

软件工程课程概论 5

1、 以下关于群体化学习平台LearnerHub的描述，哪些是正确的。

- ☐ A、开源软件开发平台
- ☒ B、学习问题交流和讨论平台
- ☒ C、学习经验和知识分享平台
- ☐ D、开源软件托管平台

2、 群体化学习平台LearnerHub提供了哪些功能。

- ☒ A、问题提问
- ☒ B、问题回答
- ☒ C、评论
- ☒ D、学习知识查询

3、 借助群体化学习平台LearnerHub，可以解决哪些方面的学习问题。

- ☒ A、解决学习中遇到的问题
- ☒ B、分享学习经验
- ☒ C、贡献实践成果
- ☒ D、对问题及回答进行评论

1、 阅读和标注优质开源代码是学习编程开发、提升软件工程素养的一种有效途径。

- ☒ A、该说法是正确的
- ☐ B、该说法是错误的

2、 基于Codepedia标注平台能够对以下哪些粒度的代码进行标注

- ☒ A、模块
- ☒ B、文件
- ☒ C、函数
- ☒ D、代码块
- ☒ E、代码行

1、 开源的群体协同开发模式在开源领域以及企业开发实践中广泛应用。以下说法哪些是错误的？

- ☒ A、要进行基于PR模式的开发，必须要首先Fork项目
- ☒ B、根项目管理者对Fork产生的副本项目具有管理权限
- ☒ C、提交PR的源分支和目标分支可以相同
- ☐ D、PR提交可以在同一个项目内部进行

1、 在群体协同开发过程中，开发任务的分解和安排对项目推进实施具有很大的影响。下面关于协同开发平台中的任务管理说法错误的是？

- ☒ A、在协同开发平台中，每一个任务就是一个软件缺陷
- ☒ B、在协同开发平台中，一个任务发布后不能再进行修改
- ☐ C、在协同开发平台中，任务截止前会发送通知给任务发布者和被指派者
- ☒ D、一个里程碑中只能有一个任务

三

第1关：本地版本库创建

100

任务要求

参考答案

评论 5

- 任务描述
- 相关知识
 - Linux基础操作
 - 创建某个目录
 - 创建文件
 - 进入目录
 - 使用Git前的准备
 - 安装
 - Git配置
 - 如何创建一个本地版本库
- 编程要求
- 评测说明

代码文件

命令行

已保存

```
1 #创建gitTraining文件夹
2 mkdir gitTraining
3
4 #进入gitTraining文件夹
5 cd gitTraining
6
7 #请添加初始化本地Git仓库的命令
8 #***** Begin *****#
9 git init
10
11 #***** End *****#
```

三

第2关：添加修改到暂存区域

100

任务要求

参考答案

评论 5

任务描述

相关知识

编程要求

测试说明

任务描述

现在我们已经有了一个本地仓库，但是现在它里面空空如也。那么我们到底要怎么去保存我们的修改呢？

本关任务：使用 Git 命令将本地文件添加到暂存区中。

相关知识

代码文件

命令行

已保存

```
1 #创建gitTraining文件夹
2 mkdir gitTraining
3
4 #进入gitTraining文件夹
5 cd gitTraining
6
7 #请添加初始化本地Git仓库的命令
8 #***** Begin *****#
9 git init
10
11 #***** End *****#
12
13 #创建helloGit.txt文件
14 touch helloGit.txt
15
16 #请添加提交helloGit.txt到暂存区的命令
17 #***** Begin *****#
18 git add helloGit.txt
19
20 #***** End *****#
21
```

三

第3关：提交修改到本地仓库

100

任务要求

参考答案

评论 5

任务描述

相关知识

编程要求

测试说明

任务描述

本地 Git 操作三部曲，是“修改-添加-提交”。现在你已经完成了前两个，相信你也迫不及待地想进行第三个操作了：将本地修改提交到本地仓库中去。

本关任务：将暂存区中的修改提交到本地仓库中。

相关知识

代码文件

命令行

已保存

```
1 #创建gitTraining文件夹
2 mkdir gitTraining
3
4 #进入gitTraining文件夹
5 cd gitTraining
6
7 #请添加初始化本地Git仓库的命令
8 #***** Begin *****#
9 git init
10
11 #***** End *****#
12
13 #创建helloGit.txt文件
14 touch helloGit.txt
15
16 #请添加提交helloGit.txt文件到暂存区的命令
17 #***** Begin *****#
18 git add helloGit.txt
19
20 #***** End *****#
21
22 #请添加提交helloGit.txt至本地仓库的命令
23 #***** Begin *****#
24 git commit -m "helloGit.txt"
25
26 #***** End *****#
```

三

第4关：克隆远程版本库

100

任务要求

参考答案

评论 5

任务描述

相关知识

编程要求

测试说明

任务描述

相关知识

代码文件

命令行

已保存

```
1 #请在下面的Begin/End内填写语句以将远程版本库clone到本地
2 #***** Begin *****#
3
4 git clone /tmp/sample.git
5
6 #***** End *****#
```

三

第5关：添加远程版本库

100

任务要求

参考答案

评论 5

任务描述

相关知识

编程要求

测试说明

任务描述

代码文件

命令行

已保存

```
1 #创建gitTraining文件夹
2 mkdir gitTraining
3
4 #进入到gitTraining文件夹
5 cd gitTraining
6
7 #将gitTraining初始化为一个本地仓库
8 git init
9
10 #请在下面的Begin/End内填写语句添加符合要求的主机名和远程仓库
11 #***** Begin *****#
12 git remote add git /tmp/sample.git
13
14 #***** End *****#
```

三

第6关：推送本地内容到远程仓库

100

任务要求

参考答案

评论 5

• 任务描述

• 相关知识

• 编程要求

• 评测说明

任务描述

在团队开发中一个很频繁、很普通的行为，就是将你的修改推送到远程仓库，以便于测试或者和团队中其他人协作。

本文任务：创建本地仓库并添加远程主机，然后创建 `helloGit.txt` 文件，并向其中写入 "Hello, Git! I am coming"，最后将其推送到远程仓库的 `master` 分支。

相关知识

推送本地内容时，会将所有未推送到远程仓库的内容，都提到远程仓库。它用到的命令是 `git push`，使用方法如下：

```
1. git push 远程仓库名 本地分支名 远程分支名
```

具体的使用方法如下：

代码文件

命令行

已保存

```
1 #创建gitTraining目录
2 mkdir gitTraining
3
4 #进入gitTraining目录
5 cd gitTraining
6
7 #将gitTraining初始化为一个本地仓库
8 git init
9
10 #创建helloGit.txt
11 touch helloGit.txt
12
13 #添加远程仓库
14 git remote add git /tmp/educoder.git
15
16 #将helloGit.txt添加到暂存区
17 git add helloGit.txt
18
19 #将helloGit.txt提交到本地仓库
20 git commit -m "hello Git!"
21
22
23 #请在下面的Begin/End内填写语句，将本地master分支的修改推送到
24 #远程仓库的master分支
25 #***** Begin *****#
26 git push git master master
27
28
29 #***** End *****#
```

三

第7关：拉取远程分支到本地

100

任务要求

参考答案

评论 5

• 任务描述

• 相关知识

• 编程要求

• 评测说明

任务描述

在前三个关卡，我们已经学习了，如何克隆版本库、添加远程仓库及推送本地内容到远程分支这三种操作。那么在团队协作中，如果别人对项目做了修改，而你需要将这些修改合并到你本地时，该怎么做呢？

本文任务：将远程分支内容拉取到本地。

相关知识

拉取远程仓库的内容到本地，需要使用 `git pull` 命令，其命令格式为：

代码文件

命令行

已保存

```
1 #创建gitTraining目录
2 mkdir gitTraining
3
4 #进入gitTraining目录
5 cd gitTraining
6
7 #将gitTraining初始化为一个本地仓库
8 git init
9
10
11
12 #请在下面的Begin/End内填写语句以拉取远程仓库git的master分支内容到本地
13
14 #***** Begin *****#
15 ## #创建helloGit.txt
16 touch helloGit.txt
17 # #添加远程仓库
18 git remote add git /tmp/educoder.git
19 #拉取远程仓库git的master分支内容到本地
20 git pull git master:master
21 #***** End *****#
```

1、

随着互联网、计算机基础设施、用户需求等多方面因素的影响，计算机软件系统在很多方面都发生了深刻的变化。下面哪些方面计算机软件系统发生了变化？

☒

A、软件系统的部署平台

☒

B、软件系统的部署运行形态

☒

C、软件系统的要素交互模式

☒

D、软件系统的代码规模

2、

关于计算机软件形态的变化，下面哪种说法是错误的？

☒

A、软件系统是一个纯粹的技术系统，与社会、物理等要素是相对独立的。

☒

B、集中同构系统是当前软件系统的主要部署和运行方式。

☐

C、软件系统的规模和复杂度正在快速增加。

☒

D、人类对于软件系统的需求和演化的认知已经比较全面

1、

近年来，软件开发方法和技术发生了很大的变化。下列关于软件开发过程和的说法哪些是错误的？

☐

A、敏捷开发方法是拥抱软件系统变化的一种开发模型

☒

B、螺旋模型解决了软件需求快速变化的问题

☒

C、DevOps模型严格区分软件开发、部署、运行和维护等阶段

☐

D、瀑布模型要求规范的需求文档

2、

软件开发技术不断推陈出新，对于提高软件开发效率和质量产生了重要作用。下面哪些说法是错误的？

☐

A、开源软件的开发采用了基于群智的软件开发模式

☒

B、依靠松散组合的互联网大众是无法开发出高质量软件产品的

☐

C、软件重用是提供软件开发效率和质量的重要手段

☐

D、设计模式是一种软件重用形式

从程序到软件 5

1、 开源软件可以任意的下载、传播和修改，无需遵循任何的要求。

- ☐ A、正确
- ☒ B、错误

2、 免费使用的软件属于开源软件。

- ☐ A、正确
- ☒ B、错误

3、 关于开源软件的开发，以下哪些阐述是错误的。

- ☐ A、互联网大众参与贡献的结果
- ☐ B、需要依托开源托管平台
- ☒ C、采用团队的组织方式
- ☐ D、任何人都可参与修改代码

1、 软件质量等同于程序质量。

- ☐ A、正确
- ☒ B、错误

2、 软件之所以需要文档，是处于以下的原因。

- ☒ A、记录开发成果
- ☒ B、分享和交流开发成果
- ☒ C、分析和发现开发成果中的问题
- ☐ D、减少开发工作量

1、 以下哪些软件工具可以完成程序代码的静态分析工作。（多选题）

- ☐ A、JUnit
- ☒ B、Sonarqube
- ☒ C、CheckStyle
- ☐ D、Eclipse

2、 结对编程可有效地支持代码的人工审查工作。

- ☒ A、正确
- ☐ B、错误

3、 以下哪一项是通过运行程序来发现代码问题的。

- ☐ A、代码检查
- ☐ B、代码扫描
- ☐ C、人工检查
- ☒ D、程序测试

4、 人工审查代码包含以下哪几个步骤。

- ☒ A、发现缺陷和问题
- ☒ B、阅读和理解代码
- ☐ C、运行程序代码
- ☒ D、提出改进的建议

1、 以下的代码片段存在哪些方面的编码风格问题（多选）

```
#include "stdafx.h"
#include "stdio.h"
void test
int _tmain
(int argc,
_TCHAR* argv[])
{ test(); return
0; } char C[25]
[40];void d(int x,
int y){C[x][y]=
C[x][y+1]=32;}
int f(int x){return
(int)x*x*.08;}
void test(){int i,j;
char s[5]="TEST";
for(i=0;i<25;i++)
for(j=0;j<40;j++)
C[i][j]=s[(i+j)%4];
for(i=1;i<=7;i++)
{d(18-i,12);
C[20-f(i)][i+19]=
C[20-f(i)][20-i]=32;
}d(10,13);d(9,13);
d(8,14);d(7,15);
d(6,16);d(5,18);d(5,20);          d(5,22);d(5,26);
d(6,23);d(6,25);d(7,25);for(i=0;i<25;i++,printf("\n"))
for(j=0;j<40;printf("%c",C[i][j+i]));
scanf("%c", &s[0]);
```

- ☒ A、没有缩进
- ☒ B、没有有用有意义的名称
- ☐ C、没有采用模块化编程
- ☒ D、一行多条语句

2、 以下哪些方法可以有效地保证和提高代码质量（多选）

- ☒ A、互相检查代码
- ☒ B、代码重用
- ☒ C、采用面向对象的程序设计方法
- ☐ D、多编写程序

3、 代码重用有何实际的意义和价值

- ☒ A、提高开发效率
- ☒ B、提高代码质量
- ☐ C、学习他人的编码风格
- ☒ D、减少开发工作量

1、 编程风格影响程序的可读性，但不会影响最终软件产品的质量。

- ☐ A、正确
- ☒ B、错误

2、 关于程序标识符命名，以下说法不正确的是。

- ☐ A、命名应使用标准英文单词或缩写，不要使用拼音
- ☐ B、命名应遵循望文知义原则，即名称应含义清晰、明确
- ☐ C、命名都不要过长，应控制在规定的最大长度以内
- ☒ D、所有命名都应尽量使用缩写

1、 代码的结构清晰、可理解性好，这类代码的内部质量高。

☒ A、正确

☐ B、错误

2、 程序具有良好的可扩展性且易于维护，则该程序的外部质量高。

☐ A、正确

☒ B、错误

3、 在编写代码的过程中，对程序做适当的注释可有助于提高代码的哪些属性（多选）。

☒ A、代码的内部质量

☐ B、代码的外部质量

☒ C、代码的可理解性

☐ D、代码的可扩展性

软件工程概述 4

1、 开源软件的开发是一项基于互联网大众群体创作的过程。

- ☒ A、正确
- ☐ B、错误

2、 针对特定的软件（如微信），构思软件需求是一项软件创作的活动。

- ☒ A、正确
- ☐ B、错误

3、 基于软件设计的模型和文档来编写程序是一项什么样的活动。

- ☒ A、生产活动
- ☐ B、创作活动
- ☐ C、既是生产活动也是创作活动

4、 根据确定的软件需求来进行软件设计，这是一项什么样的活动。

- ☐ A、生产活动
- ☐ B、创作活动
- ☒ C、既是生产活动也是创作活动

1、 计算机辅助软件工程对于软件开发而言有何作用。

- ☒ A、提高开发效率
- ☒ B、提高开发质量
- ☒ C、减少开发工作量
- ☒ D、减低开发成本

2、 Microsoft Office也是一个CASE工具。

- ☒ A、正确
- ☐ B、错误

3、 SonarQube是一个CASE工具，以下哪一项不是该工具提供的功能。

- ☐ A、分析代码的内部质量
- ☐ B、发现代码中的缺陷
- ☒ C、修复代码中的缺陷
- ☐ D、产生代码质量的报告

4、 Eclipse和Visual Studio都是CASE环境。

- ☒ A、正确
- ☐ B、错误

1、 软件工程中“工程”的内涵包括哪些方面。

- ☒ A、遵循过程来开发
- ☒ B、保证产品质量
- ☐ C、采用最新的技术
- ☒ D、借助工具支持

2、 软件工程要求采用各种技术和文档规范，以进行规范化的开发，产生规范化的软件产品。

- ☒ A、正确
- ☐ B、错误

3、 软件工程的原则对于软件开发而言可产生什么样的成效。

- ☐ A、保证软件开发不会出错
- ☒ B、确保软件开发的质量
- ☒ C、有效应对软件系统的复杂性
- ☐ D、减少软件开发的工作量

4、 以下哪些属于软件工程的基本构成要素。

- ☒ A、过程
- ☐ B、建模语言
- ☐ C、程序设计语言
- ☒ D、辅助工具

1、 编写程序仅仅是软件开发的一项工作，软件开发还牵涉其他方面的工作。

- ☒ A、正确
- ☐ B、错误

2、 以下哪些是导致软件危机的主要原因。

- ☐ A、软件工程师人数太少
- ☒ B、软件过于复杂
- ☒ C、对软件及其开发认识不够
- ☒ D、缺乏理论指导和技术的支持

3、 在现阶段软件危机依然存在。

- ☒ A、正确
- ☐ B、错误

4、 软件危机有哪些主要的表现形式。

- ☒ A、软件开发成本太高
- ☒ B、软件质量无法保证
- ☐ C、待开发软件项目的数量太多
- ☒ D、软件项目无法按时交付

软件过程模型和开发方法 3

1、 群体化开发方法所依赖的软件开发者是开放的，而不是封闭的。

- ☒ A、正确
☐ B、错误

2、 群体化开发方法需要依赖于互联网平台来吸引大量的软件开发者群体。

- ☒ A、正确
☐ B、错误

3、 关于群体化开发方法，以下哪些阐述是正确的。

- ☒ A、开发者包括核心开发者和外围开发者
☒ B、开发成果来自于群体的智慧
☒ C、群体化开发方法的成功实践之一是开源软件
☐ D、由于开发群体数量大，质量难以保证

1、 关于群体化开发的Issue机制，以下哪些描述是错误的。

- ☐ A、Issue用于支持开发任务管理
☒ B、Issue就是指某个错误
☐ C、任何人都可以提出Issue
☐ D、Issue机制体现了群体创作的理念

2、 基于Git的分布式版本管理支持创建同一个软件的多个代码仓库，每个仓库由不同的开发人员进行管理，但是通常只有一个中心代码仓库。

- ☒ A、正确
☐ B、错误

3、 关于Pull/Request机制，以下哪些描述是正确的。

- ☒ A、支持群体的分布式协同开发
☒ B、允许开发者将其贡献的代码汇聚到中心仓库中
☐ C、开发者只要提交了Pull/Request，其代码就会被接纳到中心仓库中
☒ D、开发者所提交了Pull/Request必须进行质量分析

1、 开源软件是通过群体化方法开发出来的。

- ☒ A、正确
☐ B、错误

2、 参与群体化开发人员的水平参差不齐，所编写代码的质量也不一样，因而群体化开发方法无法保证代码的质量。

- ☐ A、正确
☒ B、错误

3、 软件开发知识分享具有哪些方面的特点。

- ☒ A、任何人都可以参与知识分享
☒ B、所分享知识的质量参差不齐，可能会有低质量甚至错误的知识
☐ C、只有核心人员才有资格提问
☒ D、需要依赖于互联网平台

- 1、敏捷开发方法以代码为中心。
- ☒ A、正确
- ☐ B、错误
- 2、敏捷开发方法具有哪些优点。
- ☒ A、应对需求变化
- ☒ B、快速交付软件产品
- ☐ C、软件质量要优于其他方法
- ☐ D、软件更加易于维护
- 3、Scrum、极限编程均属于特殊的敏捷开发方法，因而具有敏捷开发方法的基本特征。
- ☒ A、正确
- ☐ B、错误

- 1、Scrum开发方法具有以下的特点。
- ☒ A、迭代开发
- ☒ B、每个“Sprint”周期只实现某个功能
- ☒ C、持续交互产品
- ☐ D、以质量为优先
- 2、关于测试驱动开发，哪些描述是不正确的。
- ☒ A、只需要编写测试代码，无需编写功能代码
- ☒ B、需要同时编写测试代码和功能代码
- ☐ C、先编写测试代码，再编写功能代码
- ☒ D、先编写功能代码，再编写测试代码

- 1、以下哪些过程模型支持迭代开发。
- ☐ A、瀑布模型
- ☐ B、增量模型
- ☒ C、原型模型
- ☒ D、螺旋模型
- 2、以下哪些模型可以有效应对软件需求的变化。
- ☐ A、瀑布模型
- ☒ B、迭代模型
- ☒ C、原型模型
- ☐ D、增量模型
- 3、瀑布模型、迭代模型、螺旋模型都是以文档为中心的，而非以代码为中心的，属于重型的软件开发方法。
- ☒ A、正确
- ☐ B、错误

- 1、UP模型是以代码为中心的一类开发方法。
- ☐ A、正确
- ☒ B、错误
- 2、软件过程不等于软件生命周期，前者是针对开发而言的，后者是针对软件而言的。
- ☒ A、正确
- ☐ B、错误
- 3、关于增量模型和迭代模型，以下哪些描述是正确的。
- ☐ A、都支持需求变化
- ☒ B、都支持快速交付产品
- ☐ C、都支持迭代软件开发
- ☒ D、都属于重型软件开发方法

软件需求工程基础 4

1、 需求工程完成后将输出哪些方面的软件制品。

- ☒ A、软件需求模型
- ☒ B、软件需求文档
- ☒ C、软件需求原型
- ☐ D、软件设计模型

2、 软件需求文档采用图文并茂的方式，详细描述了软件系统的功能性和非功能性需求。

- ☒ A、正确
- ☐ B、错误

3、 软件需求缺陷存在哪些方面的表现形式。

- ☒ A、没有如实地记录用户所提出的任何需求
- ☒ B、对软件需求的描述与用户的理解不一致
- ☒ C、多个软件需求项之间存在冲突
- ☒ D、对软件需求的描述不详尽，导致不清楚需求细节

1、 如果软件需求存在缺陷，会对软件开发带来什么样的后果。

- ☒ A、导致设计存在缺陷
- ☒ B、导致代码存在缺陷
- ☒ C、影响软件开发进度
- ☒ D、无法满足用户要求

2、 软件需求确认 (Requirement Validation) 和软件需求验证 (Requirement Verification) 二个概念的内涵是一样的。

- ☐ A、正确
- ☒ B、错误

3、 软件需求的追溯管理涉及哪些工作。

- ☒ A、追溯谁提出了需求变更
- ☒ B、追溯对需求进行了哪些变革
- ☒ C、追溯需求变更会对开发产生什么样的影响
- ☒ D、追溯需求变更的可行性

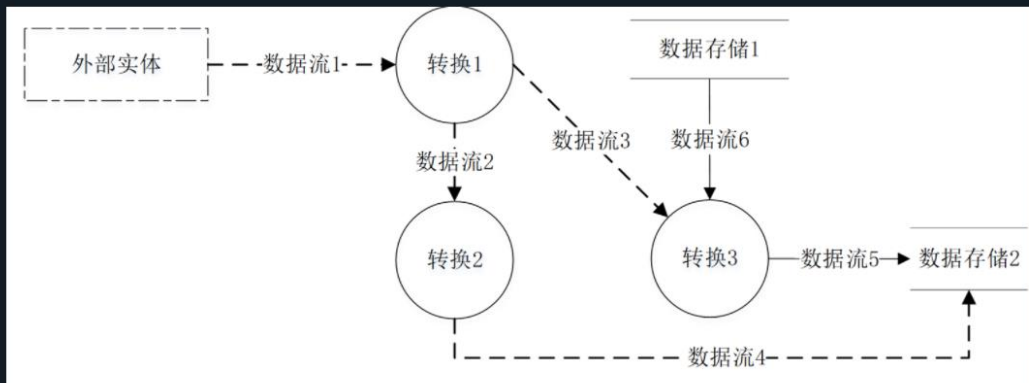
1、 结构化需求分析方法学有多种，面向数据流的需求分析方法学仅仅是其中的一种。

- ☒ A、正确
- ☐ B、错误

2、 关于面向数据流的需求分析方法学，以下哪些描述是正确的。

- ☒ A、该方法学只能用于描述和分析功能需求，无法刻画和分析非功能性需求
- ☒ B、软件系统的功能对应于有何数据以及对数据进行什么样的处理
- ☐ C、该方法学可以用于描述和分析功能性和非功能性需求
- ☐ D、软件系统的性能对应于有何数据以及对数据进行什么样的处理

3、 以下是一张数据流图，图中哪些图形化符号画错了。



- ☒ A、外部实体
- ☐ B、转换
- ☒ C、数据流
- ☐ D、数据存储

1、 面向数据流的需求分析方法学体现了哪些方面的需求分析思想和原则。

- ☒ A、抽象建模
- ☒ B、自顶向下、逐步求精
- ☐ C、软件重用
- ☐ D、模块化

2、 数据流图在精化过程中，需要保持哪些方面的平衡？

- ☐ A、数据流的平衡
- ☐ B、转换的平衡
- ☐ C、外部实体的平衡
- ☒ D、父图和子图的平衡

3、 类是对象的实例，对象是类的模板。

- ☐ A、正确
- ☒ B、错误

1、 以下关于UML的陈述，哪些是正确的。

- ☒ A、UML是一种建模语言，而不是程序设计语言
- ☒ B、UML支持对软件系统的多个观点进行建模，有助于获得软件系统的完整视图
- ☐ C、UML模型可在计算机上执行
- ☒ D、UML可建立图形化的软件模型，有助于加强对软件的直观理解和认识

2、 面向对象的方法既可对软件需求进行建模和分析，也可以对软件设计及实现进行建模和分析。

- ☒ A、正确
- ☐ B、错误

3、 UML提供了哪些图以对软件需求进行建模和分析。

- ☒ A、用例图
- ☒ B、类图
- ☒ C、状态图
- ☐ D、构件图

4、 UML主要支持对软件的功能需求进行建模，无法对非功能性需求（如性能等）进行建模。

- ☐ A、正确
- ☒ B、错误

1、 开展软件需求获取和分析，不仅需要需求工程的相关知识，也需要软件所对应领域工程的知识。

☒ A、正确

☐ B、错误

2、 需求获取工作可以一次性完成。

☐ A、正确

☒ B、错误

3、 下列哪些属于需求工程的活动。

☒ A、需求获取和构思、需求建模和分析

☒ B、需求获取和构思、基于需求构建软件原型

☒ C、需求建模与分析、软件需求的管理

☐ D、基于需求的软件设计、软件需求评审

1、 软件需求工程需要具备哪些方面的知识和能力。

☒ A、需求工程知识

☒ B、领域知识

☐ C、软件设计能力

☒ D、与用户沟通和协商的能力

2、 需求工程需要提供哪些技术手段来支持软件需求的获取和分析。

☒ A、抽象和建模技术

☒ B、系统的需求工程步骤

☒ C、CASE工具和规范

☒ D、需求分析的策略

3、 需求工程的CASE工具可以提供哪些方面的辅助和支持。

☒ A、编写需求文档

☒ B、绘制需求模型

☒ C、跟踪需求变化

☐ D、开展需求的设计

1、 软件需求可能来自于哪些软件利益相关方。

☒ A、用户

☒ B、客户

☐ C、软件项目经理

☒ D、软件开发者

2、 以下哪一项不属于合法的软件需求。

☐ A、用户需要某项功能

☐ B、用户需要某项性能

☐ C、用户要求在某个时间点交付产品

☒ D、用户要求采用某种设计模式和技术来开发软件

3、 以下关于软件需求的描述，哪些项是不正确的。

☐ A、软件需求易于变化

☐ B、开源软件的需求通常来自于软件开发工程师

☐ C、同一个需求可能来自于多个利益相关者，可能会存在冲突

☒ D、软件需求一旦确定就不允许变化

1、需求工程师必须获得完整、准确和一致的软件需求，才能进行后续的软件开发。

☐ A、正确

☒ B、错误

2、遗留的硬件系统也可以作为软件的利益相关者，对待开发软件系统的需求提出要求。

☒ A、正确

☐ B、错误

获取软件需求 4

1、 可以采用哪些手段来描述获得的初步软件需求。

- ☐ A、UML类图
- ☒ B、UML的用例图
- ☒ C、自然语言
- ☒ D、软件原型

2、 获取软件需求该项工作结束后， 软件需求工程将可能产生哪些软件制品。

- ☒ A、UML的用例图
- ☐ B、UML的类图
- ☒ C、用自然语言描述的初步软件需求文档
- ☒ D、软件原型

3、 用自然语言描述初步的软件需求存在哪些不足。

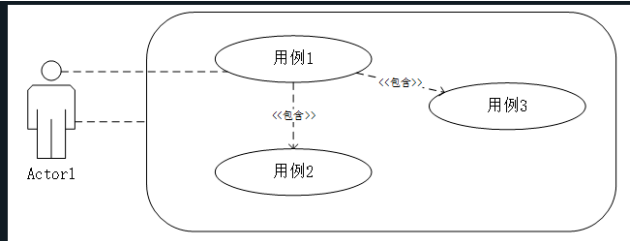
- ☐ A、有些初步软件需求无法用自然语言表述
- ☒ B、自然语言描述存在二义性和歧义性问题
- ☐ C、自然语言描述无法给出初步软件需求的详实信息
- ☒ D、自然语言描述存在不直观的问题

4、 与自然语言描述相比较， 软件原型的初步需求描述有何优点。

- ☒ A、直观， 易于理解初步软件需求
- ☒ B、易于发现初步软件需求描述中存在的问题
- ☐ C、该方法比较简单
- ☒ D、易于帮助用户导出更多的软件需求

1、

以下是一张UML用例图， 请指出该图存在哪些问题。



- ☒ A、执行者与用例之间的边不正确
- ☒ B、系统边界画的不正确
- ☐ C、用例之间的关系不正确
- ☐ D、用例画的不正确

2、 在用例图中， 外部执行者与用例之间的边表示什么样的内涵。

- ☐ A、依赖关系
- ☐ B、继承关系
- ☒ C、触发关系
- ☒ D、交互关系

3、 如果某项初步软件需求存在可行性问题， 需求工程师和用户就需要斟酌是否将该项需求作为最终的软件需求。

- ☒ A、正确
- ☐ B、错误

1、 用户提出的任何要求都可以作为软件需求。

- ☐ A、正确
- ☒ B、错误

2、 相比较而言，软件的功能性需求比非功能性需求更为重要。

- ☐ A、正确
- ☒ B、错误

3、 非功能性需求包括哪些方面的需求。

- ☐ A、软件行为需求
- ☒ B、软件开发约束需求
- ☒ C、软件质量需求
- ☐ D、软件结构需求

4、 任何软件需求都有其出处，也即都有提出该需求的相应利益相关者。

- ☒ A、正确
- ☐ B、错误

1、 需求工程师需要与用户或客户一道，通过调研分析，明确待开发软件系统欲解决的问题，并确保问题有意义、有价值。

- ☒ A、正确
- ☐ B、错误

2、 只要是用户提出的问题，都值得交由软件来加以解决。

- ☐ A、正确
- ☒ B、错误

3、 任何问题都可以找到基于软件的解决方法。

- ☐ A、正确
- ☒ B、错误

1、 其他的物理或软件系统也可以作为软件系统的利益相关者。

- ☒ A、正确
- ☐ B、错误

2、 软件开发者不能作为软件系统的利益相关者，提出软件系统的需求。

- ☐ A、正确
- ☒ B、错误

3、 有哪些方式可以获得软件需求。

- ☒ A、听取客户或用户的意见
- ☒ B、听取软件开发者的建议
- ☒ C、听取互联网大众的意见
- ☐ D、听取软件测试人员的意见

1、 获取软件需求难在哪些方面。

- ☒ A、 软件利益相关者说不清楚软件需求是什么
- ☒ B、 软件利益相关者区分不清哪些是软件需求、哪些不是软件需求
- ☒ C、 软件利益相关者不掌握领域知识，导致提不出软件需求
- ☒ D、 软件利益相关者没有投入时间来想定和构思软件需求

2、 有哪些常见的软件需求获取方法。

- ☒ A、 与软件利益相关者的面谈和会议
- ☒ B、 针对软件利益相关者的业务调查问卷
- ☒ C、 开发者实际体验业务
- ☒ D、 开发者构思软件需求

3、 获取软件需求包含哪些方面的工作。

- ☒ A、 开展业务调查，明确软件开发的目的是和动机，即明确软件要解决的问题
- ☒ B、 采取多种方式和手段，从软件利益相关者哪里导出和构思软件需求
- ☐ C、 对获取的软件需求进行分析，以发现不一致和矛盾的软件需求
- ☒ D、 尽可能描述清楚所获得的每一个软件需求

分析软件需求 5

1、 哪些人员需要参与软件需求文档的撰写工作。

- ☒ A、用户
- ☒ B、客户
- ☒ C、需求工程师
- ☐ D、设计工程师

2、 哪些人员需要参与软件需求文档的评审

- ☒ A、用户
- ☒ B、软件测试工程师
- ☐ C、程序员
- ☒ D、软件需求工程师

3、 基于UML的软件需求模型可能包括哪些UML图

- ☐ A、用例图、顺序图、类图、部署图
- ☒ B、用例图、顺序图、类图、状态图
- ☐ C、用例图、顺序图、类图、构件图
- ☒ D、用例图、通信图、类图、状态图

1、 需要为每个用例建立一个或者多个的交互模型，以获得关于这些用例的详实需求信息。

- ☒ A、正确
- ☐ B、错误

2、 软件需求模型所涉及的分析类主要包括以下几种。

- ☒ A、实体类
- ☐ B、虚拟类
- ☒ C、边界类
- ☒ D、控制类

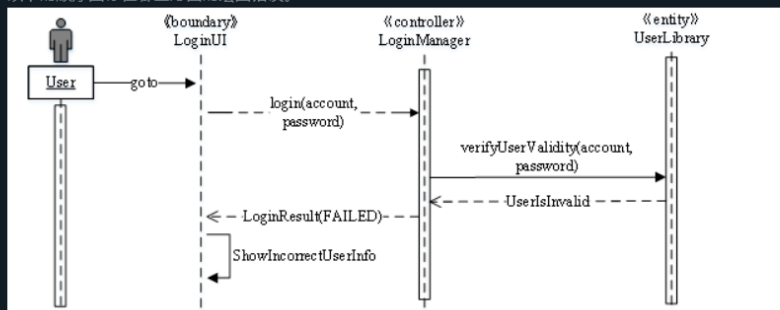
3、 如果类A的对象向类B的对象发消息M,那么意味着类B具有相应的职责和行为来处理消息M。

- ☒ A、正确
- ☐ B、错误

4、 如果类A的对象向类B的对象发送一个带有参数P的消息M,那么意味着类A和/或类B具有与P相对应的属性。

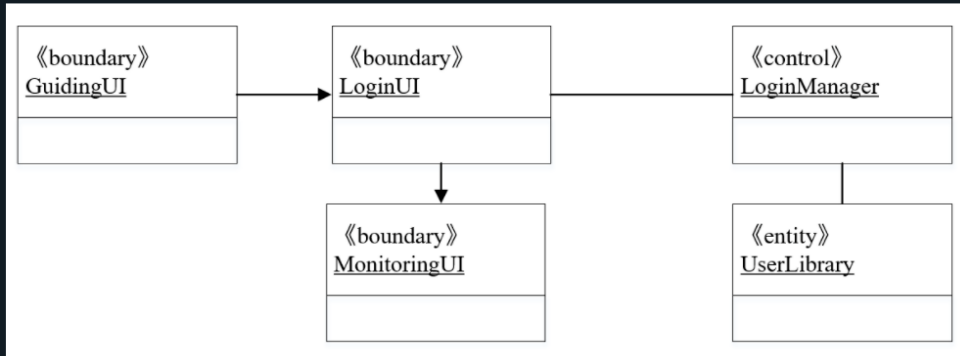
- ☒ A、正确
- ☐ B、错误

1、 以下的顺序图存在哪些方面的绘图错误。



- ☐ A、LoginResult(FAILED)返回消息绘制错误
- ☒ B、LoginUI对象没有活跃期
- ☒ C、LoginUI没有下划线
- ☒ D、Login (account, password) 消息绘制错误

2、 以下是一张分析类图，请说明哪些方面存在绘制错误。



- ☒ A、类名不应该带有下划线
- ☒ B、带箭头的关系图符错误
- ☐ C、不带箭头的关系图符错误
- ☐ D、每个类不应该带有《boundary》《control》《entity》等版类符号

3、 在需求分析阶段，需要为每一个类建立起状态图。

- ☐ A、正确
- ☒ B、错误

1、 对于用户而言，所有的软件需求都不可或缺，重要性是一样的。

- ☐ A、正确
- ☒ B、错误

2、 需求分析工程师可以考虑哪些方面的因素来确定软件需求的优先级。

- ☐ A、功能点的数量
- ☒ B、需求的重要性
- ☒ C、用户需要的紧迫程度
- ☐ D、需求的规模

3、 在软件开发过程中，具有高优先级的软件需求需要进行优先的分析、设计和实现。

- ☒ A、正确
- ☐ B、错误

1、 需求分析人员需要从哪些视点来详细地刻画软件需求。

- ☒ A、用例视点
- ☐ B、构件视点
- ☒ C、行为视点
- ☐ D、部署视点

2、 对软件需求进行多视点建模的目的是为了加强对软件需求的理解、描述和分析。

- ☒ A、正确
- ☐ B、错误

3、 用例视点的需求模型用UML的哪个图来描述。

- ☐ A、类图
- ☐ B、通信图
- ☒ C、用例图
- ☐ D、状态图

1、 哪些UML的图可以用来描述软件需求的行为模型。

- ☒ A、状态图
- ☐ B、类图
- ☒ C、顺序图
- ☒ D、通信图

2、 在需求分析阶段，可用类图来描述软件需求中的行为特征。

- ☐ A、正确
- ☒ B、错误

3、 状态图有助于加强对软件需求行为特征的理解、描述和分析。

- ☒ A、正确
- ☐ B、错误

1、 在获取软件需求阶段，所获得的初步软件需求还存在哪些方面的问题和不足。

- ☒ A、有遗漏的需求
- ☒ B、需求描述不准确
- ☒ C、需求之间的关系不清晰
- ☐ D、尚未形成需求文档

2、 分析软件需求是要在获取软件需求的基础上，对需求进行进一步的建模、精化和细化。

- ☒ A、正确
- ☐ B、错误

3、 分析软件需求的任务是。

- ☒ A、建立需求模型
- ☒ B、细化软件需求
- ☒ C、发现需求问题
- ☒ D、形成需求文档

软件设计基础 7

1、 软件设计制品通过评审后就不需要进行更多管理了。

- ☐ A、正确
- ☒ B、错误

1、 软件设计模型包括哪些内容。

- ☒ A、类图
- ☒ B、活动图
- ☐ C、用例图
- ☒ D、用例交换图

2、 软件设计评审一般只需要软件设计人员参加，程序员不需要参加。

- ☐ A、正确
- ☒ B、错误

3、 软件设计通常会存在哪些方面的质量问题。

- ☒ A、没有遵循和满足需求
- ☒ B、设计不够详尽，无法指导编程
- ☒ C、设计存在矛盾和不一致
- ☒ D、软件设计的模块化、重用程度不高

1、 下面哪些既是需求分析工具也是设计工具。

- ☐ A、Juint
- ☒ B、StarUML
- ☒ C、IBM Rational Rose
- ☐ D、Visual Studio

2、 . Microsoft Office和WPS可以作为软件设计CASE工具，支持软件设计文档的编写。

- ☒ A、正确
- ☐ B、错误

3、 软件设计工具通常可以提供哪些方面的支持。

- ☒ A、生成设计模型
- ☐ B、生成需求模型
- ☒ C、绘制设计模型
- ☒ D、检查设计模型的质量

1、 面向对象软件从需求模型到设计模型是精化过程，而不是转换过程。

- ☒ A、正确
- ☐ B、错误

2、 面向对象软件设计原则包括

- ☐ A、尽量开放原则
- ☒ B、接口隔离原则
- ☒ C、依赖倒置原则
- ☒ D、最少知识原则

3、 与结构化设计方法学相比较，面向对象的软件设计有何优势？

- ☐ A、技术更为简单
- ☒ B、采用精化而非转换的方法
- ☒ C、支持粗粒度的软件重用
- ☒ D、与分析采用相同的概念模型

1、 面向数据流的结构化软件设计方法在当前已经完全过时了。

☐ A、正确

☒ B、错误

2、 变换型数据流图中包括

☒ A、输入流

☒ B、输出流

☐ C、事务中心

☒ D、交换流

3、 变换型和事务型数据流图的设计过程完全相同。

☐ A、正确

☒ B、错误

1、 数据流图经过转换后，得到什么样的设计模型。

☐ A、设计类图

☐ B、流图

☒ C、层次化的软件模块图

☐ D、构件图

2、 变换型数据流图中的变换流可能有多个转换，事务型数据流图中的事务中心只对应于一个转换。

☒ A、正确

☐ B、错误

3、 结构化软件设计方法学包含有多种方法，面向数据流的软件设计方法学只是其中的一种。

☒ A、正确

☐ B、错误

1、 软件设计过程主要是形成软件设计规格说明，不需要进行管理。

☐ A、正确

☒ B、错误

2、 软件设计过程是迭代的，需要对不完善的设计制品进行不断迭代

☒ A、正确

☐ B、错误

3、 设计约束包括哪些方面

☒ A、资源约束

☐ B、领导意图

☒ C、技术制约

☐ D、功能需求

4、 软件设计工程师需要较高的技术水平和丰富经验，其余都不重要。

☐ A、正确

☒ B、错误

1、 软件设计根据实际可有可无，软件开发主要是根据需求把代码编写出来。

☐ A、正确

☒ B、错误

2、 软构件作为设计元素的主要目的是为了促进对软构件的重用。

☒ A、正确

☐ B、错误

3、 高质量的软件设计应该具有哪些特性？

☒ A、满足要求

☒ B、充分优化

☐ C、尽量使用新技术

☒ D、简单易懂

软件体系结构 8

1、 软件体系结构设计文档可作为单独的文档，加以撰写和管理。

- ☒ A、正确
☐ B、错误

2、 软件体系结构设计评审的任务是要看体系结构的设计是否满足软件需求要求。

- ☐ A、正确
☒ B、错误

3、 软件体系结构评审要关注的主要问题包括

- ☒ A、满足性
☒ B、可追踪性
☒ C、优化性
☐ D、创新性

1、 软件部署模型只需要描述系统物理节点的结构。（答案B）

- A. 正确 B. 错误
☐ A、正确
☒ B、错误

2、 软件的部署模型描述了哪些方面的信息。

- ☒ A、计算节点
☒ B、软构件
☒ C、软件制品
☒ D、节点之间的交互

3、 部署图中的计算节点既可以是计算机，也可以是诸如手机、可穿戴设备等嵌入式设备。

- ☒ A、正确
☐ B、错误

1、 软件系统的基础服务包括

- ☒ A、数据持久服务
☒ B、隐私保护服务
☒ C、安全控制服务
☒ D、消息通讯服务

2、 软件系统中的子系统、构件、设计类等设计元素是在详细设计阶段才确定的

- ☐ A、正确
☒ B、错误

3、 以下哪些元素属于设计元素。

- ☐ A、用例
☐ B、交互图
☒ C、构件
☒ D、模块

1、 重用已有软件资源就是要找到完全符合自己需求的相关制品

- ☐ A、正确
- ☒ B、错误

2、 在软件体系结构设计阶段，可重用的软件资源包括。

- ☒ A、云服务
- ☒ B、开源软件
- ☐ C、代码片段
- ☒ D、软构件

3、 可以从哪些地方寻找可重用的软件资源

- ☒ A、历史项目库
- ☒ B、开源社区
- ☒ C、网络中的云服务
- ☐ D、对其他公司制品进行破解

1、 影响软件体系结构初步设计的关键软件需求有何特点。

- ☒ A、核心、基础的需求
- ☒ B、对软件系统影响大的质量需求
- ☒ C、重要开发约束需求
- ☒ D、实现难度大、风险高的需求

2、 软件体系结构设计师应结合自己的开发经验来选择合适的软件体系结构风格，从而开展软件体系结构设计

- ☒ A、正确
- ☐ B、错误

3、 软件架构师需根据选定的软件体系结构风格，依据关键软件需求，进一步确定软件体系结构中的软件模块、每个模块的职责、模块间的逻辑关系，形成软件系统体系结构的逻辑视图。

- ☒ A、正确
- ☐ B、错误

1、 软件体系结构的设计是一个迭代的过程。

- ☒ A、正确
- ☐ B、错误

2、 软件体系结构设计过程要形成哪些制品

- ☒ A、体系结构设计文档
- ☒ B、体系结构模型
- ☐ C、详细设计文档
- ☒ D、软件部署模型

1、 类图不能用来描述软件体系结构

- ☐ A、正确
- ☒ B、错误

2、 部署图中的边包括

- ☒ A、计算节点之间的通信关联
- ☒ B、软件制品之间的依赖关系
- ☐ C、节点之间的继承关系
- ☒ D、软件制品与构件之间的依赖关系

3、 构件图中的构件接口包括

- ☒ A、供给接口
- ☐ B、实现接口
- ☒ C、需求接口
- ☐ D、转换接口

1、 对于复杂的软件系统需要设计体系结构，简单的软件就不需要了。

☐ A、正确

☒ B、错误

2、 软件体系结构就只是一个模型，不需要从多个视点或不同层次来刻画。

☐ A、正确

☒ B、错误

3、 下面属于软件体系结构风格的有哪些？

☒ A、管道/过滤器

☒ B、层次

☒ C、MVC

☐ D、适配器

1、 每个软件开发都应该从已有的体系结构风格中选择一个来直接使用，无需对体系结构风格进行必要的修改。

☐ A、正确

☒ B、错误

2、 软件体系结构在软件开发中起到承上启下、影响全局的作用，非常重要。

☒ A、正确

☐ B、错误

3、 可以用哪些图来描述软件体系结构。

☒ A、包图

☐ B、用例图

☒ C、部署图

☐ D、顺序图

1、 软件体系结构由哪些要素组成。

☒ A、构件或模块

☐ B、数据流

☒ C、连接子

☒ D、约束和限制

2、 可以从哪些视点来描述软件体系结构。

☒ A、逻辑构成视点

☒ B、物理部署视点

☒ C、程序开发视点

☒ D、代码运行视点

3、 软件非功能性需求在软件体系结构设计中发挥着关键性的作用。

☒ A、正确

☐ B、错误

软件用户界面设计 8

1、 用户界面设计的评审需要用户的深度参与。

- ☒ A、正确
- ☐ B、错误

1、 用户界面设计的输出就是最终实现的软件界面。

- ☐ A、正确
- ☒ B、错误

2、 用户界面设计的评审需要用户参与。

- ☒ A、正确
- ☐ B、错误

1、 精化用户界面设计的工作包括

- ☒ A、查漏补缺
- ☒ B、优化设计
- ☒ C、保持一致
- ☐ D、实现最终界面

1、 用户操作软件系统的主界面指用户开始使用某项用例时系统呈现出来的界面

- ☒ A、正确
- ☐ B、错误

2、 应尽量把界面元素放在一起，减少界面之间的跳转

- ☐ A、正确
- ☒ B、错误

1、 用户界面的初步设计需要设计出较为完善的用户界面原型

- ☐ A、正确
- ☒ B、错误

1、 用户界面设计的原则包括

- ☒ A、直观性
- ☒ B、易操作性
- ☐ C、个性化
- ☒ D、人性化

1、 用户界面设计模型一般用以展示原型的工具进行设计，采用类图、顺序图等进行界面设计没有必要

- ☐ A、正确
- ☒ B、错误

2、 下面属于动态界面元素的是

- ☐ A、确认按钮
- ☐ B、窗口中不变的图标
- ☒ C、随用户选择变化的文本
- ☐ D、多选框

3、 以下是word界面的一个截图，包含3个界面元素，第一个是“自动保存”的文字描述，用户点击后面二个图元可以设置自动保存模式，保存当前的文件。这三个界面元素分别属于什么形式的界面元素。



- ☒ A、静态元素、输入元素、命令元素
- ☐ B、动态元素、输入元素、输入元素
- ☐ C、静态元素、动态元素、输入元素
- ☐ D、动态元素、静态元素、输入元素

1、图形化的用户界面可以用UML的什么模型来表示。

- ☐ A、包图
- ☐ B、顺序图
- ☒ C、类图
- ☐ D、状态图

2、图形化用户界面的跳转关系可以用UML的什么图来表示。

- ☐ A、包图
- ☒ B、顺序图
- ☒ C、类图
- ☐ D、用例图

3、以下是word界面的一个截图，该图中包含哪些界面元素。

效果

- | | |
|----------------------------------|------------------------------------|
| <input type="checkbox"/> 删除线(K) | <input type="checkbox"/> 小型大写字母(M) |
| <input type="checkbox"/> 双删除线(L) | <input type="checkbox"/> 全部大写字母(A) |
| <input type="checkbox"/> 上标(P) | <input type="checkbox"/> 隐藏(H) |
| <input type="checkbox"/> 下标(B) | |

- ☐ A、动态元素
- ☒ B、静态元素
- ☒ C、输入元素
- ☐ D、命令元素

1、一个软件采用哪种人机交互方式，需要结合软件的特征和用户的需求考虑。

- ☒ A、正确
- ☐ B、错误

2、用户只和软件界面进行交互，看不到内部，所以界面设计比其他设计更重要。

- ☐ A、正确
- ☒ B、错误

3、用户也可通过语音、手势等方式也可以与软件进行交互，因此人机交互的方式多种多样。

- ☒ A、正确
- ☐ B、错误

软件详细设计 7

1、 撰写软件详细设计文档会增加项目开发工作量，影响开发进度，因此该文档应该越简单越好。

- ☐ A、正确
- ☒ B、错误

2、 下列哪些人员需要参与软件详细设计评审。

- ☒ A、用户
- ☒ B、软件架构师
- ☒ C、程序员
- ☒ D、测试人员

3、 设计评审时不仅需要关注功能性需求的实现方案，还需要关注非功能性需求的实现方案。

- ☒ A、正确
- ☐ B、错误

4、 以下属于详细设计评审时需要关注的内容的是____

- ☒ A、规范性，即文档是否遵循相应的文档规范
- ☒ B、正确性，即设计模型是否能支持所有的软件需求项
- ☒ C、充分性，即设计模型是否足够详细到能支持编码实现
- ☒ D、优化性，即设计模型是否已充分优化方式支持所有软件需求的实现

1、 如果一个软件在设计时没有划分子系统，也没有构件，则不需要开展子系统/构件设计。

- ☒ A、正确
- ☐ B、错误

2、 子系统/构件的设计与整个软件系统的设计是类似的，也包括用例设计、类设计、和数据设计。

- ☒ A、正确
- ☐ B、错误

1、 数据设计就是设计数据库的表格

- ☐ A、正确
- ☒ B、错误

2、 数据库表通常可用类图来描述，类名表示表格名称，类的属性表示表格字段，类的方法表示数据操作

- ☒ A、正确
- ☐ B、错误

3、 在确定需要永久保存的数据时，通常需要根据软件需求模型和设计模型明确系统涉及到哪些实体，以及实体之间有什么关系，从而建立系统的实体-关系模型

- ☒ A、正确
- ☐ B、错误

1、 确定数据存储和组织的方式时，不仅要关心系统需要考虑哪些数据，还需要考虑这些数据之间的关系

- ☒ A、正确
- ☐ B、错误

2、 确定数据存储和组织的方式时，对二个类之间存在1:1或者1：n的关联关系，可以使用外键的方式降低数据的冗余性

- ☒ A、正确
- ☐ B、错误

1、 类设计的任务是，对详细软件设计所产生的设计类作进一步的精化和细化，使得程序员通过类设计模型就可进行相应的编码工作

- ☒ A、正确
☐ B、错误

2、 类设计过程中需要考虑的输入包括____。

- ☒ A、软件需求模型，包括用例图、用例的交互图、分析类图等
☒ B、软件体系结构设计模型、用户界面设计模型
☒ C、用例设计阶段产生的用例实现图
☒ D、用例设计阶段产生的设计类图

3、 类设计应遵循的原则包括____。

- ☒ A、模块化和信息隐藏
☒ B、高内聚度、低耦合度
☒ C、对类的内部结构、行为等给予准确的表达，对类的接口、属性、方法等方面给予足够详细的设计
☒ D、类的内部属性、方法等设计与类的职责、关系等是相互一致的

4、 按照信息隐藏的原则，除非确有必要，否则应将类“隐藏”于包的内部，只对包中的其它类可见

- ☒ A、正确
☐ B、错误

1、 下面关于类之间关系描述正确的是____

- ☒ A、在不违背简单性、自然性原则的前提下，应尽量采用语义连接强度较小的关系
☒ B、依赖是最弱的类关系，表现为类A在某个方法中临时性使用类B
☒ C、关联表现的是两个类间存在稳定的关系
☐ D、聚合表示的是两个对象间存在一个对象撤销另一个类也消亡的部分整体关系，组合表示的是多个整体对象共享同一部件对象的部分整体关系

2、 在设计类时，一个类的方法和属性主要来源于用例实现图中对象间传递的消息及其参数

- ☒ A、正确
☐ B、错误

3、 类设计的成果包括____

- ☐ A、用例实现图
☒ B、精化后的设计类图
☒ C、描述类方法实现算法的活动图
☒ D、描述类对象状态变化的状态图

1、 下面关于用例设计的描述中正确的是____。

- ☒ A、用例设计阶段需要对需求分析阶段的每一个用例都进行设计
☒ B、用例设计的主要目标是设计每个用例的具体实现方案
☐ C、用例设计阶段用例实现图和设计类图与需求分析阶段用例交互图和分析类图是一样的
☒ D、用例设计不仅要考虑需求分析阶段的软件需求模型，还要考虑体系结构和界面设计阶段的设计成果

2、 下面关于用例交互图和用例实现图的关系描述正确的是____

- ☐ A、用例交互图和用例实现图都是描述用例的交互行为，因此其实是一样的
☒ B、用例交互图面向的是用户，用例实现图面向的是编程人员，二者针对的对象不同，因此两者的参与对象和交互行为描述方式也不一样
☒ C、用例实现图的设计必须以用例交互图为基础，不能脱离用例交互图
☒ D、用例交互图中的分析类通常都与用例实现图中的设计类存在一定的对应关系

3、 下面关于设计类图的描述正确的是____

- ☐ A、设计类图中的类与分析类图中的类应该是完全一样、一一对应的
☐ B、必须针对每个用例的实现都建立一个设计类图
☒ C、设计类图中的节点可以是子系统/构件
☒ D、根据用例实现图中相关对象之间的交互来确定设计类图中相应类之间的关系

- 1、 使用软件设计模式可以起到什么作用
- ☒ A、使软件设计更加规范化，代码编写更为工程化
 - ☒ B、提高软件开发效率，缩短软件开发周期
 - ☒ C、提高软件设计的质量
 - ☒ D、提高软件设计工程师以及程序员的逻辑思维和抽象分析能力
- 2、 下面关于单例模式的描述中正确的是
- ☒ A、单例模式适用于那些一个类只允许实例化一个对象的场景
 - ☒ B、单例模式通过将构造函数设为私有函数来避免对类进行实例化操作
 - ☐ C、单例模式提供一个全局的实例对象变量
 - ☒ D、单例模式中在不同地方调用getInstance()函数返回的都是同一个实例化对象
- 3、 下面关于观察者模式的描述中正确的是
- ☒ A、观察者模式适用于当某对象状态发生改变时，需要通知其所有依赖该对象的应用场景
 - ☐ B、在观察者模式中，所有的观察者应该都是同一类型的
 - ☐ C、在观察者模式中，需要在每个具体目标里面保存其所有的观察者
 - ☒ D、使用观察者模式可有效降低目标主题与观察者之间的耦合关系
- 1、 软件详细设计是体系结构设计 with 软件实现间的“桥梁”，相对于软件体系结构设计，它更加关注软件的“实现”，是体系结构设计得到落实的“关键”。
- ☒ A、正确
 - ☐ B、错误
- 2、 软件详细设计阶段需要针对哪些设计元素进行精化和细化。
- ☒ A、子系统
 - ☒ B、构件
 - ☒ C、关键设计类
 - ☒ D、用户界面类
- 3、 在体系结构设计阶段，每个设计元素都是一个“黑盒子”，软件架构师仅给出了这些设计元素的职责划分以及对外接口，软件详细设计则是要给出这些设计元素的内部细节信息，让每个设计元素从“黑盒子”变为“白盒子”，进而指导后续的编码实现。
- ☒ A、正确
 - ☐ B、错误
- 4、 对一个关键类的详细设计，只要给出该类包含的所有成员变量和成员方法就可以了。
- ☐ A、正确
 - ☒ B、错误

1、 软件详细设计包括____。

- ☒ A、用例设计
- ☐ B、界面设计
- ☒ C、关键类设计
- ☒ D、数据设计

2、 软件详细设计原则包括____。

- ☒ A、模块化和信息隐藏
- ☒ B、高内聚、低耦合
- ☒ C、针对软件需求
- ☒ D、深入优化设计

3、 下面对软件详细设计模型的表示描述正确的是____。

- ☒ A、用顺序图来表示用例设计
- ☒ B、用类图、状态图和活动图来表示类设计
- ☒ C、用E-R图和类图来表示数据设计
- ☒ D、用构件图来表示构件设计

4、 详细设计中“详细”是指设计要提供足够多的细节，可有效支持编码实现。

- ☒ A、正确
- ☐ B、错误

1、 详细设计是在软件体系结构设计的基础上开展，无需参考软件需求。

- ☐ A、正确
- ☒ B、错误

2、 以下关于UML活动图的描述，哪些是正确的。

- ☐ A、用于描述结构特征
- ☒ B、用于描述行为特征
- ☒ C、可刻画对象的并发和同步动作
- ☒ D、可用于刻画操作流程

3、 以下是一个活动图，该图存在哪些错误。

```
graph TD
    subgraph LoginUI
        Start(( )) --> A[接受用户输入登录的账号和密码]
        A --> B{检查输入合法性}
        B -- 不合法 --> A
        B -- 合法 --> C[请求检查用户身份的合法性]
    end
    subgraph LoginManager
        C --> D{检查输入是否为空}
        D -- 为空 --> A
        D -- 不为空 --> E[ ]
    end
    style E fill:none,stroke:none
```

- ☒ A、LoginUI、LoginManager应该有下列线以表示对象
- ☒ B、终止节点不应有外出的连线
- ☐ C、箭头符号出错
- ☐ D、泳道绘制出错

软件实现基础 5

1、 软件实现的输出包括____

- ☒ A、源程序代码
- ☒ B、可执行程序
- ☒ C、软件测试报告
- ☐ D、用户使用手册

1、 下面属于C/C++编译器的是

- ☒ A、gcc/g++
- ☒ B、cl
- ☒ C、clang
- ☐ D、javac

2、 集成开发环境（IDE）是用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面等工具。

- ☒ A、正确
- ☐ B、错误

3、 下面属于集成开发环境的是____

- ☒ A、Visual Studio
- ☒ B、Eclipse
- ☒ C、Android Studio
- ☒ D、PyCharm

1、 代码质量分为外部质量和内部质量，确保代码内部质量的有效手段是软件测试。

- ☐ A、正确
- ☒ B、错误

2、 为提高代码的可理解性和可维护性，下面描述正确的是____

- ☒ A、在编写代码时，程序员要遵循编码规范来编写代码
- ☒ B、在编写代码时，程序员要依据模块化的原则尽量把相对独立的功能封装成函数
- ☒ C、在编写代码时，程序员要尽可能地降低代码的复杂度
- ☐ D、在编写代码时，程序员要编写必要的异常定义和处理代码

3、 编码过程中程序员可能发现软件设计模型和文档中存在不合理、有问题的地方，这种情况下，程序员在编写代码的同时还要同步修改和完善相应的软件设计模型和文档。

- ☒ A、正确
- ☐ B、错误

4、 关于良好的代码编程风格，下面描述正确的是____

- ☒ A、格式化代码的布局，尽可能使其清晰、明了
- ☒ B、尽可能提供简洁的代码，不要人为地增加代码的复杂度
- ☒ C、对代码辅之以适当的文档，以加强程序的理解
- ☒ D、加强程序代码的结构化组织，提高代码的可读性

1、 下面关于程序设计语言的描述正确的是____

- ☒ A、虽然机器语言代码的执行效率比较高，但由于用机器语言编程效率低且容易出错，因此现在几乎很少有人用机器语言进行编程
- ☒ B、汇编语言代码可以直接操作计算机硬件，且占用存储空间少、执行效率高
- ☒ C、结构化程序设计语言以过程和函数作为基本模块，整个程序由一个个函数组成
- ☒ D、面向对象程序设计以类作为基本的模块单元，以对象作为程序运行的基本要素，以消息传递作为对象之间的交互手段

2、 下面属于面向对象程序设计语言的是____

- ☐ A、C
- ☒ B、Java
- ☒ C、Python
- ☒ D、C++

3、 软件开发时，程序设计语言的选择需要考虑以下哪些因素____

- ☒ A、软件的应用领域
- ☒ B、与遗留软件系统的交互
- ☒ C、软件的特殊功能及需求
- ☒ D、软件的目标平台

1、 软件实现就是程序编码。

- ☐ A、正确
- ☒ B、错误

2、 下面关于软件实现的描述正确的是____。

- ☒ A、软件实现需要考虑软件体系结构设计模型和用户界面设计模型
- ☒ B、软件实现需要考虑软件详细设计模型
- ☒ C、软件实现阶段需要开展必要的单元测试
- ☒ D、软件实现的主要目标是产生目标软件系统的高质量程序代码

3、 软件设计模型往往存在不够详细、考虑不周全的情况，因此程序员在软件实现时既要依据软件设计模型，同时也需要发挥他们的智慧和主观能动性，发现设计中存在的问题。

- ☒ A、正确
- ☐ B、错误

4、 下面哪些活动属于软件实现。

- ☒ A、编写代码
- ☒ B、单元测试
- ☒ C、代码调试
- ☒ D、软件测试

编写代码 7

1、 编写的输出包括 ____

- ☒ A、源程序代码
- ☒ B、可执行程序
- ☒ C、程序单元测试报告
- ☐ D、用户使用手册

1、 关于基于群智知识来解决编程和调试问题，下面描述正确的是 ____

- ☒ A、群智知识主要指开源技术问答社区中的互联网大众的知识
- ☒ B、典型的技术问答社区包括Stack Overflow、CSDN等
- ☒ C、程序员可以在开源社区中搜索已有的问题解决方案，也可以发布问题寻求帮助
- ☒ D、在自己难以解决身边又缺乏高手帮忙的情况下，基于群智知识是解决问题的一种有效手段

1、 关于程序调试，下面描述正确的是 ____

- ☒ A、程序调试是通过运行程序代码，确定软件缺陷的原因，定位和修复缺陷的过程
- ☒ B、对复杂的软件系统，程序员通常需要花费大量的时间和精力用于软件调试
- ☒ C、程序调试不是盲目地运行程序，程序员需要在理解缺陷及其症状的前提下，构想和假设产生缺陷的原因，然后有目标的开展调试
- ☐ D、只有通过gdb, Debugger等调试工具分步运行程序才是程序调试

1、 关于软件缺陷，下面描述正确的是 ____

- ☒ A、软件缺陷既包括程序代码中的不正确实现，也包括需求和设计中的不正确描述
- ☒ B、软件缺陷产生于软件开发全过程，只要有人介入的地方就有可能产生软件缺陷
- ☒ C、任何人都有可能软件开发过程中犯错误，进而引入软件缺陷
- ☒ D、对于大型复杂的软件系统而言，要开发出零缺陷的软件系统几乎是不可能的

2、 关于软件缺陷的描述，下面说法正确的是 ____。

- ☐ A、只有在软件公司才需要对软件缺陷进行详细的描述
- ☒ B、对发现的缺陷进行详细描述有助于缺陷的管理，进而提高软件产品的质量
- ☒ C、对发现的缺陷进行详细描述可以帮助开发人员分析、纠正和修复软件缺陷
- ☒ D、对软件缺陷的描述一般包括标识符、类型、严重程度、症状、原因等方面

3、 关于软件缺陷的应对方法，下面说法正确的是 ____。

- ☒ A、为达到预防缺陷的目的，要求开发人员对开发的每一步都要积极思考该步骤是否会产生软件缺陷，从而达到尽量少犯错误的目标
- ☒ B、由于软件缺陷不可避免，对一些核心模块，往往提供容错模块和代码来提高软件的可靠性
- ☒ C、发现缺陷不仅包括软件测试，还包括文档和模型评审、代码走查等
- ☒ D、修复软件缺陷有可能引入新的缺陷，因此缺陷修复需谨慎，且修复后需开展回归测试

4、 关于软件缺陷的状态，下面说法正确的是 ____。

- ☒ A、测试人员发现软件缺陷后，软件缺陷一开始都是处于尚未确认状态
- ☒ B、对尚未确认的缺陷进行确认后，缺陷可能变成有效、无效、或重复状态
- ☒ C、在分析软件缺陷时可能因缺陷的描述信息不完整，导致无法准确理解缺陷内容
- ☒ D、所有报告的软件缺陷最终都应处于关闭状态

1、 关于代码片段重用，下面描述正确的是 ____

- ☒ A、一个代码片段通常实现了类中的一个具体、细粒度的功能
- ☒ B、开源技术问答社区中有大量经过实践检验、具有较高质量的代码片段
- ☒ C、程序员可以自己编写精巧、优雅、高效的程序代码片段
- ☒ D、程序员可以通过重用开源社区中的高质量的代码片段来完成相应的编程任务

- 1、 下面关于基于设计编写代码的描述正确的是____
- ☒ A、基于设计编写代码指将软件设计模型映射为用程序设计语言所描述的程序代码
 - ☒ B、基于设计编写代码包括编写类代码、编写用户界面代码、编写数据设计代码
 - ☐ C、基于设计编写代码只是简单的映射，对程序员要求不高
 - ☒ D、基于设计编写代码须保证代码与设计的一致性，若设计存在问题则需修改设计

- 2、 下面属于编写类代码主要工作的是____
- ☒ A、编写类的定义代码
 - ☒ B、编写类中方法的实现代码
 - ☒ C、编写描述类之间关系的代码
 - ☒ D、编写类与类间的组织关系代码

- 3、 下面属于编写用户界面代码主要工作的是____
- ☒ A、定义界面设计元素
 - ☒ B、编写界面类代码
 - ☒ C、编写界面事件的响应处理代码
 - ☒ D、编写界面之间的跳转关系代码

- 4、 下面属于编写数据设计代码主要工作的是____
- ☒ A、在数据库管理系统中创建相应的数据库关系表格
 - ☒ B、编写与数据库建立连接的程序代码
 - ☒ C、编写操作数据库的程序代码
 - ☐ D、管理数据库管理系统中的数据

- 1、 下面关于编写代码的描述正确的是____。
- ☒ A、编写代码旨在借助于程序设计语言，编写出目标软件系统的源程序代码
 - ☒ B、程序员在编写代码过程中还需要开展程序单元测试、代码审查等质量保证工作
 - ☒ C、编写代码过程中程序员要充分发挥其创新性和主观能动性，创作出算法精巧、运行高效的代码
 - ☒ D、编写代码过程中程序员需要基于软件设计模型和文档，遵循特定的编程风格，确保其程序代码满足工程化开发要求

软件测试 6

三第1关：JUnit参数化测试

任务要求 参考答案 评论 11

```
41. this.input2 = input2;
42. this.expected = expected;
43. }
44. @Test
45. public void testAdd(){
46.     MathUtils add = new MathUtils();
47.     int result = add.add(input1, input2);
48.     Assert.assertEquals(expected, result);
49. }
50. }
51. //*****【测试执行流程】*****
52. // 首先会执行@BeforeData()方法，将准备好的数据作为一个Collection返回；
53. // 接下来根据准备好的数据调用构造方法，Collection中有几个元素，构造方法就会被调用几次；
54. // 我们这里Collection中有4个元素，所以MathUtilsTest()构造方法会被调用4次，于是会产生4个该测试
55. // 对于每一个测试类的对象，都会去执行testAdd()方法；
56. // Collection中的数据是来自于JUnit传给MathUtilsTest(int input1,int input2,int expected)构造
57. // 于是testAdd()用到的三个私有参数，就被MathUtilsTest()构造方法设置好了，而它们的三个值轮流来
58. //*****
```

编程要求

给定一个减法函数Calculator.java如下：

```
27 /**
28  *该二维数组的类型必须是Object类型的
29  *该二维数组中的每个元素都是为测试calculator中的sub()方法而准备的
30  *该二维数组中的每一个元素中的数据都对应着构造方法ParameterTest()中的参数的位置
31  *所以依据构造方法的参数位置判断，该二维数组中的第一个数减去第二个数等于第三个数
32  *请在Begin/End后补充代码，要求为单元测试传递4组参数，来验证Calculator中的sub函数编写是否正确
33  *提示：只需要补2行代码
34  */
35 /**
36  ******Begin*****
37  *
38  *Object [][] bject ={{-1,-2,1},{0,2,-2},{-1,1,-2},{1,2,-1}};
39  *return Arrays.asList(bject);
40  *
41  ******End*****
42  */
43 @Test
44 public void testSub(){
45     Calculator cal = new Calculator();
46     assertEquals(cal.sub(input11, input22), expected);
47 }
48 }
```

三第2关：JUnit异常测试

任务要求 参考答案 评论 11

- 任务描述
- 相关知识
 - JUnit异常测试
 - 代码示例
- 编程要求
- 评测说明

任务描述

学员写一个JUnit异常测试，用来判断实例化的对象数据是否合法。

相关知识

JUnit异常测试

JUnit 用代码处理提供了一个追踪异常的选项。你可以测试代码是否抛出了想要得到的异常。通过@Test元素中的expected属性验证是否抛出期望的异常，expected属性的值是一个异常的类型，如果抛出了期望的异常，则测试通过，否则不通过。

现在让我们看下@Test(expected=*.class)注册的具体使用。

代码示例

```
1 package step2;
2 import static org.junit.Assert.*;
3
4 import org.junit.Rule;
5 import org.junit.Test;
6 import org.junit.rules.ExpectedException;
7 import step2.Person;
8
9
10
11 public class JUnitException {
12
13     /**
14      *请在Begin/End内加一行注解，要求检查Person对象的年龄是否合法，不合法则
15      *抛出IllegalArgumentException异常
16      */
17     /*******Begin*****
18     @Test(expected = IllegalArgumentException.class)
19     /*******End*****
20
21     public void checkAge() {
22         Person person = new Person();
23         person.setAge(-1);
24     }
25 }
26
27
28 }
```

三第3关：JUnit套件测试

任务要求 参考答案 评论 11

- 任务描述
- 代码示例
- 编程要求
- 评测说明

任务描述

根据所学内容，要求用户补全JUnit的套件测试代码。

相关知识

####本关必读

测试套件意味着捆绑几个单元测试用例并且一起执行他们。在JUnit中，@RunWith和@Suite注册用来运行套件测试。

代码示例

对下面两个类进行单元测试，组成套件测试。

类代码：

```
1 package step3;
2
3 import static org.junit.Assert.*;
4
5 import org.junit.Test;
6 import org.junit.runner.RunWith;
7 import org.junit.runners.Suite;
8 import step3.Calculate;
9 import step3.CalculateTest;
10 import step3.Car;
11 import step3.CarTest;
12
13
14 /**
15  *请在星号后加两行注解，要求实现CalculateTest类和CarTest类的套件测试
16  *套件测试代码需要紧靠SuiteTest这个类，不能有执行
17  */
18 /*******Begin*****
19 @RunWith(Suite.class)
20 @Suite.SuiteClasses({ CalculateTest.class, CarTest.class, RectangleTest.class })
21 public class SuiteTest {
22
23
24 }
25 }
```

三第4关：命令行下进行JUnit测试

任务要求 参考答案 评论 11

- 任务描述
- 环境配置
 - 代码示例
- 本关任务
- 评测说明

任务描述

补全TestRunner.java中的代码，如果测试类JUnitSubTest.java中的测试都通过，则main函数会打印true。

环境配置

环境配置：Linux主机+JDK 1.8+JUnit 4.12。

首先确保自己在linux主机装好了java环境，配置好环境变量。

```
1 package step4;
2
3
4 import org.junit.runner.JUnitCore;
5 import org.junit.runner.Result;
6 import org.junit.runner.notification.Failure;
7
8 public class TestRunner {
9     public static void main(String[] args) {
10         //请在Begin/End内加一行代码，要求如果测试类JUnitSubTest.java中的测试都通过，则main函数会打印true
11         /*******Begin*****
12         Result result = JUnitCore.runClasses(JUnitSubTest.class);
13         /*******End*****
14         for (Failure failure : result.getFailures()) {
15             System.out.println(failure.toString());
16         }
17         System.out.println(result.wasSuccessful());
18     }
19 }
20 }
```

三第1关：第一个JUnit测试程序100

任务要求

参考答案

评论

任务描述

相关知识

编程要求

评测说明

任务描述

相关知识

JUnit编写原则

任务描述

相关知识

JUnit编写原则

```
1 package step1;
2
3 import org.junit.Test;
4 import static org.junit.Assert.assertEquals;
5 import step1.JunitSub;
6
7 public class JunitSubTest {
8     //引入JUnitSub类
9     JunitSub js = new JunitSub();
10     int sub = js.sub(5,2);
11     int testSub = 3;
12
13     /*
14     请在下面的Begin/End内写一个测试函数，
15     来验证JunitSub中的sub函数编写是否正确
16     */
17     /*****Begin*****/
18     @Test
19     public void testSub() {
20         assertEquals(sub, testSub);
21     }
22
23     /*****End*****/
24
25 }
```

三第2关：JUnit注解100

任务要求

参考答案

评论

任务描述

相关知识

编程要求

评测说明

任务描述

相关知识

JUnit注解

任务描述

相关知识

JUnit注解

```
21
22
23
24 //execute before class
25 @AfterClass
26 public static void beforeClass() {
27     System.out.println("in before class");
28 }
29
30 //execute after class
31 @BeforeClass
32 public static void afterClass() {
33     System.out.println("in after class");
34 }
35
36 //execute before test
37 @After
38 public void before() {
39     System.out.println("in before");
40 }
41
42 //execute after test
43 @Before
44 public void after() {
45     System.out.println("in after");
46 }
47
48 //test case
49 @Test
50 public void test() {
51     System.out.println("in test");
52 }
53 /*****End*****/
```

三第3关：JUnit断言100

任务要求

参考答案

评论

任务描述

相关知识

编程要求

评测说明

任务描述

相关知识

JUnit断言

任务描述

相关知识

JUnit断言

```
4 import org.junit.Test;
5
6 public class AssertionsTest {
7     String obj1 = "junit";
8     String obj2 = "junit";
9     String obj3 = "test";
10    String obj4 = "test";
11    String obj5 = null;
12    int var1 = 1;
13    int var2 = 2;
14    int[] arithmetic1 = { 1, 2, 3 };
15    int[] arithmetic2 = { 1, 2, 3 };
16
17    @Test
18    public void test() {
19
20        //请在下面的Begin/End内写添加断言测试的代码，不要改动其他地方的代码
21        /*****Begin*****/
22        assertEquals(obj1,obj2);
23        assertSame(obj3,obj4);
24        assertNotSame(obj2,obj4);
25        assertNotNull(obj1);
26        assertNull(obj5);
27        assertTrue(var1 < var2);
28        assertEquals(arithmetic1,arithmetic2);
29
30        /*****End*****/
31    }
32 }
```

三第4关：JUnit时间测试100

任务要求

参考答案

评论

任务描述

相关知识

编程要求

评测说明

任务描述

相关知识

JUnit时间测试

任务描述

相关知识

JUnit时间测试

```
1 package step4;
2
3
4 import org.junit.Test;
5
6 public class TestTimeOut {
7
8     //请在下面的Begin/End内补全test()超时测试函数，要求如果超过1000毫秒执行未结束，就判定测试未通过
9     /*****Begin*****/
10    @Test(timeout = 1000)
11    public void test() {
12        while(true){}
13    }
14    /*****End*****/
15
16 }
17
```

1、 软件测试工程师独立于软件开发工程师。

- ☒ A、 正确
☐ B、 错误

2、 软件测试组织负责制定软件测试计划。

- ☒ A、 正确
☐ B、 错误

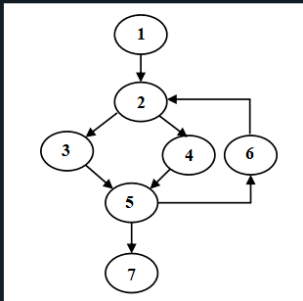
1、 白盒测试技术通常用于集成测试和确认测试阶段，黑盒测试通常用于单元测试阶段。

- ☐ A、 正确
☒ B、 错误

2、 如果某个测试技术实现路径覆盖，那么它也能实现基本路径覆盖。

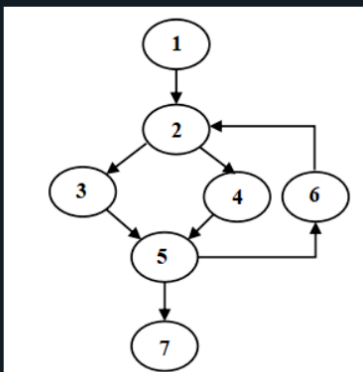
- ☒ A、 正确
☐ B、 错误

3、 以下是某个程序模块的流程图。该模块包含有多少条基本路径。其中，1是入口，7是出口。



- ☐ A、 1条
☐ B、 2条
☒ C、 3条
☐ D、 4条

1、 以下是某个程序模块的流程图。该模块包含那几条基本路径。其中，1是入口，7是出口。



- ☒ A、 1-2-3-5-7,1-2-4-5-7,1-2-4-5-6-2-3-5-7
☐ B、 1-2-3-5-7, 1-2-4-5-6-7
☐ C、 1-2-4-5-7,1-2-3-5-7
☐ D、 1-2-4-5-6-7,1-2-3-5-7

2、 如果某个模块有二个参数，参数1对应于一个集合，参数2对应于一个布尔量，那么采用等价分类法，可为该模块设计多少个测试用例。

- ☐ A、 2个
☐ B、 3个
☒ C、 4个
☐ D、 5个

3、 边界取值法是一类黑盒技术，用于支持单元测试。

- ☐ A、正确
- ☒ B、错误

1、 软件测试工作都由软件测试人员来完成。

- ☐ A、正确
- ☒ B、错误

2、 软件测试工作贯穿于软件开发全过程。

- ☒ A、正确
- ☐ B、错误

3、 在软件开发的早期阶段（需求分析和软件设计阶段），软件测试人员就可以开展软件测试工作。

- ☒ A、正确
- ☐ B、错误

1、 以下哪个软件测试的次序是正确的。

- ☐ A、系统测试、确认测试、集成测试、单元测试
- ☐ B、系统测试、集成测试、确认测试、单元测试
- ☒ C、单元测试、集成测试、确认测试、系统测试
- ☐ D、单元测试、确认测试、集成测试、系统测试

2、 α 测试、 β 测试属于确认测试。

- ☒ A、正确
- ☐ B、错误

1、 软件测试的目的是要发现和纠正代码中的错误。

- ☐ A、正确
- ☒ B、错误

2、 程序调试是软件测试的工作之一

- ☐ A、正确
- ☒ B、错误

3、 以下活动哪些需要运行程序代码。

- ☐ A、代码走查
- ☒ B、调试
- ☒ C、测试
- ☐ D、静态分析

4、 软件测试的对象是程序代码，而非软件文档。

- ☒ A、正确
- ☐ B、错误

测试结果

☒ 4 / 4 共有4题， 全部通过

1、 软件测试是如何发现软件中的缺陷的。

☐ A、程序代码能够对测试数据进行处理

☐ B、程序代码处理测试数据时不会报错

☒ C、程序处理测试数据的结果与预期结果不一致

☐ D、程序处理测试数据不会出现异常

2、 软件测试用例实际上就是一组测试数据。

☐ A、正确

☒ B、错误

3、 软件测试可以通过穷尽所有的测试用例来发现程序代码中的所有缺陷。

☐ A、正确

☒ B、错误

4、 关于测试用例的描述，以下哪些是正确的。

☐ A、测试用例就是一组测试数据

☒ B、测试用例刻画了针对特定的输入数据和前置条件，程序代码的预期运行结果

☒ C、一个程序代码的测试用例理论上有无穷多个

☒ D、测试用例必须交给待测试的代码进行处理

测试结果

4 / 4 共有4题， 全部通过

软件部署 2

1、 软件部署既包括部署和配置运行环境，也包括部署和配置软件系统。

☒ A、正确

☐ B、错误

2、 微信软件采用集中式部署方式。

☐ A、正确

☒ B、错误

3、 “12306” 软件采用分布式部署方式。

☒ A、正确

☐ B、错误

1、 软件的运行环境既可以是物理设备，也可以是软件系统。

☒ A、正确

☐ B、错误

2、 运行在Android手机上的“12306” APP，其运行环境包括哪些。

☒ A、Android

☒ B、手机

☒ C、“12306” 服务器端软件

☒ D、移动互联网

3、 部署在Android手机上的“微信” APP，其运行环境包括哪些。

☒ A、Android

☒ B、手机

☒ C、“微信” 服务器端软件

☒ D、移动互联网

软件维护与演化 3

1、 软件维护可能会产生副作用问题，导致软件质量下降。

- ☒ A、正确
☐ B、错误

2、 以下哪些维护技术不改变软件系统的功能。

- ☒ A、代码重构
☐ B、完善性维护
☒ C、设计重构
☒ D、再工程

3、 根据程序代码（包括源代码和可执行代码）得到软件的设计信息，该项工作属于逆向工程和设计重构。

- ☒ A、正确
☐ B、错误

1、 根据程序代码（包括源代码和可执行代码）和设计文档得到软件的需求信息，该项工作属于逆向工程和设计重构。

- ☐ A、正确
☒ B、错误

2、 软件再工程的目的是在不改变软件功能的前提下，得到质量更高的软件系统。

- ☒ A、正确
☐ B、错误

3、 如果软件维护过程中能够获得软件系统的设计文档，在此情况下软件维护人员需要开展哪些方面的维护工作。

- ☐ A、阅读代码以理解软件的设计信息
☒ B、阅读文档以理解软件的设计信息
☒ C、修改软件设计文档以进行软件维护
☒ D、修改程序代码以进行软件维护

1、 软件不存在物理老化的问题，但是存在逻辑老化的现象。

- ☒ A、正确
☐ B、错误

2、 软件逻辑老化具有哪些方面的表现形式。

- ☒ A、质量下降
☒ B、变更成本增加
☒ C、用户满意度降低
☒ D、维护更加困难

3、 导致逻辑老化的原因有哪些。

- ☒ A、软件外部环境的变化
☒ B、软件设计低劣
☒ C、代码质量低劣
☒ D、文档没有得到及时更新

4、 如果一个软件的价值较高，但是内部质量较低，此时该采取什么策略来解决该软件的老化问题。

- ☐ A、抛弃该软件
☐ B、不断增强该软件的功能
☒ C、对软件进行重工程，以提高软件的内部质量
☐ D、仅提供纠正性维护，不再实施完善性维护

1、 软件维护有哪些基本的形式。

- ☒ A、纠正性、完善性、适应性、预防性维护
- ☐ B、纠正性、完善性、演化性、预防性维护
- ☐ C、演化性、完善性、适应性、预防性维护
- ☐ D、纠正性、完善性、适应性、演化性维护

2、 哪种软件维护形式不改变软件系统的基本功能。

- ☐ A、完善性
- ☐ B、纠正性
- ☐ C、适应性
- ☒ D、预防性

3、 在开源软件开发过程中，某个开发者提出了某个Issue以建议增加某项功能，并为此提供代码，该项工作属于何种维护形式。

- ☐ A、预防性
- ☒ B、完善性
- ☐ C、纠正性
- ☐ D、适应性

1、 开源软件的开发过程实际上就是一个持续维护和演化的过程。

- ☒ A、正确
- ☐ B、错误

2、 在对软件进行维护和演化时，必须停止软件对外提供服务。

- ☐ A、正确
- ☒ B、错误

3、 与软件维护相比较，软件演化有何特点。

- ☒ A、软件演化功能增强的粒度大，会引起软件版本的变化
- ☒ B、软件演化持续进行，并且采用主动的方式来应对变化
- ☐ C、软件演化的持续时间长、会引起版本的变化
- ☐ D、软件演化持续进行，且持续的时间更长

软件项目管理 6

1、 软件可靠性是一类内在质量表现形式。

- ☐ A、正确
- ☒ B、错误

2、 制定标准和规范有助于提高软件制品的质量。

- ☒ A、正确
- ☐ B、错误

3、 软件质量保证活动应该贯穿于软件开发全过程。

- ☒ A、正确
- ☐ B、错误

1、 以下哪些属于软件质量保证活动。

- ☒ A、测试程序代码
- ☒ B、审查开发活动
- ☒ C、审核软件制品
- ☒ D、组织人员培训

2、 软件质量保证小组应该在软件项目实施初期就制定好软件质量保证计划。

- ☒ A、正确
- ☐ B、错误

1、 在软件开发过程中，已经发生的有害事件不能称为软件风险。

- ☒ A、正确
- ☐ B、错误

2、 软件风险具有哪些方面的特点。

- ☒ A、尚未发生但是可能会发生
- ☒ B、给软件项目开发和管理带来不利影响
- ☐ C、软件风险一定会发生
- ☒ D、事先难以确定

3、 随着软件项目实施的进展，软件风险发生的概率以及危险度会发生变化。

- ☒ A、正确
- ☐ B、错误

4、 软件风险管理包含哪些方面的工作。

- ☒ A、识别风险、分析其危险度
- ☒ B、持续观察和监控风险发生的变化
- ☒ C、化解和消除软件风险
- ☒ D、制定风险管理计划

- 1、 软件开发过程中会产生大量不同形式、不同类别的软件制品。所有这些软件制品都需要进行配置管理。
☒ A、正确
☐ B、错误
- 2、 软件文档、程序模块、测试用例甚至标准和规约都可作为软件配置项。
☒ A、正确
☐ B、错误
- 3、 软件配置管理应该贯穿于软件开发全过程。
☒ A、正确
☐ B、错误
- 4、 关于软件基线，以下哪些说法是正确的。
☒ A、软件基线是一个软件制品
☒ B、软件基线是一个软件配置项
☐ C、软件基线不可对它进行修改
☒ D、软件基线必须通过正式复审和批准

- 1、 每个软件配置项都有一个唯一的标识符。同一个软件制品（如软件需求规格说明书）可能会有多个不同版本的软件配置项。
☒ A、正确
☐ B、错误
- 2、 以下关于软件配置项之间的关系，哪些陈述是正确的。
☒ A、任何软件配置项都不应是孤立的，否则这样的软件配置项没有意义和价值
☐ B、软件项目中可能会存在孤立的、与任何其他软件配置项没有关联关系的软件配置项
☒ C、如果二个软件配置项之间存在关联关系，那么对一个软件配置项的修改可能会影响另一个软件配置项。
☐ D、软件配置项及其之间的关系可以用树来加以表示。
- 3、 可采用版本树来表示软件配置项的版本演化。
☒ A、正确
☐ B、错误

- 1、 软件项目跟踪需要关注哪些方面的内容。
☒ A、进度偏差
☒ B、资源使用情况
☐ C、软件规模
☒ D、项目实施问题和风险
- 2、 所有项目组成员都要加入到软件项目跟踪小组，并对软件项目进行跟踪。
☐ A、正确
☒ B、错误
- 3、 软件项目需要对照软件项目计划来开展。
☒ A、正确
☐ B、错误

- 1、 软件项目追踪的目的是要获得软件项目的可视性，掌握软件项目的实际状况和问题。
☒ A、正确
☐ B、错误
- 2、 应周期性或者定期地进行软件项目追踪。
☒ A、正确
☐ B、错误

1、 软件项目计划包含哪些方面的规划。

- ☒ A、进度计划
- ☐ B、质量计划
- ☒ C、资源计划
- ☒ D、活动计划

2、 软件项目计划用于指导后续的软件开发工作。

- ☒ A、正确
- ☐ B、错误

3、 可以用什么样的图来表示软件项目计划。

- ☐ A、UML图
- ☒ B、甘特图
- ☐ C、数据流程图
- ☒ D、网络图

4、 软件项目进度计划所包含的关键路径可能有多条。

- ☒ A、正确
- ☐ B、错误

1、 制定软件项目计划时要考虑哪些方面的内容和因素。

- ☒ A、项目所采用的过程模型
- ☒ B、软件项目的工作量和成本等估算
- ☒ C、项目已有的资源（如人力资源）
- ☒ D、客户或用户对项目提出的具体要求

2、 必须要基于项目的估算来制定软件项目计划。

- ☒ A、正确
- ☐ B、错误

3、 软件项目计划所涉及的所有人员都应参与软件项目计划的制定，以确保他们就计划的内容达成一致意见，并掌握和承诺软件项目计划的相关内容。

- ☒ A、正确
- ☐ B、错误

1、 任何两个软件项目都不一样，软件项目具有唯一性。

- ☒ A、正确
- ☐ B、错误

2、 软件项目管理主要包含哪些方面的管理对象。

- ☐ A、人员、制品、薪酬
- ☒ B、人员、制品、过程
- ☐ C、制品、薪酬、过程
- ☐ D、薪酬、人员、过程

3、 对各类软件模型、文档、原型、代码、数据等的管理属于软件制品管理的范畴。

- ☒ A、正确
- ☐ B、错误

1、 对人员、过程和制品等的管理均涉及风险管理的内容。

☒ A、正确

☐ B、错误

2、 关于软件度量、测量和估算的描述，哪些是正确的。

☒ A、度量是对简单属性的定量描述，测量和估算是对复杂属性的描述。

☒ B、测量是对当前状况的定量描述，估算是未来状况的定量描述。

☐ C、测量是未来状况的定量描述，估算是当前状况的定量描述。

☒ D、它们均是对软件过程、产品和资源等项目特征的定量描述。

3、 可以基于代码行数量、功能点的数量等来测量和估算软件项目的规模。

☒ A、正确

☐ B、错误

1、 在项目实施之前，可以参照历史数据、经验模型、类比等方式来估算软件项目的规模。

☒ A、正确

☐ B、错误

2、 在项目实施之前，有必要通过对软件项目进行估算，以辅助合同的签署以及软件项目计划的制定。

☒ A、正确

☐ B、错误

3、 在项目实施过程中，有必要通过对软件项目进行度量和测量，以掌握软件开发的实际情况。

☒ A、正确

☐ B、错误

4、 由于软件项目的唯一性，任何一个项目的度量、测量和估算数据对其他的项目而言没有任何参考意义和价值。

☐ A、正确

☒ B、错误