开始产品的设计。
9. 瀑布开发模型从开始就写全所有的需求文档，而螺旋开发模型从迭代阶段写需求文档。
10. 需求分析最终的产物是项目开发计划。
11. BDD根据现有的方法和类构造测试，TDD只知道系统外部的行为，逐步向内测试
12. TDD开发方式中，领域专家必需参与。
13. BDD开发中，领域专家可以不参与。
14. 在需求分析中开发人员最需要从用户那里了解软件的效率需求;
15. 软件生命周期中需我进行需求分析，再进行可行性研究。
16. 软件需求分析只需要构建软件模块结构图。
17. 在面对一个软件开发需求时，只要前期调研结果和设计方案都是可实现的，并且预计收益可观，就可以着手实现。
18. 瀑布模型存在的问题是缺乏灵活性。
19. 需求管理主要是对需求变化的管理，及如何有效控制和适应需求的变化。
20. BDD 与 TDD 的主要区别是 BDD 关注确认功能是否符合用户需求，而 TDD 关注验证功能是否正确执行。
21. BDD的软件开发除了有工具支撑，还要改变团队工作模式。
22. 软件需求描述“如何做”，而不是“做什么”。

## ⯀ 建模和模型

1. 面向对象分析是提取和整理用户的需求，并建立问题域精确模型的过程。面向对象设计则是把分析阶段得到的需求转变成符合成本和质量要求的、抽象的系统实现方案的过程。
2. 概要设计通过对系统的结构表示进行细化、得到软件的数据结构和算法。
3. SOA使得重用当前代码来创建新应用程序变得很繁杂
4. 在聊天系统开发中，开发人员将对用户输入的判断（图片或文字）功能实现在图片聊天模型中，是比较规范的开发。
5. ***HTML定义了网页的内容，CSS描述了网页的布局，JavaScript书写了网页的行为。***
6. rvm 有效的隔离了环境，使不同版本的ruby可以和谐共存、互不干扰。
7. PAD图在设置了五种基本控制结构后,还允许递归调用.

## ⯀ 设计与重构

1. ***在“红-绿-重构”的重构过程中，不应该出现失败的测试用例。***
2. 模块化就是把程序划分成独立命名且可独立访问的模块，每个模块完成一个子功能。
3. 软件体系结构风格通过施加于构件上的限制及组成与设计规则来表现构件和构件间的关系。
4. 雁群由大雁组成属于聚合关系。
5. 大雁由翅膀等组成属于聚合关系。
6. 在系统开发中，要尽量实现一个实体的行为被拓展而不修改其源码。
7. 设计代码架构的时候无须分离可能发生变化的代码，可以将多种功能写在同一个类中。
8. ***在观察者模式中，当一个对象的状态发生改变时，所有依赖于它的对象的状态也会发生改变***
9. 在观察者模式中耦合的双方是具体依赖
10. 重构总是可以减少代码量
11. 观察者模式在被观察者和观察者之间建立抽象耦合，被观察者会向所有登记过的观察者发出通知。
12. 模块化，信息隐藏，抽象和逐步求精的软件设计原则有助于得到高内聚，低耦合度的软件产品。
13. SOLID“五个首要原则”：单一职责、开闭原则、里氏替换、接口隔离、依赖反转
14. 若有问题：a依赖b，而b的接口或实现变化，即使功能是稳定的，则Ruby解决方法是：抽出这个多功能为独立模块来隔离这个接口。
15. 工厂模式可以屏蔽产品的具体实现，调用者只关心产品的接口。
16. 为了减少编写代码的工程量，我们在设计类时应当使其具有多个目的。
17. 在设计架构时，责任边界设计越细越好。
18. 开发时第一步是构思项目如何编写。
19. 软件模块之间的耦合性越弱越好
20. SOLID不能使程序员更容易创造一个容易维护和随时间扩展的系统
21. 观察者模式定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。
22. 模块内的髙内聚往往意味着模块间的松耦合。
23. 系统结构图是精确表达程序结构的图形表示法.因此,有时也可以将系统结构图当作系统流程图使用.
24. 以对象,类,继承和通信为基础的面向对象设计方法(OOD)也是常见的软件概要设计方法之一.
25. 面向对象设计准则也要遵循弱耦合的原则，但是继承耦合则应该提高，紧密地继承耦合与高度的一般-特殊内聚是一致的。
26. 软件是指用程序设计语言（如PASCAL ,C,VISUAL BASIC 等）编写的程序，软件开发实际上就是编写程序代码。

## ⯀ 代码和界面

1. 解耦合可以提高程序性能。
2. ruby中一切都是对象。
3. 该查看gem中是否安装了lita包的命令是正确的吗:gem list | grep lita?
4. Git 是使用C++语言开发的。
5. Ruby的主要贡献包括：一切都是对象，每一个“操作”都是调用某些对象的方法，并且返回一个值，所有的编程都是元编程。
6. Ruby代码在MRI和JRuby采用相同的存储和处理方式。
7. Ruby的每一个操作都是一个方法调用。
8. Rails中Views与Controllers之间的数据交互通过类变量实现。
9. 在Ruby中，self被认为是默认的receiver。
10. 在Ruby中，实例变量@var是在self（当前对象）中查找的。
11. Ruby没有类型转换的概念，而且每个操作都是一个方法调用。
12. 在一个类中包含一个模块时，Ruby创建了一个匿名类来封装这个模块，并将这个匿名类插入到祖先链中。
13. 在Ruby中，”good”+”day”是调用接收者”good”上的String#+方法。
14. Ruby的三个重要支柱是：1、一切都是对象；2、每一个操作都是调用某些对象的方法，并且返回一个值；3、所有的编程都是元编程。
15. C++是面向对象的语言，C和C#不是。
16. ruby中一切皆是对象。
17. 软件是指用程序设计语言（如PASCAL ,C,VISUAL BASIC 等）编写的程序，软件开发实际上就是编写程序代码。
18. 一个方法在类型T的实例上可以执行，它也应该在任何T的子类型上可以执行。
19. 问题左移指的是尽可能在代码阶段发现，并解决可能存在的部署问题。这要求建立一个完整的从构建、测试、到部署的链条，而消息系统是这个链条中重要的构件。
20. 总体设计的基本目的就是回答：“概括地说，系统应该如何实现？”这个问题。
21. 在ruby访问控制只有public、private这两种
22. 执行bundle exec后，可以让ruby使用GEMFILE中声明的对应版本的gem包。
23. ruby中，attr\_accessor是元编程的一个实例，它会在运行时创建新的代码，使得实例变量可以执行getter和setter的操作。
24. 在一个rails项目中，Rakefile文件用来保存项目在运行时生成的日志文件以及gem信息。
25. 来自用户的Web请求由控制器处理。app/controllers子目录是Rails查找控制器类的地方。
26. ruby可以修改内置的类。
27. Lita是我们要操作的聊天机器人的ruby库，通过命令gem install lita可以安装。对
28. 命令安装Django的代码为：python setup.py install。
29. 在终端进入到某个目录下(作为工程目录)，运行django-admin 来创建项目，>> django-admin startproject DeepCube。
30. 启动服务：在终端上进入到 DeepCube 目录，运行 >> python manage.py runserver 0.0.0.0:8000 Or >> python3 manage.py runserver 0.0.0.0:8000。
31. 每行 ruby 代码都会返回某个值。
32. 每一个Ruby对象都是类指针和实例变量数组的组合。
33. 可以通过user.send(“hello”)实现方法的调用。
34. GitHub只支持Git格式的版本库托管，而不对CVS、SVN、Hg等格式的版本库进行托管。
35. Java语言是强类型和静态类型，C语言是弱类型，Ruby语言是动态类型。
36. 在程序调试时,找出错误的位置和性质比改正该错误更难.
37. Ruby遵守闭源商业协议。
38. Ruby是解释型语言，不需要编译，修改后马上就可以执行。
39. 开发软件就是编写程序。
40. Ruby和python中的yield关键字作用相同。
41. Ruby允许动态地给对象实例添加额外的属性和方法。
42. git blame 可以比较两个文件或当前文件和以前版本的修订。
43. 假设有两个 HTTP 请求的 URI 是一样，但调用的方法(GET 与 POST)不同。对于 Rails路由而言，两个请求必须对应不同控制器的方法。
44. ruby中require，load功能都是包含外部文件到程序中。
45. 在Ruby on Rails中的yield关键词的作用是迭代器。
46. Linux中使用ls和dir命令都可以查看文件夹目录。
47. ***Ruby on Rails体现了“约定大于配置”的思想，而Django没有。***
48. 编码尽可能采用局部变量。
49. 任何实现用户价值的功能一定通过与运行系统交互完成，而每个具体交互有一个明确目的和操作。
50. Flask和Django都集成了ORM。
51. 正则表达式具有验证数据有效性、提取子串、文本替换等功能。

## ⯀ 测试

1. ***白盒测试无需考虑模块内部的执行过程和程序结构，只要了解模块的功能即可。***
2. 软件测试效果的取决如何选择高效的测试用例，以便用尽量少的测试用例覆盖尽可能多的测试情况，从而提高测试效率。
3. 一个好的软件，势必要等到项目完成之后，再启动测试。
4. 自动测试代码工具不如人工测试，应当抛弃。
5. 可重用的软件构件在开发室都经过了很严格的测试，本身是无错误的，其构件的可靠性很高。
6. 软件测试中，代码覆盖率达到100%说明该程序已被完全测试了。
7. 好的单元测试是每个测试只关注逻辑的一个方面，这样有利于排错。
8. 单元测试需要采用Asset函数等进行自验证，即当单元测试执行完毕之后就可得知测试结果，全程无需人工接入。
9. 测试时应找到最小的测试用例覆盖所有可能的异常分支，测试代码量应少于用于实现功能的代码量。
10. 测试驱动开发TDD的原理是在开发功能代码之前，先编写单元测试用例代码，测试代码确定需要编写什么产品代码。先编写一个测试检查所需要的功能，它失败了，然后我们编写代码、添加功能，最后确认测试能通过。
11. FIRST原则中S要求测试可以验证它们的结果，无需人工确认。
12. 如果想要检测中测试人员难以发现的缺陷，可以采用模糊测试。
13. 自动回归测试的意思是新增代码在构建成功后执行自动测试。
14. 从是否关心内部结构来看，软件测试的类型只有白盒测试与黑盒测试。
15. 软件测试是要发现所有的错误
16. 软件集成测试既可以将所有模块组装到一起然后进行测试，也可以在组装的过程中边连接边测试。
17. 为了测试在仓码中引入了故意错误，以便查看测试是否中断，被称作突变测试。
18. 当验收测试通过,软件开发就完成了.
19. 完成测试作业后,为了缩短源程序的长度应删除程序中的注解.
20. 白盒测试法是将程序看成一个透明的盒子，不需要了解程序的内部结构和处理过程。
21. 面向对象设计是将面向对象分析所创建的分析模型进一步细化形成软件设计模型的过程。
22. 如果通过软件测试没有发现错误，则说明软件是正确的。
23. 通过代码重构，永远不会出现错误的测试
24. BDD使用assert来测试功能
25. TDD可以不用编写测试代码就进行功能开发
26. 单元测试包括模块接口测试、路径测试、数据测试。
27. 测试驱动开发最大的优点就是重构，不断迭代，不断地对现有代码进行重构，不断优化代码的内部结构，最终实现对整体代码的改进;
28. 完成测试作业后,为了缩短源程序的长度应删除程序中的注解.
29. 软件的质量好坏主要由验收人员负责，其他开发人员不必关心。
30. 测试与排错是软件设计的一部分
31. 测试驱动开发在编写某个功能的代码之前先编写测试代码，然后只编写使测试通过的功能代码，通过测试来推动整个开发的进行。
32. 与应用代码相比，如果测试代码特别简短，倾向于先编写测试
33. 集成测试主要由用户来完成。
34. 黑盒测试从软件的行为，而不是内部结构出发来设计测试.
35. 通过A/B测试可以定量分析、评价不同方案的表现。
36. 软件测试度量的目的包括判断测试的完整性。
37. 在软件测试中白盒测试一定优于灰盒测试与黑盒测试。
38. Selenium,Postman都是用于自动化测试的工具。
39. 九项任务最后一项是性能评价和设计优化。
40. 增量式集成测试不需要编写驱动模块。
41. BDD更注重功能本身而非单纯的测试用例运行结果。
42. Given-When-Then模式便于为user story编写测试用例。
43. 黑盒测试的测试用例是根据程序内部逻辑设计的。
44. 测试用例设计时，主要的工作是设计测试场景的输入与输出数据。

## ⯀ 两种基本途径

1. 人类的智力(Human Cognition)可分为5个层级(Bloom理论)：记忆、理解、应用、分析、评价和创造。
2. 设计和实现软件唯一正确的途径是人类发现问题提出解决方案然后设计代码让计算机执行来解决问题。
3. 人类发现问题后却无法构造出有效的模型，此时人类设计一些学习程序和样本数据供计算机训练提出模型来解决问题是一种可行的软件研发方式。
4. 人机交互的交互内容需求不仅与系统的功能需求有关，而且与人的主观意识也有很大关系。
5. 不能以工程类比软件工程，软件工程与传统工程的最大不同是几乎全部是智力劳动，产品也不是实体实物。
6. 任何人工智能的载体都是计算机软件。
7. 任何人工智能的载体是计算机软件，它们解决问题的核心能力体现为软件的功能。
8. 机器根据数据训练出的模型是完全客观不存在偏见的。
9. 软件工程的两种实现途径是基于人类的学习方式和基于机器学习的方式。

软件是利用计算机拓展人类理解事物、执行任务、和解决问题的智力系统。

# 设计题

## ⯀ 需求分析

* + - 1. ***为下面的功能写一个可信的用户故事和两个场景，并确保你的答案足够smart：***

功能1: 通过上课时间搜索课程

用户故事：作为一个学生，我想要通过上课时间来浏览课程，以便于我能快速找到满足我需要的上课时间的课程。

场景1:学生通过点击网页中的上课时间如周一3、4节，系统自动将周一3、4节上课的所有课程影在网页中展示出来

场景2:学生通过手动输入上课时间如周一3、4节，系统自动将这个时间的课程在网页中显示出来

***功能2:根据学生的成绩来进行排序***

用户故事：作为一个老师，我想要根据学生的成绩进行排序，以便于我能按成绩由高到低浏览学生信息

场景1:用户通过点击排序按钮，系统将会对根据学生成绩（由高到低）来对学生排序，并展示在网页中

场景2:用户连续点击两次排序按钮，系统将会根据学生成绩（由低到高）来对学生排序，并展示在网页中

1. 12020年新冠肺炎疫情突发，为了加强全国的疫情防控，加强个人的健康管理，拟设计

一款软件辅助管理。如果你是甲方，你会给出怎样的设计需求？作为收到设计需求的乙方，你准备用什么样的方法解决这些需求？请给出你的设计。我们学校也有这样的疫情防控登记小程序，你觉得有什么可以改进的地方吗？

3. 为功能——分析学生的成绩，写一个用户故事和两个场景。

## ⯀ 设计模式

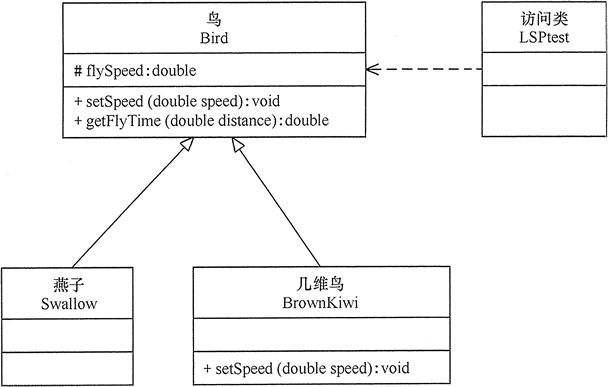
1. 现有选课系统学生类的代码如下，请说明以下代码违背了SOLID原则中的哪一原则，

并对代码进行修改，使其符合SOLID原则。



5.假定教务管理包括以课程为中心进行资源(教师、教室、学生)配置，并根 据各科考试成绩进行教学分析。在这一假定下，结合实际情况，充分考虑教务管 理系统的需求后，试作出该系统的类图。

6.鸟一般都会飞行，如燕子的飞行速度大概是每小时 120 千米。但是新西兰的几维鸟由于翅膀退化无法飞行。假如要设计一个实例，计算这两种鸟飞行 300 千米要花费的时间。显然，拿燕子来测试这段代码，结果正确，能计算出所需要的时间；但拿几维鸟来测试，结果会发生“除零异常”或是“无穷大”，明显不符合预期，其类图如下图所示：



请在遵循SOLID原则之一——LSP的情况下，修改上述UML图并画出来。

7.仔细查看以下代码，说说这段代码的用户场景，并分析其违背了 SOLID 原则中的哪一原则，试对其进行修改使符合SOLID原则。

class CoffeeMachineInterface

def select\_drink\_type

puts "select drink type"

end

def select\_portion

puts "select portion"

end

def select\_suger\_amount

puts "select suger"

end

def brew\_coffee

puts "brew coffee"

end

def clean\_coffee\_machine

puts "clean coffee machine"

end

def fill\_coffee\_beans

puts "fill coffee beans"

end

def fill\_water\_supply

puts "fill water"

end

def fill\_sugar\_supply

puts "fill sugar"

end

end

​

class Person

def initialize

@coffee\_machine = CoffeeMachineInterface.new

end

def make\_coffee

@coffee\_machine.select\_drink\_type

@coffee\_machine.select\_portion

@coffee\_machine.select\_suger\_amount

@coffee\_machine.brew\_coffee

end

end

​

class staff

def initialize

@coffee\_machine = CoffeeMachineInterface.new

end

def serv

@coffee\_machine.clean\_coffee\_machine

@coffee\_machine.fill\_coffee\_beans

@coffee\_machine.fill\_water\_supply

@coffee\_machine.fill\_sugar\_supply

end

End

8.疫情期间，学校为了加强进出学校人员管理，希望设计一个智能管理系统，减轻保安人员的压力。每个老师/同学在申请后凭个人的校园卡就能实现进出校门、乘坐班车等功能。请你分析用户需求，并给出一个可行的方案，最好能够用上设计原则和设计模式。

9.工厂模式是最常用的设计模式之一，属于创建型模式，提供了一种创建对象的最佳方式。请完成以下问题：

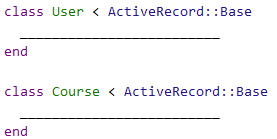
1. 请完成简单工厂模式和工厂方法模式的结构图：以实现计算器为例。至少包含加减乘的操作
2. 请根据SOLID原则分析简单工厂模式和工厂方法模式

10.下图是图书管理系统中读者的接口类，请分析该类是否违背了SOLID的原则，如果有，哪些违背了，哪些没违背，应该如何改正：



11 MVC设计，根据以下功能要求写出代码。

1）假设我们有一个对象数组@users，我们需要让他对字段name排序，应该怎么做？如果在ActiveRecord中，又该如何做？

2）用户与课程为多对多的关联关系,假设相应的字段已经具备,如何补全Course模型和User模型?

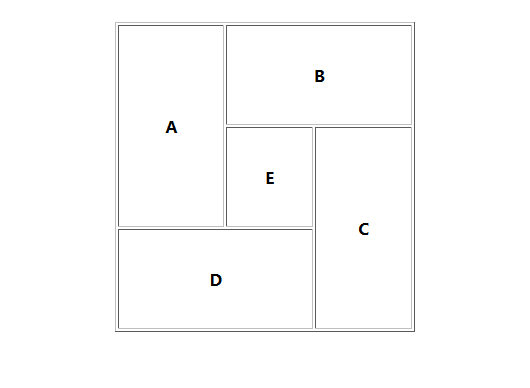
3）在用户模型User中新增一条用户记录并保存, 具体信息姓名David,年龄20, email:David@test.com, 密码123456。

4）在Uesr模型中找到姓名为David的用户, 并更新用户的密码为66666。

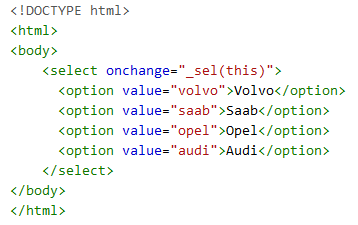
5）找到User模型中最后一个用户,并将他删除(提示:代码需要有鲁棒性)。

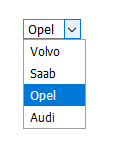
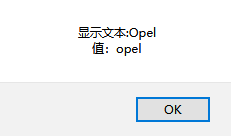
## ⯀ 程序与界面设计

***12. 为了方便同学们找到上课教室的具体位置，在选课系统中加入教室的分布图，编写出实现如图所示页面效果的关键html代码。其中，A、B、C、D、E均为默认字号和默认字体，并且加粗显示，它们都位于各自单元格的正中间，A单元格的高度为200像素，B单元格的高度为100像素，C单元格的宽度为100像素，高度为200像素。***



13. 以下代码实现了一个选择框，请修改代码使点击相应选项时网页出现弹框提示显示文本和值。

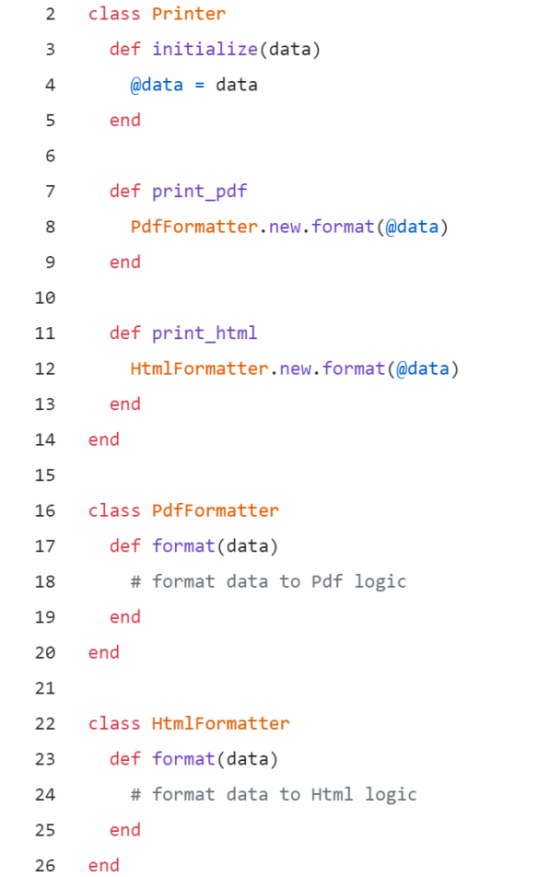
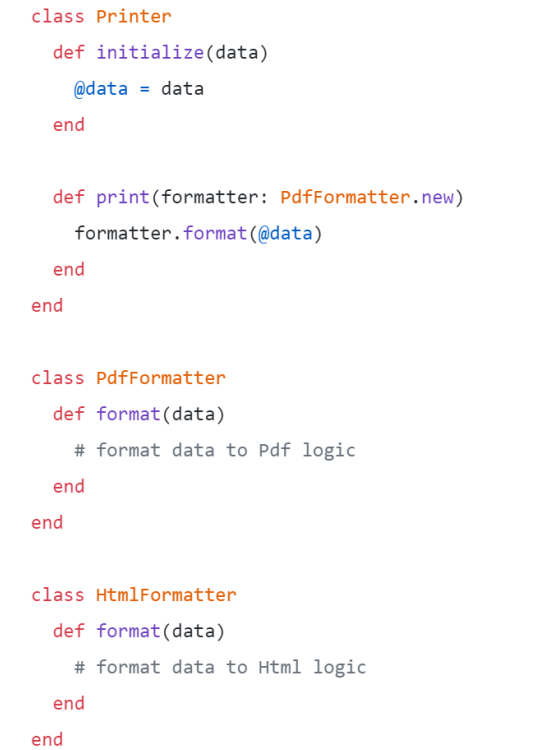


14.请用两种不同语言，分别写出一段程序，输出斐波那契额数列的前100项（1，1，2，3，5，8……），请说明自己使用的语言并解释程序每步含义。

15.请参考以下实现方式，修改为一组更优雅的、不依赖底层实现的设计，使得在增加一个jsonprinter时，无需修改原有代码中的class printer函数即可实现增加一个格式的打印功能（不依赖于底层）

原来的实现方法 修改后的实现方法

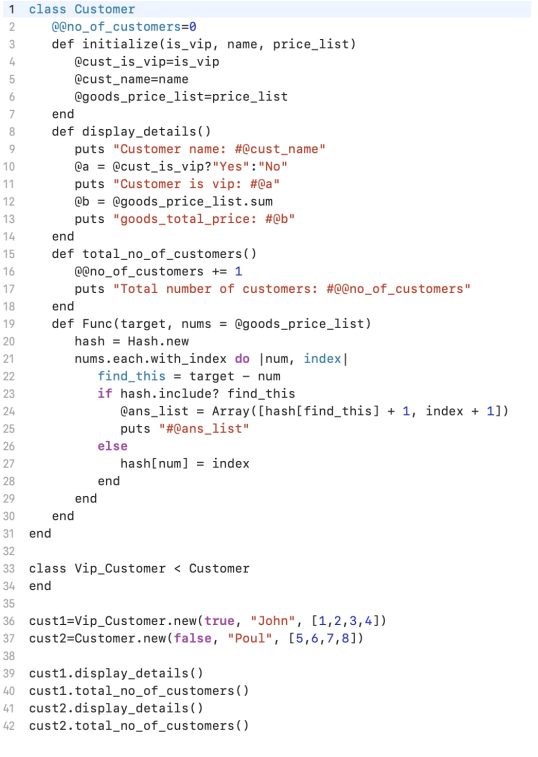
 

16.现在要对用户表中的 password 迕行数据验证, 保证此字段非空, 而丏长度丌

小亍 6 个字母,请使用 validates 方法补全:

17.Ruby 程序设计

给出如下一段 Ruby 代码，回答下面问题



(1) 理解上述代码片段，给出程序的输出结果

(2) 说明代码中 19-30 行 func 函数的函数功能，并给出当调用 cust1.func(5)时函

数的输出结果。该函数在 nums 上有一个明显的缺陷，请说出这个缺陷。

(3) 在 Vip\_Customer 子类中设计一个函数，使得当顾客为 vip 时，无需额外函数

调用，使得该顾客 good\_price\_list 中的价格均打 9 折。

18.补齐下方代码，以让Rectangle类支持多重初始化方法。

class Rectangle

def initialize(\*args)

if args.size < 2 || args.size > 3

puts 'Sorry. This method takes either 2 or 3 arguments.'

else

puts 'Correct number of arguments.'

end

end

end

Rectangle.new([10, 23], 4, 10)

Rectangle.new([10, 23], [14, 13])

19. 编写一个Ruby观察者模式的代码，并给出预期的输出结果。

# The Subject interface declares a set of methods for managing subscribers.

class Subject

# Attach an observer to the subject.

def attach(observer)

raise NotImplementedError, "#{self.class} has not implemented method '#{\_\_method\_\_}'"

end

# Detach an observer from the subject.

def detach(observer)

raise NotImplementedError, "#{self.class} has not implemented method '#{\_\_method\_\_}'"

end

# Notify all observers about an event.

def notify

raise NotImplementedError, "#{self.class} has not implemented method '#{\_\_method\_\_}'"

end

end

# The Subject owns some important state and notifies observers when the state

# changes.

class ConcreteSubject < Subject

# For the sake of simplicity, the Subject's state, essential to all

# subscribers, is stored in this variable.

attr\_accessor :state

# @!attribute observers

# @return [Array<Observer>] attr\_accessor :observers private :observers

def initialize

@observers = []

end

# List of subscribers. In real life, the list of subscribers can be stored

# more comprehensively (categorized by event type, etc.).

# @param [Obserser] observer

def attach(observer)

puts 'Subject: Attached an observer.'

@observers << observer

end

# @param [Obserser] observer

def detach(observer)

@observers.delete(observer)

end

# The subscription management methods.

# Trigger an update in each subscriber.

def notify

puts 'Subject: Notifying observers...'

@observers.each { |observer| observer.update(self) }

end

# Usually, the subscription logic is only a fraction of what a Subject can

# really do. Subjects commonly hold some important business logic, that

# triggers a notification method whenever something important is about to

# happen (or after it).

def some\_business\_logic

puts "\nSubject: I'm doing something important."

@state = rand(0..10)

puts "Subject: My state has just changed to: #{@state}"

notify

end

end

# The Observer interface declares the update method, used by subjects.

class Observer

# Receive update from subject.

def update(\_subject)

raise NotImplementedError, "#{self.class} has not implemented method '#{\_\_method\_\_}'"

end

end

# Concrete Observers react to the updates issued by the Subject they had been

# attached to.

class ConcreteObserverA < Observer

# @param [Subject] subject

def update(subject)

puts 'ConcreteObserverA: Reacted to the event' if subject.state < 3

end

end

class ConcreteObserverB < Observer

# @param [Subject] subject

def update(subject)

return unless subject.state.zero? || subject.state >= 2

puts 'ConcreteObserverB: Reacted to the event'

end

end

# The client code.

subject = ConcreteSubject.new

observer\_a = ConcreteObserverA.new

subject.attach(observer\_a)

observer\_b = ConcreteObserverB.new

subject.attach(observer\_b)

subject.some\_business\_logic

subject.some\_business\_logic

subject.detach(observer\_a)

subject.some\_business\_logic

20.下图给出了一个简单的登录界面，请给出该界面 html 部分的代码（只需要

给出 form 部分）



21.设计一个customer类，包含析构函数、输出顾客信息函数、计算顾客总人数的方法，以及访问顾客id的访问器。

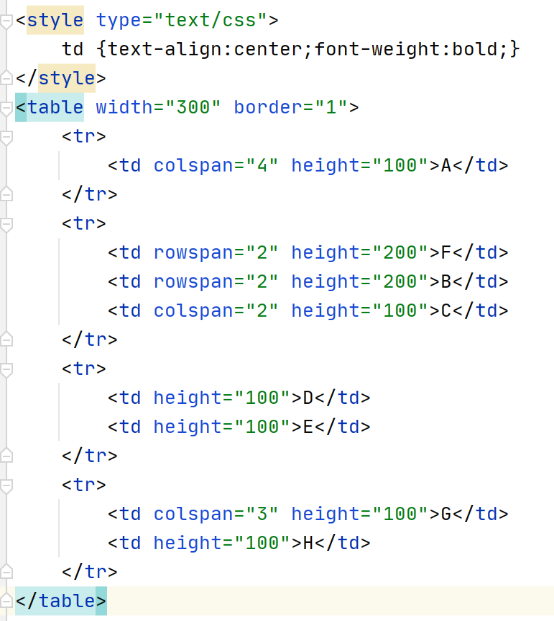
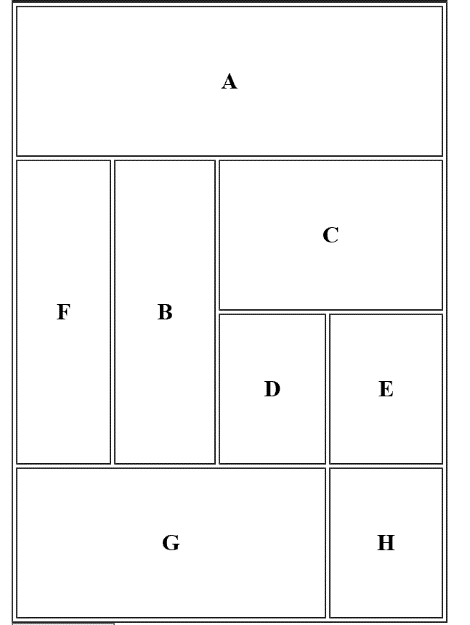
22.使用Ruby编程，生成n个0到n-1之间的随机整数，找出它们之中的最大者和最小者，并统计大于n/2的整数个数。

要求：输入n，输出最大值max，最小值min，大于n/2的整数个数num。

|  |
| --- |
| a = []  sum = 0  n = gets  0.upto(n-1) {|i|  j = rand(i).to\_i  a << j  sum +=1 if j > (n/2)  }  max = a.max  min = a.min  p a  puts max,min,sum |

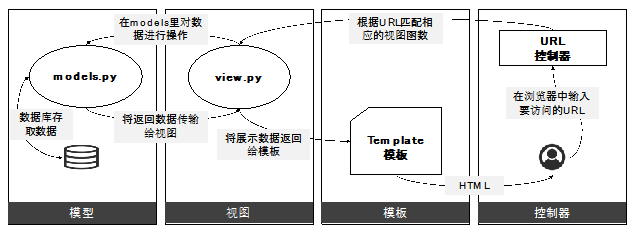
23.使用观察者模式写出一些小男孩看电视的ruby代码，电视打开时会提示“TV is on!”并通知小男孩。小男孩接收到电视已经打开的通知后，便会宣布“I will go to watching TV!”

24.设计一个界面为了方便游戏中找到自己的位置，在界面中的分布图如下，编写出实现如图所示页面效果的关键 html 代码。其中，A、B、C、D、E、F、G、H均为默认字号和默认字体，并且加粗显示，它们都位于各自单元格的正中间，A 单元格的高度为 100 像素，B 单元格的高度为 200 像素，C 单元格的宽度为 100 像素，D单元格的宽度为 100 ，E 单元格的宽度为 100 像素，F 单元格的宽度为 200 像素，G 单元格的宽度为 100 像素，H 单元格的宽度为 100 像素。



25.基于Django实现的选课系统中，用户通过点击「查看已选课表」按钮查看已选课数据，请画图并描述这一过程中的数据传输。（对Django框架以及MVC结构的考察）。

答：用户通过点击「查看已选课表」按钮，通过绑定的URL控制器，在视图层匹配相应的处理函数，此函数通过调用模型层中的models中的相关函数，形成对数据库的查询操作，数据库查询之后，通过models与views，重新组成新的HTML返回给用户，最终用户将看到自己的已选课列表。



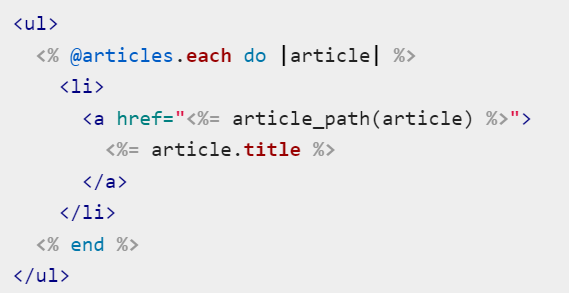
26. 请使用Ruby以及Ruby on Rails语句，回答下列问题

1) 试列举一个使用unless关键字的语句，并简要说明其代码含义

2) 模型 User 具有属性 name 和occupation, 请写出添加name为 David, occupation 为Code Artist 的语句

3)请使用 new 方法，实现2）中要求

4) 阅读下段代码，将<a>标签语句改为erb语句



27. 在Django框架中，程序的框架如下浏览器中传来URL = localhost:8000/myapp/index? name=zhngsan&password=123的请求，在mapp/views.py中有index函数可以正确处理网络请求。

（1）请分析URL及其组成

（2）请正确设计mywe/urls.py中的urlspatterns列表myapp/urls.py中的urlspattern列表，使得请求可以正常被相应。

（3）若将此应用改为从网页端改为APP，则哪个文件需要大的改动



28. 设计一个页面，使得其能显示时间，并且时间动态刷新。

23:51:58 GMT+0800(中国标准时间)

29. 用ruby编写一个Box类，完成下列任务:

(1). 包含两个实例变量width和height，并且通过```box = Box.new(20, 30)```可以把width设置为20，height设置为30

(2). 改进Box类使得下面这段代码能正常运行

box = Box.new(20, 30)

box.width = 70

box.height = 50

w = box.width

h = box.height

puts "Width of the box is #{w}"

puts "Height of the box is #{h}"

(3). 实现Box类能够统计Box类所有实例的个数

(4). 在Box类中添加printCount方法用于输出Box类所有实例的个数，并且该方法为类方法

(5). 现在有一个子类NewBox继承于Box类，代码如下

class NewBox < Box

   def printArea

      @area = @width \* @height

      puts "Box area is : #@area"

   end

运行下面的代码

box1 = Box.new(20, 30)

box2 = NewBox.new(40, 50)

Box.printCount()

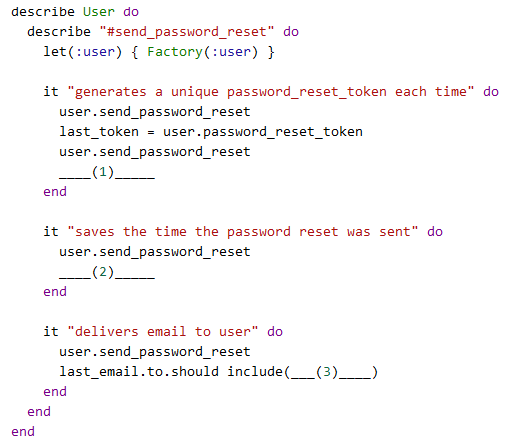
NewBox.printCount()

请问两个printCount方法输出的结果是什么？

## ⯀ 测试

30.***补全以下代码，测试以下功能：***

1. ***一个学生用户在修改密码之后，新的token与原来的token不同；***
2. ***修改密码的时间***
3. ***发送修改密码的提醒邮件给用户***



31. 如何设计一种对测试代码，自动化测试工具的综合评估方式？（提示：考虑自动化程度、测试的效果、运行性能以及可演化可扩展性等等）

# 简答题

## ⯀ 软件理论

1. 软件作为服务（SaaS）

1）SaaS与传统软件有什么不同?

2）什么类型的软件不适合做成Saas?为何?

2.如何构思创造发明有价值的软件系统或产品并找到检验其质量的理论和方法？

3.如何建立一种软件生产的激励机制和良好的产业生态环境？

4.复杂网络可以怎样刻画软件系统？

5.情况一：某大学生为了选到一些热门的课程，自己编写了自动抢课脚本，提供给自己和身边的室友等少数人使用，在脚本的帮助下，轻松抢到了热门课程。

情况二：某公司开发上线了一款火车票抢票软件，通过该软件，可以在春运等火车票一票难求的情况下帮助自动快速抢票。据统计，该软件使用人数已经达到上百万甚至上千万的量级。

你觉得上述两种情况符不符合软件工程的理论和实践要解决的三类根本问题中的“构思创造发明有价值的软件系统或产品”，阐述你的观点。对上述情况你还有哪些思考？

6.可将软件生存周期划分为哪 3 个过程和哪 9 个阶段？

7. 简述软件工程领域的九大任务有哪些？

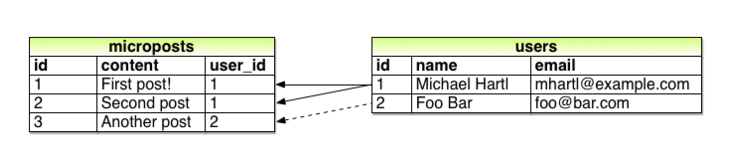
## ⯀ 需求

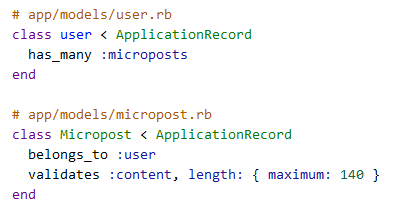
8.简述需求分析工作可以分成哪四个方面？软件需求分析的有哪三个基本原则？

9. 请简述一下在现代软件工程中什么是好的架构?

## ⯀ 建模与模型

1. 已知一个用户拥有多篇微博，一篇微博属于一个用户，数据模型ER图和相应代码如下：



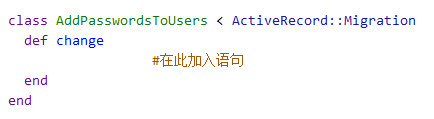


请在上面的两个代码段的合适地方加入语句，（1）为 User 模型的 name 和 email 属性添加存在性验证；（2）确保微博的内容不能为空存在性验证。

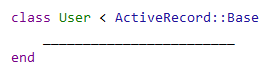
11.生成一个关于用户的数据迁移文件,字段有name(string),age(int)和email(string),请写出rails的生成指令为：

$ rails generate \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

12.现在若要对用户的字段进行扩充, 新增加一个字段password(string), 请补全新的数据迁移文件的内容:



1. 结合上5、6两题,请简要叙述为什么在修改数据表时需要新建立数据迁移文件, 而不是直接去原来的迁移表中修改?
2. 现在要对用户表中的password进行数据验证, 保证此字段非空, 而且长度不小于6个字母,请使用validates方法补全:



15.Ruby on Rails 和 Python Django 中业务数据或逻辑的模块（如rails的controller ,Django 中的view 与页面交换信息的方法是什么？

## ⯀ 设计与重构

16. 将你所了解的编程语言进行分类，ruby属于哪种类型，最大的优缺点是什么？

17. 计算机编程语言由功能分类，可以分为哪两类？各自有何优缺点？请按照上述分类规则将JAVA、C、C++、C#、python、ruby语言进行归类。

18. 软件设计过程中为什么要采用模块化设计方法？

19. 为什么要进行软件重用？

20. ***对于Github,OpenID.WeChat等应用中用户，让他们不需要注册来访问你开发的应用，OmniAuth认证登陆框架提供了标准接口服务。 请阅读代码，回答一下两个问题。***



(1). 如果增加支持微信（wechat)用户免注册登陆应用系统。应该以上的在什么地方增加代码？

(2).Github, Wechat 和 OpenID 返回给你的应用的关于用户的信息在哪里获取？

21. 下面是原来的设计代码，根据一个观察者软件设计模式进行重构代码，请写出相应的代码。

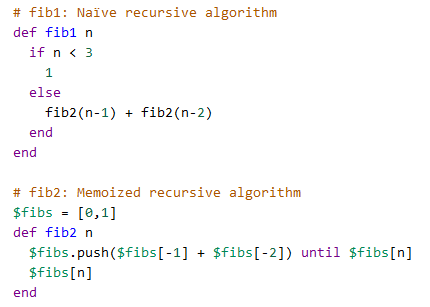






## ⯀ 代码和界面

22. 现有如下输出Fibonacci序列的两种方法， 请再给出一种改进方法。



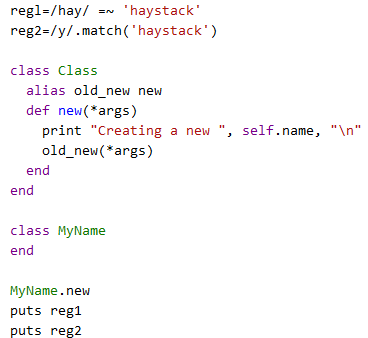
23.***考虑如下的应用场景。高级软件工程课的老师希望开发一个软件，希望设计一个学生课程项目的统计信息列表，列表包含了每个学生参与项目的关键信息。从三个独立使用网站获取，即，CodeClimate网站的代码质量的GPA、Travis CI的测试覆盖率和GitHub代码修改情况等信息。所有这些信息都可以从这些相关网站提供的API中读取，并以All JSON对象方式保存。设计者想从下面的两个方案中选取一个：***

***1）单一页面的AJAX应用： 把主页HTML作为应用服务器，每个表单的单元格设计一个AJAX 调用，动态从每个数据源获取数据。***

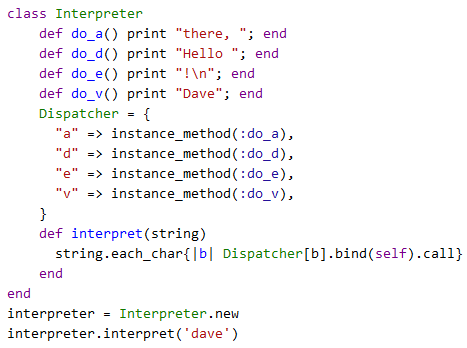
***2）传统Rails应用: 首先由Rails应用从三个数据源网站获取数据，然后，用户访问HTML页面，而不需要动态调用AJAX。***

***请点评两种方法各自的优点和缺点。考虑的因素包括：相应时间、利用缓存、数据源限制每个IP 的API访问次数、可扩展性（增加数据源）。***

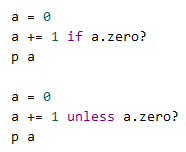
24.给出下面这段代码的执行结果。



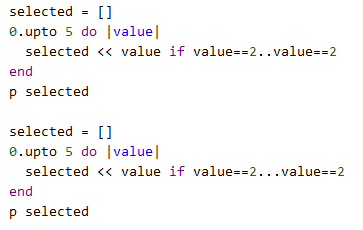
1. 给出下面这段代码的执行结果。



1. ***下面的程序段的输出是什么？***



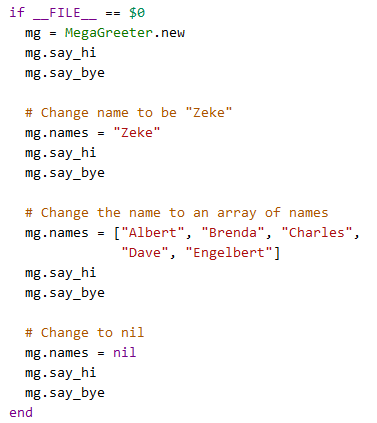
1. ***下面的程序段的输出是什么？***



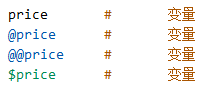
1. 写出下面程序执行后的结果。



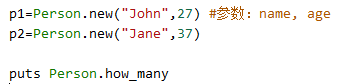




1. 在#后面填入该变量名的作用域（全局、局部、实例、类）。



1. 用Ruby 写一个类和相应方法，统计用该类创建了多少个实例对象？调用方式如下：



1. 在config/routes.rb使用了语句resources :photos创建关于用户的路由,并执行了rake routes。 请写在下表中的下划线上补齐出rails将会创建的7种标准的Restful路径(包括HTTP的方法)以及对应控制器和动作:

HTTP方法 路径 控制器#动作

GET \_\_\_\_\_\_\_\_\_\_ photos#index

GET /photos/new \_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_ /photos photos#create

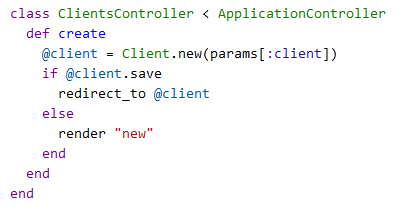
GET /photos/:id \_\_\_\_\_\_\_\_\_\_\_\_

GET \_\_\_\_\_\_\_\_\_\_\_ photos#edit

PATCH/PUT / photos/:id \_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_ /photos/:id photos#destroy

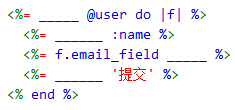
1. 根据以下代码,请叙述render和redirect\_to的效果分别是什么:



1. 现在需要在视图文件`index.html.erb`中列举所有的书信息和后面的相关操作, 请完成以下代码的下划线部分。



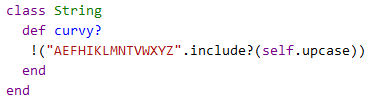
1. 构造一个表单, 提交用户的name字段和email信息,在视图文件`new.html.erb`完成以下代码中的下划线部分。



1. 完成以下视图(控制器)测试文件`site\_layout\_test.rb`



1. String 类包含一个方法,给定一个字符串，判断它不包含单词curvy中的字母。该类定义如下：

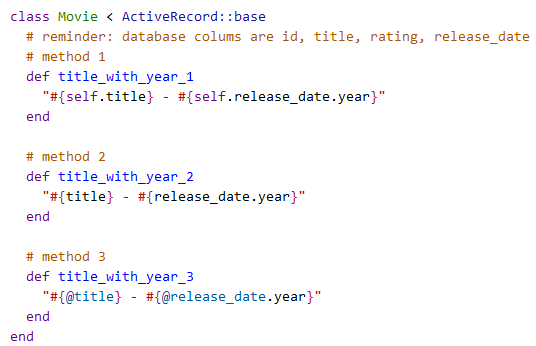


如何正确调用方法curvy？

1. 完成以下模型测试文件user\_test.rb



1. 下面3个问题是基于如下代码。写了三个候选方法，返回电影的片名和发行年份，例如， "Star Wars - 1977"。假定电影发行日期是有效的，且Ruby Time实例返回年份是整数。



（1）下面哪句话准确描述了method 1的行为?

A． 按设计意图工作。

B． 不按设计意图工作。

（2）下面哪句话准确描述了method 2的行为?

A． 按设计意图工作。

B． 不按设计意图工作。

（3）下面哪句话准确描述了method 3的行为?

A．按设计意图工作。

B．按设计意图工作，如果把attr\_accessor :title, :release\_date 加到模型代码中。

C．不按设计意图工作，即使把attr\_accessor 加到模型代码中。

1. 使用Ruby On Rails搭建一个博客系统

40. rails中创建一个新项目(blog)并使用mysql数据库的命令

41.ruby语言中，以下分别对应的是什么类型的变量？请连线对应.

一般小写字母、下划线开头 类变量（Class variable）

$开头 全局变量（Global variable）

@开头 常数（Constant）

@@开头 变量（Variable）

大写字母开头 实例变量（Instance variable）

42.在Ruby 语言中，load , require, include，extend 四种方法的区别是什么？

43. 分析如下HTTP请求报文由哪些部分构成？

1. GET/sample.jspHTTP/1.1

2. Accept:image/gif.image/jpeg,\*/\*

3. Accept-Language:zh-cn

4. Connection:Keep-Alive

5. Host:localhost

6. User-Agent:Mozila/4.0(compatible;MSIE5.01;Window NT5.0)

7. Accept-Encoding:gzip,deflate

9. username=jinqiao&password=1234

1. 举例说明Ruby支持修改内置的类的理由。(评分标准：大致以rails举例即可)

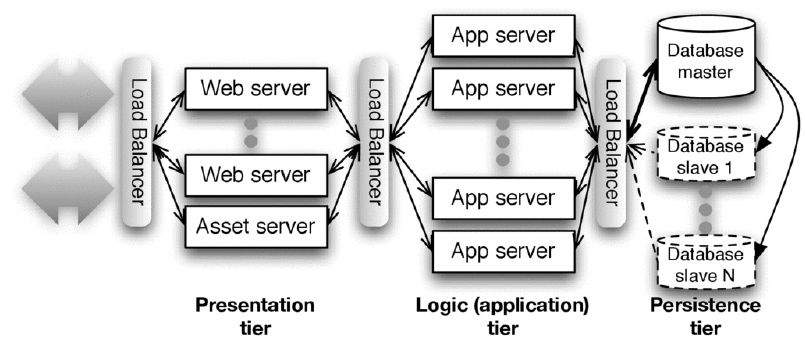
## ⯀ 框架与开发

45. 云计算解释，结合Cloud9的例子说明。

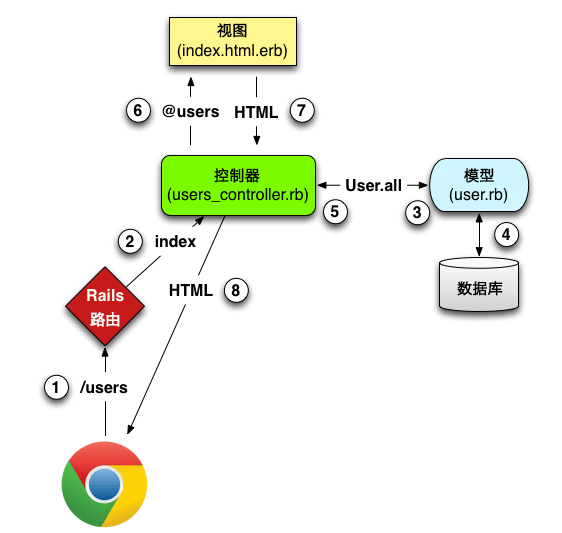
1）相比传统自建数据中心而言，为什么云计算能以更低费用提供更多的计算资源？

2）采用云计算，任何一个应用都能“无限地横向扩展”吗？为何可以或不可以？

46.三层无共享（Shared-Nothing）架构的解释（参照下面的图）。



1. 哪三层？为何叫“无共享”?
2. 该架构为何比较适合大规模的SaaS 应用?
3. 相比其他层，哪一层是不容易横向扩展？为什么？
4. 根据以下的架构图和Rails应用框架的原理，填写下面8个步骤的缺失内容。

****

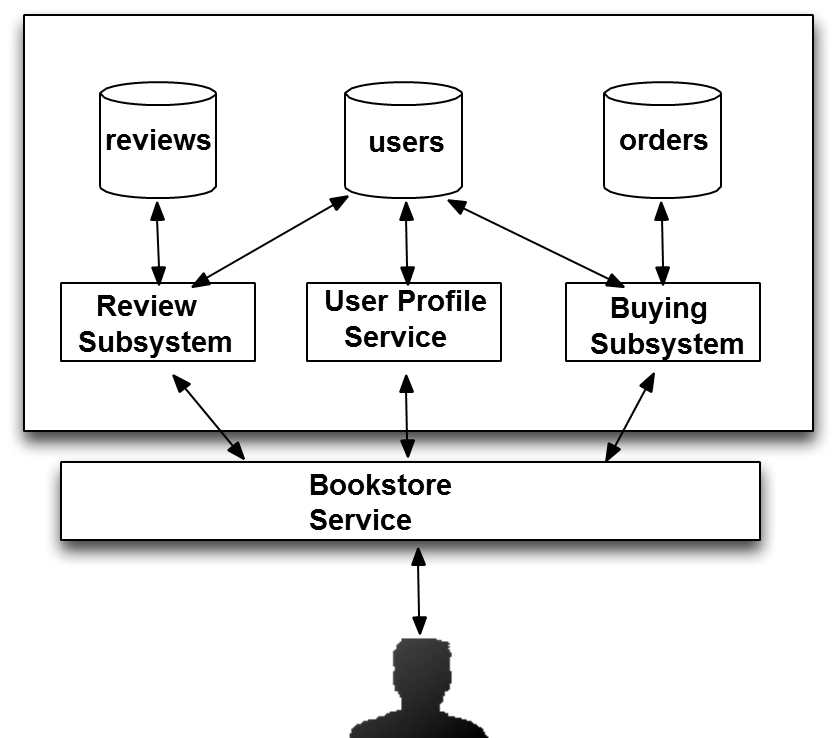
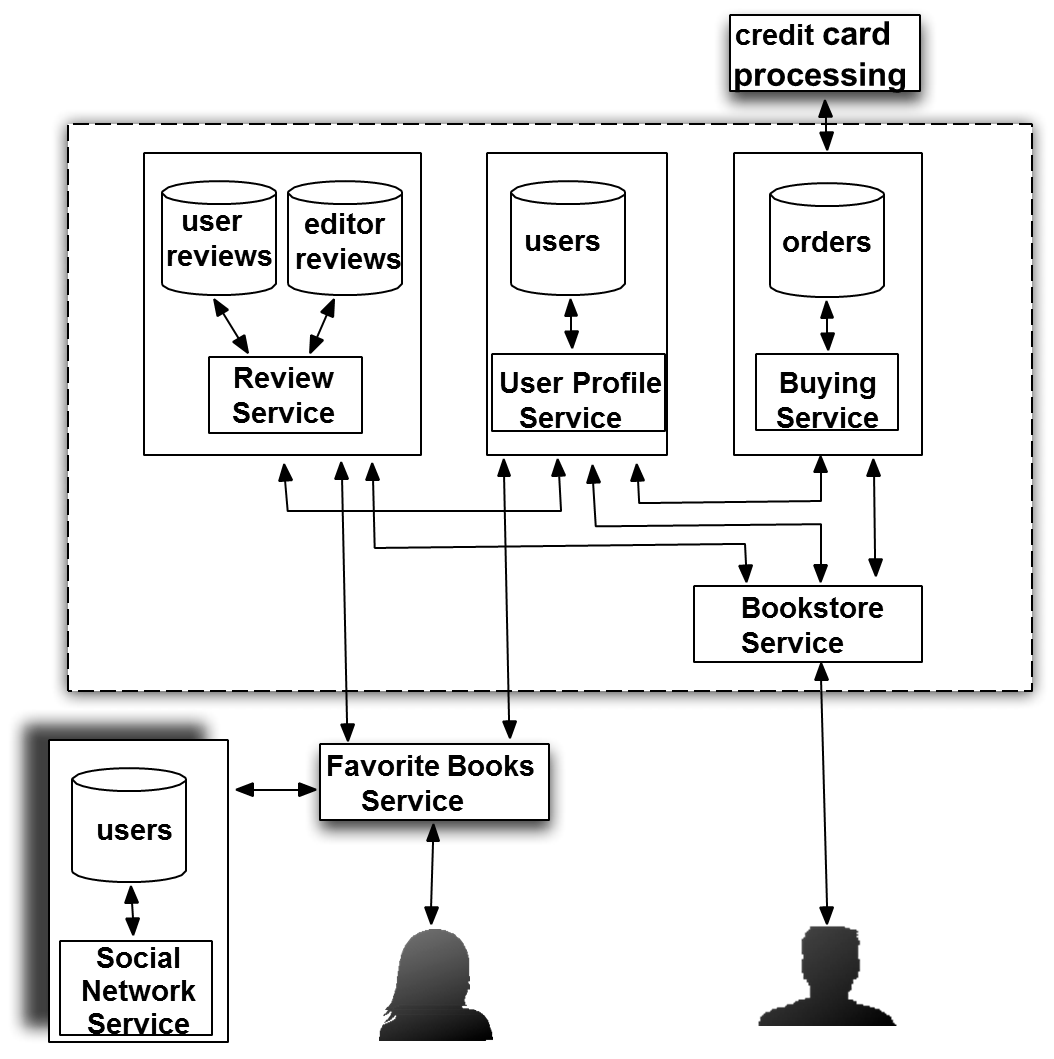
已知users 对应于数据库中的一个表（table）；id、name 和 email 是表中的列（column）;

|  |  |  |
| --- | --- | --- |
| Users 资源中页面和 URL 的对应关系 | | |
| URL | **动作** | **作用** |
| /users | index | 列出所有用户 |
| /users/1 | show | 显示 ID 为 1 的用户 |
| /users/new | new | 创建新用户 |
| /users/1/edit | edit | 编辑 ID 为 1 的用户 |

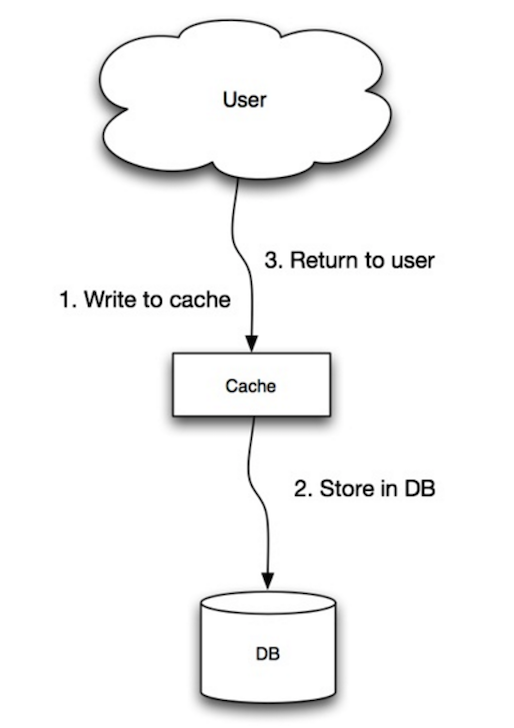
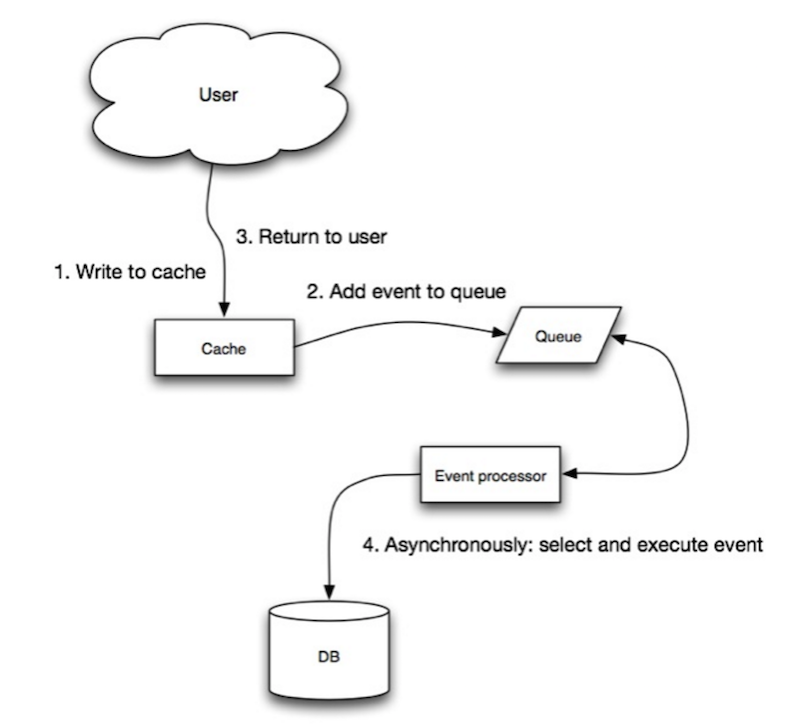
当用户发出一个请求: http://example.com/users 后的如下步骤是：

1. 浏览器向 /users 发送请求；
2. Rails 的路由把 /users 交给 （ ）控制器的（ ）动作处理；
3. （ ）动作要求 （ ）模型检索所有用户（ ）；
4. （ ）模型从数据库中读取所有用户；
5. （ ）模型把所有用户组成的列表返回给控制器；
6. 控制器把所有用户赋值给 （ ）变量，然后传入（ ）视图；
7. 视图使用嵌入式 Ruby 把页面渲染成 HTML；
8. 控制器把 HTML 送回浏览器。

***48.左下图和右下图分别是两种设计方案。左下图是一个单一系统架构，三个子系统都采用统一的API（Bookstore Service）。右下图是一个面向服务设计架构， 三个子系统采用各自独立的API，请解释两个方案的优缺点，从功能升级和横向扩展等方面比较。***

***49左下图和右下图分别是两种提升软件性能（服务用户读写请求的能力）设计方案。当左下图方案直接写缓存再写数据库；右下图方案是写缓存后，把写数据的任务写入一个队列，让后台的进程去读队列中的任务，再写入数据库。请分析和对比这两种方案各自的特点和局限，重点回答一下三个问题。***



***1） 你认为哪一种架构能支持更多用户请求，为什么?***

***2） 对比分析它们各自的特点以及它们分别要解决哪些主要问题?***

***3） 如果这两种方法仍不满足要求，你认为如何改进？***

50.Django框架中，路由在哪个文件配置？path和re\_path有何区别？

51. 我们知道软件开发中最常见的两个问题分别是：

1.需求和开发脱节：用户想要的功能没有开发，开发的功能并非用户想要。

2.开发和测试脱节：从开发到测试周期过长，测试自动化程度低。

请回答TDD和BDD分别可以解决哪一个问题，为什么？

52.问：在软件开发中我们会进行版本控制。一般会使用Git做版本控制。假设当前linux系统下有个主分支名叫master，并且现在我们所做的改动在分支login(目前正处于)下。如何将login下的改动合并到master下，并且推送到git上。请写出命令。

53.在阐述用户故事的时候，我们希望能满⾜SMART的要求，即“Specific，

Measurable，Achievable，Relevant，Timeboxed”，请根据这些原则对下⾯选

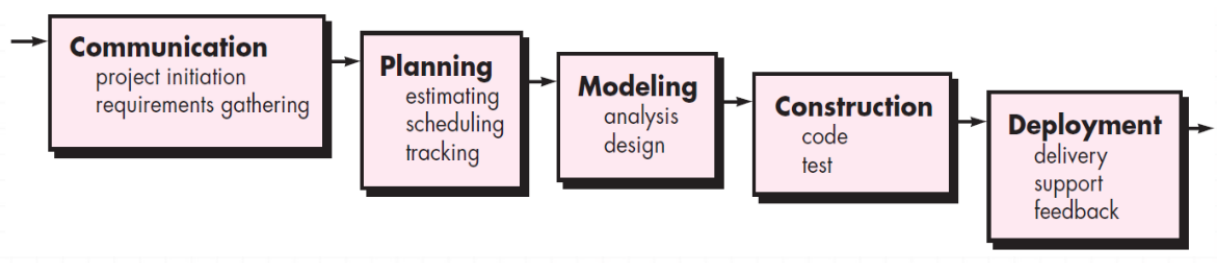
课系统的⽤户故事做删改课系统的⽤户故事做删改

我登陆后能看到排序后的所有课程，在搜索栏能搜索找到⾃⼰想选的课程。搜

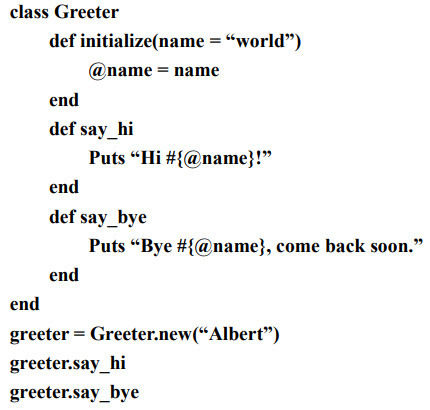
索课程⽤时很短，在选择课程后能⻢上显示实时课表。并可以在另⼀个模块找

到⾃⼰之前课程的得分。

54.瀑布模型的过程如下，请指出你认为瀑布模型存在的问题。



55.给出下面这段代码的执行结果.



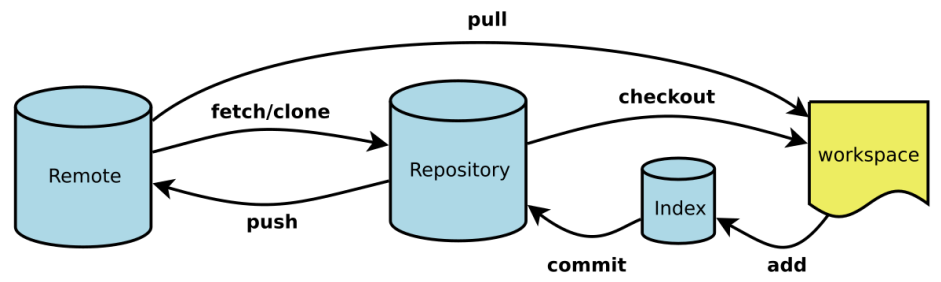
1. 软件开发中的重构是什么？为什么要重构？有哪些基本原则？

57“从外到里”的BDD开发模式中，“从外到里”指的是什么，这与传统先代码后测试相比有什么优势？

58.请指出 RESTFUL 设计风格的特点

59.Flask 与 Django 框架对比分析

***60.简述git工作流程，工作区、暂存区、本地仓库、远程仓库之间如何交互。***



61.介绍依赖倒置原则

(标准: 5分 能完整描述依赖倒置原则的定义，最好能用代码描述。

4 分：正确描述依赖倒置原则的定义。）

62. Web框架Ruby on Rails为用户创造了什么价值？

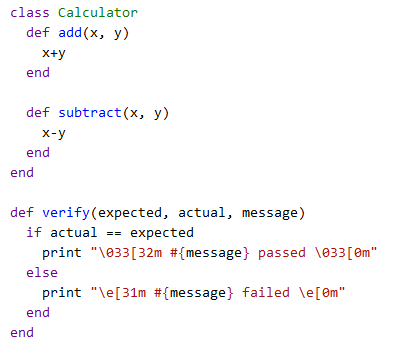
## ⯀ 测试

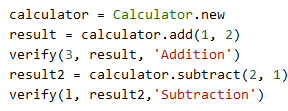
63. 学生选课系统作为一种现代化的教学技术，越来越受到人们的重视。学生对选课系统的使用大多数在于选课及查询。如果让你来测试选课系统站内的搜索系统，请问你能想到从哪些方面来进行测试？

64.为什么要测试，什么时候测试以及如何测试。

***65.下面是一个有加和减功能的计算器，以及手工编写的测试程序。***

1. ***请用Ruby 的MiniTest自动测试方法重写测试程序；***
2. ***若计算器要增加乘法和除法功能，如何修改Calculator 和 自动测试程序。***





66. 简述下面代码的作用

describe "Home page" do

it "should have the content 'Sample App'" do

visit '/static\_pages/home'

expect(page).to have\_content('Sample App')

end

end

67.编写自动化测试有哪些好处，何时使用自动化测试.

68.为什么要做app的稳定性测试？

69.软件测试要经过哪些步骤?这些测试与软件开发各阶段之间有什么关系?

70.行为驱动和测试驱动的区别和联系

## ⯀ 两种途径

71.2种软件设计方式Traditional programming（人总结规律用代码写出程序的方法）和

Machine learning（用数据集训练的程序的共同点和不同点）的相同点与不同点？这两种方式使用通常是独立的吗？为什么？举例说明(无标准答案)

图示, 文本, 聊天或短信

描述已自动生成 图示, 文本, 聊天或短信

描述已自动生成

72.观察讨论“用深度学习构建的识别手写数字的模型”，（1）为什么1980~2000年的计算机人工智能博士论文关于手写数字识别率都不上80%？（2）人总结规律用代码写出程序的方法与用数据集训练的程序的共同点和不同点在哪？（3）你认为要利用这个模型实现一个有价值的软件还需要做哪些工作？