

一、 软件工程概述

一.单选题（共14题,70.0分）

【单选题】下面的（ ）是正确的。

- A、运行正确的软件就是高质量的软件
- B、软件产品质量越高越好，最理想的情况是达到“零缺陷”
- C、软件质量是由产品的功能、性能、易用性等外在特性决定的
- D、软件质量是在开发过程中逐渐构建起来的

答案：D

解析：

软件质量：软件与明确地和隐含地定义的需求相一致的程度。（本题更应该在 第9章 项目管理-质量保证）

- A、在评价软件的质量时，可靠性、可维护性、可扩展性和易用性等特点都要纳入评价。不能简单说“运行正确的软件就是高质量的软件”。
- B、最理想的情况是“以最低成本满足所有要求”。

【单选题】下面几个描述中属于软件工程方法学三要素中方法这个范畴的是（ ）。

- A、对于需求确定的软件问题，可以按照问题定义、需求分析、设计、实现、测试和部署的步骤进行顺序的软件产品开发
- B、对于跨平台的软件产品，可以采用Java语言进行开发
- C、为了有效捕获系统的功能性需求，可以采用用例的方法
- D、在面向对象的软件设计和开发过程中，可以使用Rational Rose创建和管理可视化模型

答案：C

解析：

- A、【三要素：过程】
- B、【三要素：工具】
- C、【三要素：方法】
- D、【三要素：工具】

【单选题】下面的（ ）决策是在需求分析时做出的。

- A、自动售票机系统将为使用者提供在线帮助
- B、自动售票机系统已经达到交付的要求
- C、自动售票机系统由用户界面子系统、价格计算子系统以及与中心计算机通信的网络子系统组成
- D、自动售票机系统的开发时间预计是6个月

答案：A

解析：

- A、需求分析阶段
- B、综合测试阶段
- C、总体设计阶段
- D、（其实这项也差不多在需求分析阶段进行，但是A更贴合一些）

【单选题】在软件开发过程中的每个阶段都要进行严格的（ ），以尽早发现在软件开发过程中产生的错误。

- A、度量
- B、检验
- C、验证
- D、评审

答案：D

解析：传统方法学：在每一个阶段结束之前都必须进行正式严格的技术审查和管理复审；

评审/审查：特指了软件过程中每一阶段结束前的审查工作。（详见 2 软件过程）

度量：指对软件开发过程和产品进行量化和测量的活动，例如代码行数、缺陷密度、测试覆盖率等。度量可以帮助评估软件质量，但它不是用于尽早发现错误的特定活动。（详见 9.1 软件项目管理-成本与工作量估算）

检验：指对软件工作产品进行仔细的检查，以查找错误和问题。例如代码审查。它是评审的一种实现方式。（详见 7 实现）

验证：指确认软件产品是否满足特定需求和规范的过程。它通常通过测试来进行。（详见 7.5 实现-确认测试）

【单选题】在软件开发和维护过程中需要变更需求时，为了保持软件各个配置成分的一致性，必须实施严格的（ ）。

- A、产品控制
- B、开发规范
- C、产品标准化
- D、产品检验

答案：A

解析：配置成分的一致性 ==>> 产品控制

【单选题】在以下关于软件生命周期的描述中，不正确的是（ ）。

- A、在可行性阶段主要判断客户提出的问题是否存在可行的软件求解方案，如果没有则马上终止该项目。
- B、编码后要对产品进行一系列测试，以确定软件达到了预定的要求。
- C、需求分析阶段主要解决的问题是将如何实现客户提出的需求。
- D、在编码之前要先进行软件设计，这是形成软件质量的地方。

答案：C

解析：

需求分析阶段：确定目标系统必须具备哪些功能；估计完成该项工程所需要的资源和成本，制定工程进度表。

解决“如何实现客户提出的需求”应该是总体设计阶段。

【单选题】随着开发小组人数的（ ），因交流开发进展情况和讨论遇到的问题而造成的通信开销也急剧增加。

- A、降低
- B、稳定
- C、不稳定
- D、增加

答案：D

【单选题】以下关于软件的特点的描述中，错误的是（ ）。

- A、软件是一种逻辑实体，而不是具体的物理实体，因而它具有抽象性
- B、硬件磨损后可以用备用零件替换，而软件出故障，则无法用备用零件替换来解决，是因为设计开发过程中存在错误
- C、软件存在老化和退化问题，软件老化缘于使用过久，软件退化缘于修改
- D、软件的生产与硬件不同，在它的开发中没有明显的制造过程。对软件的质量控制，必须着重在软件开发方面下功夫

答案：C

解析：软件在运行和使用期间，没有机械磨损、老化问题。软件存在退化问题。软件退化缘于修改。

【单选题】软件工程方法是（ ）。

- A、为开发软件提供技术上的解决方法
- B、为支持软件开发、维护、管理而研制的计算机程序系统
- C、为了获得高质量软件而实施的一系列活动
- D、为了理解问题和确定需求而采取的一些技术和方法

答案：A

解析：

为开发软件提供技术上的解决方法 —— 软件工程方法（软件工程方法学）

为支持软件开发、维护、管理而研制的计算机程序系统 —— 软件工程工具（软件工程方法学）

为了获得高质量软件而实施的一系列活动 —— 软件工程过程（软件工程方法学）

为了理解问题和确定需求而采取的一些技术和方法 —— 需求提取（需求分析-需求过程）

【单选题】软件会逐渐退化而不会磨损，其原因在于（ ）。

- A、不断的变更使组件接口之间引起错误
- B、软件通常暴露在恶劣的环境下
- C、软件备件很难订购

D、软件错误在经常使用之后会逐渐增加

答案：A

解析：B、C、D都是硬件的说法。软件存在退化问题。软件退化缘于修改。

【单选题】“软件工程”术语是在（ ）被首次提出。

A、美国卡内基·梅隆大学的软件工程研究所

B、IEEE的软件工程知识体系指南（SWEBOK）

C、1968年NATO会议

D、Fred Brooks的《没有银弹：软件工程中的根本和次要问题》

答案：C

【单选题】软件是计算机系统中与硬件相互依存的另一部分，它是包括程序、数据及（ ）的完整集合。

A、图表

B、代码

C、硬件

D、文档

答案：D

解析：软件包括程序、数据及其相关文档。

【单选题】下面的（ ）说法是正确的。

A、购买大多数计算机系统所需的硬件比软件更昂贵

B、由于软件是产品，因此可以应用其他工程制品所用的技术进行生产

C、大多数软件系统是不容易修改的，除非它们在设计时考虑了变化

D、一般来说，软件只有在其行为与开发者的目标一致的情况下才能成功

答案：C

解析：

A、上世纪50年代时是正确的，但之后一般来说软件更贵。

D、软件的成功并不仅仅取决于与开发者的目标一致，更重要的是与用户的期望和需求一致。

【单选题】软件开发费用只占软件生命周期全部费用的（ ）。

A、1/3

B、1/4

C、1/2

D、2/3

答案：A

二.判断题（共6题,30.0分）

【判断题】软件在运行和使用期间，由于修改会出现和硬件一样的老化问题。

答案：×

解析：软件不会老化，只会退化。

【判断题】对软件的质量控制，必须着重在软件开发方面下功夫。

答案：√

【判断题】软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。

答案：√

【判断题】软件危机的产生主要是因为软件开发人员缺少有效的高级程序设计语言。

答案：×

解析：软件危机的产生主要是在程序系统阶段，软件技术的发展不能满足需要。一方面是由于软件本身的逻辑性，和程序的复杂性、规模庞大；另一方面是由于软件开发与维护的方法不正确。

【判断题】文档是与程序开发、维护和使用有关的图文材料，是构成软件不可或缺的一部分。

答案：√

【判断题】软件既可以是一种逻辑实体，也可以是具体的物理实体。

答案：×

解析：软件是一种逻辑实体，不是物理实体。

二、软件过程

一.单选题（共17题,68.0分）

【单选题】RUP定义了初启阶段、精化阶段、构建阶段、移交阶段，每个阶段以达到某个里程碑时结束，其中（ ）的里程碑是生命周期架构。

- A、构建阶段
- B、精化阶段
- C、初启阶段
- D、移交阶段

答案：B

解析：不同阶段对应的里程碑

初启阶段 - 生命周期目标

精化阶段 - 生命周期架构

构建阶段 - 初始可操作性

移交阶段 - 产品发布

【单选题】下列关于敏捷联盟宣言的理解错误的是（ ）。

- A、个体和交互胜过过程和工具，表明团队的构建比开发环境的构建重要得多
- B、可以工作的软件胜过面面俱到的文档，表明过多的文档比过少的文档更糟糕
- C、响应变化胜过遵循计划，表明项目计划不能考虑过远，只需要为下周做详细计划即可，再远的时间可制定简略的规划

D、客户合作胜过合同谈判，表明成功的项目需要客户配合开发团队详细明确地制定合同，避免开发时关于工作内容的过多交互

答案：D

解析：敏捷过程认为规定了需求、进度和项目成本的合同在根本上是存在缺陷的，更重要的是客户在开发中充分互动和协作。

【单选题】瀑布模型是（ ）。

- A、适用于需求被清晰定义的情况
- B、一种需要快速构造可运行程序的好方法
- C、一种不适用于商业产品的创新模型
- D、目前业界最流行的过程模型

答案：A

解析：瀑布模型的缺点就是如果需求没有被很好地定义，会造成大量反馈浪费或产品和实际需求不符（因为它假设一开始需求就是定义好的）。因此特别适用于需求被清晰定义的情况。

【单选题】以下关于软件过程的说法，错误的是（ ）。

- A、软件过程可被检查、理解、控制和改进。
- B、软件过程是在软件生命周期中所实施的一系列活动的集合。
- C、软件过程可以保证各活动之间是有组织的和一致的，因此会缺乏灵活性。
- D、软件过程是指为建造高质量软件所需完成的任务的框架，它规定了完成各项任务的工作步骤。

答案：C

解析：例如敏捷过程，就是以灵活性为特点。

【单选题】极限编程采用（ ）工具来了解与需求相关的内容。

- A、用户素材
- B、访谈大纲
- C、用例图
- D、思维导图

答案：A

【单选题】以下关于极限编程（XP）的最佳实践的叙述中，不正确的是（ ）。

- A、系统最终用户代表应该全程配合XP团队
- B、只处理当前的需求，使设计保持简单
- C、编写完程序之后编写测试代码
- D、可以按日甚至按小时为客户提供可运行的版本

答案：C

解析：极限编程的每次迭代都要有对应的测试代码进行测试。

【单选题】下列关于瀑布模型的说法正确的是（ ）。

- A、实际工作中，瀑布模型仅能依次经过需求、设计、编码、测试、集成和维护这一路径进行项目开发，不能从后一阶段返回到前面的阶段。
- B、瀑布模型并非在每一阶段都要求生成相应的文档。
- C、经典瀑布模型只有在需求阶段和运行阶段才能有客户方和用户的参与，其余阶段只能开发团队参加。
- D、经典瀑布模型要求需求必须在最初阶段就要完整和明确。

答案：D

解析：

- A、通过反馈环（就是指向上一个阶段的箭头），后一个阶段可以返回到前一个阶段。
- B、瀑布模型是文档驱动模型，它关心每一阶段的文档正确性（编码阶段也会生成相关文档），例如需求文档、设计文档、测试计划等。
- C、瀑布模型并不限制客户方和用户的参与阶段。

【单选题】下列关于增量模型的说法正确的是（ ）。

- A、增量模型并不要求需求在最初阶段就是完整和明确的。
- B、增量模型能够更早地向用户交付第一个版本的系统。
- C、相比于瀑布模型，采用增量模型开发的项目所需的成本和时间要少。
- D、采用增量模型时，每一增量必须按照既定顺序依次开发，不能更改。

答案：B

解析：

- A、增量模型需要完整和明确的需求，因为增量模型要在一开始就对整体架构进行设计。
- C、开发的总成本和时间并不一定更少。增量模型的优点在于交付第一个版本的速度更快、开发的初期成本要求更少。
- D、增量模型可以根据实际情况改变开发顺序，这是它管理风险的方式。

【单选题】原型化模型是（ ）。

- A、适用于客户需求难以清楚定义的情况
- B、适用于客户需求被明确定义的情况
- C、很难产生有意义产品的一种冒险模型
- D、提供一个精确表述的形式化规格说明

答案：A

解析：原型化模型利用原型能统一客户和开发人员对软件项目需求的理解，有助于需求的定义和确认。通过快速的原型迭代，原型化模型能把需求定义的更加清晰。

【单选题】下列软件开发模型中，以面向对象的软件开发方法为基础，以用户的需求为动力，以对象来驱动模型是（ ）。

- A、瀑布模型
- B、增量模型

C、喷泉模型

D、演化模型

答案：C

解析：面向对象 ==> 喷泉模型

【单选题】增量模型是（ ）。

A、适用于需求被清晰定义的情况

B、一种需要快速构造核心产品的好方法

C、一种不适用于商业产品的创新模型

D、已不能用于现代环境的过时模型

答案：B

解析：增量模型的优点：快速交付第一个版本。

【单选题】以下关于原型化开发方法的叙述中，不正确的是（ ）。

A、原型化方法利于确认各项系统服务的可用性

B、在开发过程中，可以废弃不用早期构造的软件原型

C、原型化方法适用于需求不明确的软件开发

D、原型化方法可以直接开发出最终产品

答案：D

解析：原型化方法其实也可以直接开发出最终产品，但这是非常不推荐的，因为原型程序中可能存在非常多的缺陷。

【单选题】某大型公司计划开发一款新产品，开发周期长，但是与重要客户相关的风险也很多，最适于采用（ ）开发过程模型。

A、原型

B、螺旋

C、瀑布

D、增量

答案：B

解析：螺旋模型的重点：控制风险。

【单选题】软件工程中描述生存周期的瀑布模型一般包括计划、（ ）、设计、编码、测试、维护等几个阶段。

A、问题定义

B、需求分析

C、需求调查

D、可行性分析

答案：B

【单选题】敏捷设计要求系统设计在任何时间都尽可能的简单、干净和富有表现力，下面对敏捷设计理解错误的是（ ）。

- A、对于软件管理者来说，只需要做很少的管理工作
- B、对于软件设计者来说，通过简单、直观的划分，使其具有最小耦合的内部结构
- C、对于用户来说，通过直观、简单的界面呈现出恰当特征的程序
- D、对于开发人员来说，每周都会取得一些重大进展

答案：A

解析：不管怎样，软件管理者都是重要的角色，他不应该没事干的。

同时，D选项其实也有一点问题，每周确实都会取得进展，但不一定是“重大”进展。

【单选题】衡量原型开发人员能力的重要标准是（ ）。

- A、很强的组织协调能力
- B、快速获取需求
- C、灵活使用开发工具
- D、丰富的编程技巧

答案：B

解析：原型开发和需求分析绑定了都，它就是用来弄需求的。

【单选题】下列软件哪个最适合采用敏捷开发方法（ ）。

- A、小型创业项目软件
- B、铁路12306购票网
- C、Windows操作系统
- D、火箭飞行控制系统

答案：A

解析：敏捷开发适合小型的，快速变化的软件开发。其中，B和C都是大型项目，BCD都不需要快速变化（D根本不需要适应变化）。

二.判断题（共8题,32.0分）

【判断题】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。

答案：×

解析：“极限”的含义是指把最好的开发实践运用到极致，并不是指极限的快交付。

【判断题】V模型的本质是对瀑布模型的需求获取活动进行改造，有助于需求的定义和确认。

答案：×

解析：

V模型的本质是对瀑布模型的需求获取活动进行改造，（进一步明确了设计环节和测试环节）。

（将原型模型和瀑布模型结合），有助于需求的定义和确认。

【判断题】为了防止软件腐化，敏捷开发团队不需要一个成熟的初始设计，而是通过多次使用单元测试和验收测试，灵活、持续地改进设计。

答案：√

【判断题】使用快速原型模型收集了客户的真实需求之后，可以在这个原型基础上快速扩充得到最终的软件系统。

答案：×

解析：不能把原型开发成最终产品。

【判断题】RUP是可配置的过程，如各阶段持续时间的长短、每个阶段的迭代次数、不同迭代过程持续的时间等都是可变化和调整的因素，因此RUP具有通用性。

答案：√

【判断题】采用原型法时，关键的因素是建立模型的速度，而不是原型运行的效率。

答案：√

【判断题】软件过程有时也会被称为软件生命周期。

答案：√

【判断题】过程可以看做是一组精心安排的任务：包含有活动、约束和资源的一系列步骤，能产出某种特定输出。

答案：√

三、可行性研究

一.单选题（共19题,76.0分）

【单选题】结构化语言、判定表和判定树属于（ ）规格说明的描述工具。

- A、控制
- B、脚本
- C、加工
- D、数据描述

答案：C

解析：加工到后续开发中就成为一段程序，它的表达方式有判定表、判定树、结构化语言等。

【单选题】在结构化分析方法中，与数据流图配合使用的是（ ）。

- A、程序流程图
- B、数据字典
- C、ER图
- D、网络图

答案：B

【单选题】研发资源的有效性属于（ ）可行性的一部分。

A、社会/法律

B、技术

C、经济

D、操作

答案：B

解析：

侵权、版权、责任 ==> 法律可行性

研发、资源、技术、工具 ==> 技术可行性

成本、效益 ==> 经济可行性

操作方式、用户 ==> 操作可行性

【单选题】可行性研究的目的是（ ）。

A、判断项目是否值得开发

B、规划项目

C、开发项目

D、维护项目

答案：A

【单选题】分层数据流图是一种比较严格又易于理解的描述方式，它的顶层数据流图描述了系统的（ ）。

A、细节

B、绘制的时间

C、输入与输出

D、软件的作者

答案：C

解析：顶层数据流图只含有一个代表目标软件系统整体处理功能的转换，以及软件系统和外部环境的输入输出数据流。

【单选题】结构化分析模型中，支持数据抽象的是哪些术语（ ）。

①数据流 ②数据存储 ③数据源 ④数据汇 ⑤加工

A、①②③④

B、①②

C、③④

D、③④⑤

答案：B

解析：

①数据流：表示数据在系统中的流动或传递，可以抽象地表示数据的输入、输出、传输和转换过程。

②数据存储：表示数据在系统中的存储位置，可以抽象地表示数据的持久存储、临时存储或数据的来源和目的地。

这两个术语支持了数据的抽象，将数据视为系统中的实体，而不关注数据的具体内容或实现细节。

【单选题】结构化分析阶段，不适用于描述加工逻辑的工具是（ ）。

- A、层次图
- B、判定表
- C、结构化语言
- D、判定树

答案：A

解析：加工逻辑的表达方式有判定表、判定树、结构化语言等。

【单选题】下面关于DFD基本系统模型的描述中，错误的是（ ）

- A、基本系统模型重点刻画了当前系统与外界环境之间的关系，即数据从哪些数据源点流入，最终又流向哪些数据终点。
- B、基本系统模型中只有一个加工/处理，代表目标软件系统。
- C、对于复杂的系统常采用分层绘制DFD的方法，应该最先画出基本系统模型。
- D、从基本系统模型中可以看出目标系统的主要功能有哪些。

答案：D

解析：基本系统模型中只有一个加工/处理，代表目标软件系统，无法看出具体功能。

【单选题】结构化分析方法使用DFD描述（ ）。

- A、系统的功能
- B、系统的基本加工
- C、系统的控制流程
- D、系统的数据结构

答案：A

解析：数据流图（Data Flow Diagram）：描述数据在系统中如何被传送或变换，以及描述如何对数据流进行变换的功能。

【单选题】一个局部数据存储当它作为（ ）时就把它画出来。

- A、某个加工的特定输入
- B、某个加工的特定输出
- C、某些加工的数据接口或某个加工的特定输入/输出
- D、某些加工的数据接口

答案：C

【单选题】下面关于数据流图（DFD）基本符号元素的说明中，错误的是（ ）。

- A、加工/变换/处理是对数据进行处理单元，输入数据经过加工/变换/处理后将转换成相应的输出数据。

- B、数据流由一组数据项组成，表示运动中的数据，它可以在其它三类符号元素之间任意流动。
- C、源点和终点是系统之外的实体，可以是人、物或其他软件系统。源点和终点是为了帮助理解系统接口而引入的。
- D、数据存储/文件用来暂时存储数据的，表示处于静止状态的数据。

答案：B

解析：数据流不能在源点、终点、数据存储之间流动。

【单选题】以下说法正确的是（ ）。

- A、数据流图中的每个加工必须在数据字典的加工说明中予以定义
- B、数据流图中的图形元素个数应当控制在 $7+2$ 以内
- C、数据流图绘制时为方便起见，通常先为加工命名，再为与之相关联的数据流命名
- D、数据流图中的每个数据流和数据存储必须在数据字典中予以定义

答案：D

解析：

- A、对于一些简单的加工过程可以直接写在数据字典的条目里，而如果是一些很复杂的加工就要使用判定表等方式来进行记录。
- B、一个加工的分解子加工个数应当控制在 $7+2$ 以内。
- C、通常先为数据流命名，然后再为与之相关联的处理命名。

【单选题】在可行性研究过程中，对每一个合理的候选方案，分析人员都应准备如下资料（ ）。

- A、1、2、3全部
- B、1、系统流程
- C、3、实现该系统的进度计划
- D、2、组成系统的物理元素清单、成本—效益分析

答案：A

【单选题】对于分层的数据流图，父图与子图的平衡是指子图的输入、输出数据流同父图的输入、输出数据流（ ）。

- A、数目必须不等
- B、名字必须相同
- C、数目必须相等
- D、必须一致

答案：D

【单选题】技术可行性要解决（ ）。

- A、成本效益问题
- B、技术风险问题
- C、运行方式可行

D、存在侵权否

答案：B

【单选题】DFD的每个加工都必须至少有（ ）。

A、一个输出数据流

B、一个输入和输出数据流

C、一个输入或输出数据流

D、一个输入数据流

答案：B

【单选题】进行需求分析可使用多种工具，但（ ）是不适用的。

A、数据流图

B、数据词典

C、PAD图

D、判定表

答案：C

解析：PAD图用二维树形结构的图来表示程序的控制流，是一种过程设计工具。一般用在详细设计阶段。

【单选题】下面关于数据流图（DFD）的描述中，错误的是（ ）。

A、DFD是一种图形化技术，描绘了信息流和数据从输入移动到输出的过程中所经受的变换。

B、DFD除了刻画了系统的基本逻辑功能之外，也展现了这些功能将会如何实现的设计决策。

C、DFD描绘数据在软件中流动和被处理的逻辑过程，是系统逻辑功能的图形表示。

D、DFD简单易懂，方便分析员与用户之间的沟通。

答案：B

解析：DFD不包括具体实现，只是一个逻辑模型。

【单选题】数据词典是用来定义（ ）中的各个成分的具体含义。

A、功能结构图

B、结构图

C、数据流图

D、流程图

答案：C

二.判断题（共6题,24.0分）

【判断题】画数据流图时可以加少量的控制流，使加工之间有时序的关系。

答案：×

解析：数据流图只关心逻辑关系，因此不包括任何控制流。

【判断题】DFD中的箭头表示数据流，即特定数据的流动方向，与程序流程图中的控制流不同。

答案：√

【判断题】需求规格说明书可以作为软件可行性研究的依据。

答案：×

解析：需求规格说明书在软件开发过程中起到指导和定义需求的作用，它是根据可行性研究的结果来进行编写的，而不是作为可行性研究的依据。

也可以从生命周期的阶段顺序看出来：问题定义阶段 => 可行性研究阶段 => 需求分析阶段。需求分析是在可行性分析后面的

【判断题】功能分解级数据流图将基本系统模型中的处理进一步细化，即目标软件系统有几个主要功能，就将基本系统模型中的处理分解成几个子处理，分别对应着系统的各个主要功能。

答案：√

【判断题】在结构化分析方法中，可以使用判定表与判定树来表达系统内数据的运动情况。

答案：×

解析：判定表和判定树只能表示加工/变化内部的逻辑结构，无法表示数据流动。

【判断题】可行性研究报告应评述为了合理地达到开发目标而可能选择的各种方案，以使用户抉择。因此，编写者不必提出结论。

答案：×

解析：可行性研究的任务是为以后的行动方针提出建议。如果问题没有可行的解，应建议停止项目。如果问题值得解，应推荐一个较好的解决方案，并为项目制定一个初步的计划。

四、 软件需求

一.单选题（共15题,75.0分）

【单选题】快速原型化思想是在研究（ ）阶段的方法技术中产生的。

- A、程序编码
- B、软件设计
- C、可行性研究
- D、需求分析

答案：D

解析：快速原型化思想是在需求分析-需求提取中使用的技术。

【单选题】我们通常把确定需求，或者说确定系统应提供哪些服务以及系统运行受到哪些限制的过程及其相关的活动称之为（ ）。

- A、软件过程
- B、需求过程
- C、软件需求
- D、软件说明

答案：B

解析：我们通常把确定需求，或者说确定系统应提供哪些服务以及系统运行受到哪些限制的过程及其相关的活动称之为（ ）。

【单选题】软件需求分析阶段的工作，可以分为以下4个方面：对问题的识别、分析与综合、编写需求分析文档以及（ ）。

- A、总结
- B、需求分析评审
- C、阶段性报告
- D、以上答案都不正确

答案：B

解析：对问题的识别（需求提取）、分析与综合（需求分析与协商）、编写需求分析文档、需求分析评审（需求确认）

【单选题】传统结构化需求分析的目的是理清数据流或数据结构，导出完整的、精致的（ ）。

- A、系统物理模型
- B、数据流图
- C、系统结构图
- D、系统逻辑模型

答案：D

解析：不选数据流图，数据流图至少要和数据字典组合才能称为逻辑模型。

【单选题】在一个列车控制软件的需求文档中，我们发现了以下两条需求描述：“列车车门在两个停靠站之间要保持关闭”；“列车发生紧急停车时，要打开车门”。这里出现的需求问题是什么？

- A、无法测试的需求
- B、不完整的需求
- C、含糊的需求
- D、矛盾与不一致的需求

答案：D

【单选题】以下哪种方法最适用于身处多个不同地点的人在各自方便的时间参与并围绕同一个主题表达自己的观点？

- A、面向数据流自顶向下求精
- B、简易的应用规格说明技术
- C、问卷调查
- D、面谈

答案：C

解析：A和身处多个不同地点的人没什么关系；B是需要组织会议共同讨论的，也不行。

【单选题】软件需求分析的任务不应包括（ ）。

- A、信息域分析

- B、问题分析
- C、确定逻辑模型
- D、结构化程序设计

答案：D

解析：结构化程序设计是需求分析的下一个环节。

【单选题】以下（ ）是满足软件需求特征的非功能性需求的描述。

- A、系统提供的用户界面应该是用户友好的。
- B、系统能够对用户提供查询、修改和打印工资数据的功能。
- C、系统应该能及时返回对目标对象的准确定位。
- D、来自调度站的响应应该在1分钟内到达。

答案：D

解析：非功能性需求：可靠性、有效性、性能、可用性、健壮性、完整性。

D：性能

【单选题】在选择软件需求获取技术的时候，以下哪种策略最优？

- A、3、考虑目前系统所属的行业及应用领域的现状
- B、1、考虑尚不了解的那部分需求的特点
- C、综合考虑1、2、3因素
- D、2、考虑需求工程师本身对各种需求获取技术的驾驭能力

答案：C

【单选题】需求规格说明书在软件开发中具有重要的作用，但其作用不应当包括（ ）。

- A、用户和开发人员对软件要“做什么”的共同理解
- B、软件设计的依据
- C、软件可行性分析的依据
- D、软件验收的依据

答案：C

解析：可行性分析是需求分析的上一个环节，在需求分析前就完成了。

【单选题】下列哪项需求描述属于业务需求描述？

- A、我们公司的主营业务是销售飞机票
- B、我们的目标是让客户将我们的品牌和质量联系在一起
- C、我们的任务是无缝集成有竞争力的软件信息服务来解决商业问题
- D、公司网站上销售的产品必须满足所有食品药品监管需求

答案：A

【单选题】下列哪项不是需求规约的性质。

- A、不可修改性
- B、一致性
- C、有效性
- D、完整性

答案：A

解析：需求规约性质：一致性、完整性、现实性、有效性

【单选题】需求规格说明书的内容不应当包括（ ）。

- A、对算法的详细过程描述
- B、对重要功能的描述
- C、软件的性能
- D、用户界面及运行环境

答案：A

解析：需求不包括详细实现。

【单选题】在需求分析中，分析员要从用户那里解决的最重要的问题是（ ）。

- A、要让软件做什么
- B、要求软件工作效率如何
- C、要让软件具有什么样的结构
- D、要给该软件提供哪些信息

答案：A

【单选题】需求分析最终结果是产生（ ）。

- A、项目开发说明书
- B、需求规格说明书
- C、设计说明书
- D、可行性分析报告

答案：B

二.判断题（共5题,25.0分）

【判断题】需求分析过程是确定项目如何实现的过程，并确定项目的技术方案。

答案：×

解析：需求分析不给出具体的技术方案。

【判断题】相比硬件而言，软件更容易被修改，而且更容易被正确地进行修改。

答案：×

【判断题】需求分析的目标之一是给出解决问题的具体算法和数据结构。

答案：×

解析：需求分析不给出具体的技术方案。

【判断题】软件需求是待开发产品或系统的功能描述。

答案：×

解析：软件需求不仅有功能需求，还有非功能需求，以及对设计、制造的约束。

【判断题】非功能需求必须依附于功能需求而存在。

答案：√

五、总体设计

一.单选题（共18题,72.0分）

【单选题】下面关于面向数据流的设计方法的描述中，不正确的是（ ）。

A、任何一个信息处理系统的信息流都可以看作是一个变换流，只有当其具有明显的“事务”特征（即有一个明显的事务中心）时，才按照事务分析的映射规则进行转换。

B、面向数据流的设计方法可以利用前面需求阶段得到的数据流图，按照一定的映射规则生成相应的软件结构图。

C、数据流图有可能全局特征是变换流，而局部区域属于事务流，反之易可，所以在进行向软件结构图的转换过程中要区分全局特征和局部特征。

D、按照变换分析或事务分析从数据流图出发转换得到的软件结构图就是最终的软件总体设计结果，不需要再进行任何的调整和优化。

答案：D

解析：变换分析第七步：使用设计度量和启发式规则对第一次分割得到的软件结构进一步精化。

【单选题】模块内聚性用于衡量模块内部各成份之间彼此结合的紧密程度。一组语句在程序中多处出现，为了节省内存空间把这些语句放在一个模块中，该模块的内聚性是（ ）的。

A、逻辑内聚

B、偶然内聚

C、功能内聚

D、信息内聚

答案：B

解析：模块内各部分之间没有联系，或者即使有联系，这种联系也很松散。——偶然内聚

【单选题】为了提高模块的独立性，模块之间应该选择下列（ ）耦合度？

A、标记耦合

B、数据耦合

C、功能耦合

D、控制耦合

答案：B

注：不存在功能耦合

【单选题】关于软件设计规约，以下说法错误的是（ ）。

- A、概要设计规约对应于系统的确认测试
- B、详细设计规约主要作为软件设计人员与编程人员之间交流的媒体
- C、概要设计规约主要作为软件项目管理人员、系统分析人员与设计人员之间交流的媒体
- D、详细设计规约对应于系统的单元测试

答案：A

解析：

概要设计 - 系统测试

需求分析 - 确认测试

【单选题】（ ）着重反映的是模块间的隶属关系，即模块间的调用关系和层次关系。

- A、实体关系图
- B、系统结构图
- C、程序流程图
- D、数据流图

答案：B

【单选题】如果一个模块调用下层模块时传递一个数据结构，而被调用的模块只使用其中一部分数据，这种耦合属于（ ）。

- A、数据耦合
- B、控制耦合
- C、内容耦合
- D、标记耦合

答案：D

【单选题】软件体系结构风格代表了软件体系结构设计中的惯用模式，在下面几种体系结构风格中，（ ）支持基于抽象程度递增的系统设计。

- A、隐式调用风格
- B、黑板风格
- C、管道-过滤器风格
- D、层次系统风格

答案：D

层次系统风格优点：支持基于抽象程度递增的系统设计，使设计者可以把一个复杂系统按递增的步骤进行分解。

【单选题】（ ）是数据说明、可执行语句等程序对象的集合，它是单独命名的并可通过名字访问。

- A、模块
- B、数据块

- C、程序块
- D、复合语句

答案：A

解析：模块的定义。

【单选题】（ ）把已确定的软件需求转换成特定形式的软件表示，使其得以实现。

- A、详细设计
- B、系统设计
- C、逻辑设计
- D、软件设计

答案：D

解析：软件设计是将需求准确转换为完整的软件产品或系统的唯一办法；

需求 ==> 软件实现，这整个过程是软件设计。

【单选题】内聚性和耦合性是度量软件模块独立性的重要准则，软件设计时应力求（ ）。

- A、高内聚，高耦合
- B、高内聚，低耦合
- C、低内聚，高耦合
- D、低内聚，低耦合

答案：B

【单选题】模块内聚性用于衡量模块内部各成份之间彼此结合的紧密程度。模块中所有成分引用共同的数据，该模块的内聚性是（ ）的。

- A、过程内聚
- B、通信内聚
- C、功能内聚
- D、顺序内聚

答案：B

【单选题】以下（ ）不是总体设计环节的工作。

- A、结构设计，即确定系统由哪些模块组成，以及这些模块之间的关系
- B、过程设计，即确定每个模块的具体实现算法和使用的局部数据结构
- C、确定系统的最佳实现方案
- D、制定测试计划，在软件开发的早期阶段考虑测试问题，可以促使软件设计人员在设计时注意提高软件的可测试性

答案：B

【单选题】软件体系结构风格代表了软件体系结构设计中的惯用模式，在下面几种体系结构风格中，（ ）适合构造批处理的系统。

- A、管道/过滤器风格
- B、数据抽象与面向对象风格
- C、黑板风格
- D、客户机-服务器风格

答案：A

【单选题】管道过滤器风格的特点不包括（ ）。

- A、适合处理与用户有交互的应用
- B、该风格使得软件构件具有良好的隐蔽性和高内聚、低耦合的特点
- C、支持软件重用和构件的并行执行
- D、允许对吞吐量和死锁等系统性质的分析

答案：A

【单选题】下面关于控制耦合的描述中，不正确的是（ ）

- A、控制耦合意味着不良的模块结构，必须将其转换为数据耦合。
- B、如果两个模块间传递的信息是控制信息，即模块A通过向模块B发送一个控制变量，模块B根据该控制变量的值决定在多个功能中执行哪一个，这种情况下模块A和B之间存在控制耦合。
- C、在控制耦合下，被调用模块B其实是一个单入口多功能模块，对模块B的任何改动都会影响其调用模块A。
- D、在控制耦合下，被调用模块B本身是一个逻辑内聚模块。

答案：A

解析：控制耦合其实特别常见，不是什么很坏的结构。

【单选题】关于程序模块化的启发规则有若干条，以下规则中不符合模块化原则的是（ ）。

- A、将模块的作用范围限制在模块的控制范围之内
- B、降低上层模块的扇出，提高中层模块的扇出
- C、通过模块的合并和分解，降低模块的耦合度，提高模块的内聚性
- D、降低模块之间接口的复杂性，避免“病态连接”

答案：B

解析：模块化原则更看重模块的独立，而且一般是增加上层的扇出。

【单选题】下列对软件体系结构的描述错误的是（ ）。

- A、软件体系结构是对子系统、系统构件以及它们之间相互关系的描述。
- B、子系统和构件一般定义在不同的视图内，以显示软件系统的相关功能属性和非功能属性。
- C、软件体系结构是软件系统的一组关键设计决策。
- D、软件体系结构是软件需求活动的一种工作产品。

答案：D

【单选题】下面的（ ）不是模块化设计的目的。

- A、提高模块的可靠性和复用性
- B、清楚地描述系统的功能和性能
- C、易于维护和功能扩展
- D、降低程序设计的复杂性

答案：B

解析：模块设计的目的：降低模块之间的依赖关系，使各模块更加独立和可维护。

二.判断题 (共7题,28.0分)

【判断题】模块的控制域必须在模块的作用域内。否则，必须将判定上移或将受影响的模块下移。

答案：×

解析：模块的作用域应该在控制域之内。

【判断题】内聚标志一个模块内各个元素彼此结合的紧密程度，模块独立性由弱到强的同时模块内聚性由低到高的顺序是：功能内聚、顺序内聚、通信内聚、过程内聚、时间内聚、逻辑内聚、偶然内聚。

答案：×

解析：刚好反了。

【判断题】基于事件/隐式调用风格增加了构件之间进行数据交换的难度。

答案：√

【判断题】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。

答案：×

解析：概要设计不涉及具体算法。

【判断题】信息隐藏是指把一些关系密切的元素物理地放得彼此靠近。

答案：×

解析：

信息隐藏：模块应该设计成其中包含的信息（过程和数据）对不需要这些信息的其他模块来说是不可访问的。

局部化：指把一些关系密切的软件元素物理地放得彼此靠近。

【判断题】错误处理类模块是典型的逻辑内聚模块。

答案：√

【判断题】变换分析得到的结构图是一个三分支结构，即包括输入部分、变换中心部分和输出部分；而事务分析得到的结构图是一个二分支结构，即一个接收分支和一个发送分支。

答案：√

六、详细设计

一.单选题 (共12题,60.0分)

【单选题】设计人机交互的界面时，应当遵循一定的设计原则，不包括（ ）。

- A、操作步骤少
- B、减少人脑的记忆负担
- C、增加复杂的功能
- D、提供undo功能

答案：C

【单选题】程序片段如下。则其环形复杂度V(G)的值为（ ）。

```
Begin                                1
If  a or b                          2, 3
    then procedure  x1              4
    else procedure  y1              5
Endif                                6
If  c and d                          7, 8
    then procedure  x2              9
    else procedure  y2             10
Endif                                11
End                                  12
```

- A、5
- B、6
- C、4
- D、3

答案：A

【单选题】下面哪条属于伪码的优点？

- A、支持逐步求精
- B、不仅可以作为设计工具，还可以作为注释工具
- C、形象直观
- D、便于描述较为复杂的条件组合

答案：B

【单选题】以下说法正确的是（ ）。

- A、所有改变循环条件的成分都在循环体外
- B、在当型循环中，循环体至少要执行一次
- C、在直到型循环中，循环体至少要执行一次
- D、基本程序结构不允许嵌套

答案：C

【单选题】按McCabe程序环形复杂度计算方法，若E代表流图G的边数，n为G的节点数，则程序的环形复杂度V(G) = （ ）。

- A、E - n + 2
- B、E - n + 1
- C、E - n
- D、n

答案：A

【单选题】软件详细设计的主要任务是确定每个模块的（ ）。

- A、程序
- B、功能
- C、算法和使用的数据结构

D、外部接口

答案：C

【单选题】下面描述中，符合结构化程序设计风格的是（ ）。

- A、使用顺序、选择和重复（循环）三种基本控制结构表示程序的控制逻辑
- B、不使用goto语句
- C、模块只有一个入口，可以有多个出口
- D、注重提高程序的执行效率

答案：A

【单选题】不属于详细设计工具的是（ ）。

- A、PDL
- B、PAD图
- C、N-S图
- D、DFD

答案：D

【单选题】程序控制分为（ ）三种基本结构。

- A、顺序、嵌套、选择
- B、顺序、选择、循环
- C、分块、顺序、循环
- D、递归、堆栈、队列

答案：B

【单选题】软件详细设计的图形工具中，（ ）可以纵横延伸，图形的空间效果好。

- A、流程图
- B、PAD图
- C、HIPO图
- D、N-S图

答案：B

【单选题】下面哪条属于程序流程图的优点？

- A、便于表达数据结构
- B、描绘直观，便于掌握
- C、支持逐步求精
- D、程序员可以很早地开始考虑程序的控制流程

答案：B

【单选题】人机界面设计是软件设计工作的一项重要内容，下面各项中（ ）不属于良好的人机界面设计原则。

- A、出错信息应该使用面向用户的术语并为用户提供有助于从错误中恢复的建设性意见
- B、减少用户的记忆负担
- C、系统的响应时间越短越好
- D、帮助系统应该为用户提供多个可能的入口

答案：C

二.判断题（共8题,40.0分）

【判断题】详细设计评审应尽可能和概要设计评审一同进行。

答案：×

【判断题】在用McCabe方法计算程序的环形复杂度时，嵌套IF语句与简单CASE语句的复杂性是一样的。

答案：√

【判断题】一个程序流程图对应的盒图表示是唯一的。

答案：×

【判断题】详细设计的目的是要把系统中的模块组织成良好的层次结构。

答案：×

【判断题】用某一种软件复杂性来度量不同类型的程序，所得到的度量值不一定真正反映它们的复杂性。

答案：√

【判断题】使用PAD符号所设计出来的程序必然是结构化程序。

答案：√

【判断题】PDL可以作为注释直接插在源程序中间，这样做能促使维护人员在修改程序代码的同时也相应地修改PDL注释，因此有助于保持文档和程序的一致性，提高了文档的质量。

答案：√

【判断题】在软件设计规约中，应当有针对出错的处理。

答案：√

七、 软件实现

一.单选题（共16题,64.0分）

【单选题】在测试一个长度为48000条指令的程序时，第一个月由甲、乙两名测试员各自独立测试这个程序。经过一个月测试后，甲发现并改正20个错误，使MTTF达到8h。与此同时，乙发现24个错误，其中的6个甲也发现了。以后由甲一人继续测试这个程序。估算刚开始测试时程序中总共有（ ）个潜藏的错误？

- A、 100
- B、 60
- C、 80
- D、 120

答案：C

【单选题】等价划分法的一个假设是，每类中的一个（ ）在测试中的作用与这一类中所有其他值的作用相同。

- A、 最大值
- B、 最小值
- C、 边界值
- D、 典型值

答案：D

【单选题】逻辑覆盖标准主要用于白盒测试。它主要包括语句覆盖、判定覆盖、条件覆盖、条件/判定覆盖和条件组合覆盖等几种。其中，最弱的覆盖标准是（ ）。

- A、 语句覆盖
- B、 判定覆盖
- C、 条件组合覆盖
- D、 条件/判定覆盖

答案：A

【单选题】下列叙述中，符合程序设计风格指导原则的5条叙述是（ ）。

- ① 嵌套的重数应加以限制。
- ② 尽量多使用临时变量。
- ③ 不滥用语言特色。
- ④ 不用可以省略的括号。
- ⑤ 使用有意义的变量名。
- ⑥ 应尽可能把程序编得短些。
- ⑦ 把常见的局部优化工作留给编译程序去做。
- ⑧ 注解越少越好。
- ⑨ 程序的格式应有助于读者理解程序。
- ⑩ 应尽可能多用GOTO语句。

A、②③⑤⑥⑦

B、①③⑤⑦⑨

C、①④⑤⑧⑨

D、②④⑤⑦⑨

答案：B

【单选题】如果在某班级管理系统中，班级的班委有班长、副班长、学习委员和生活委员，且学生年龄在15~25岁。若用等价类划分来进行相关测试，则（ ）不是好的测试用例。

A、（班长，20）

B、（班长，15）

C、（队长，15）

D、（队长，12）

答案：D

【单选题】使用等价类划分法时，完全不考虑软件的（ ）。

A、内部结构

B、外部环境

C、顺序

D、业务流程

答案：A

【单选题】为了提高易读性，源程序内部应加功能性注释，用于说明（ ）。

A、数据的用途

B、程序段或语句的功能

C、模块参数的用途

D、模块总的功能

答案：B

【单选题】给定C语言程序如下：

```
int foo(int x, int y, int d) {  
    if (x!=0) {  
        if (y==0)  
            d=d/x;  
        else  
            d=d/(x*y);  
    } else {  
        if (y==0)  
            d=0;  
        else  
            d=d/y;  
    }  
    return d;  
}
```

当用路径覆盖法进行测试时，至少需要设计（ ）个测试用例。

- A、 3
- B、 5
- C、 4
- D、 8

答案： C

【单选题】下列不属于软件配置复查目的的是（ ）。

- A、 确保文档与程序完全一致
- B、 保证软件配置的所有成分都齐全
- C、 确保软件具有完成测试所必须的细节
- D、 确保软件质量符合要求

答案： C

【单选题】下面的（ ）是错误的。

- A、 在程序设计中应尽可能对程序代码进行优化
- B、 在程序设计中使⽤括号以改善表达式的清晰性
- C、 不要修补不好的程序，要重新写
- D、 不要在注释中重复描述代码

答案： A

【单选题】单元测试的重点包括模块接口、（ ）、重要的执行通路、出错处理通路和边界条件。

- A、 不正确或遗漏的功能
- B、 外部信息访问
- C、 初始化或终止性错误
- D、 局部数据结构

答案： D

【单选题】黑盒测试注重于测试软件的功能性需求，主要用于软件的后期测试。（ ）不能用黑盒测试检查出来。

- A、 程序控制结构错误
- B、 功能不对或遗漏错误
- C、 界面错误
- D、 外部数据库访问错误

答案： A

【单选题】在改正当前故障的同时可能会引入新的故障，这时需要进行（ ）。

- A、 性能测试
- B、 验收测试
- C、 功能测试
- D、 回归测试

答案： D

【单选题】某城市的电话号码由3部分组成。这3部分的名称与内容分别是：地区码：空白或3位数字；前缀：非'0'或'1'开头的3位数字；后缀：4位数字。使用等价类划分法进行黑盒测试，如下测试用例中不合理的是（ ）。

- A、 (635) 805-9321
- B、 (20A) 223-4567
- C、 (3B4) 043-12345
- D、 (7777) 345-6789

答案： C

【单选题】下面的（ ）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

答案：D

【单选题】关于代码性能优化，下面（ ）是错误的。

- A、应先通过测试找出限制效率的真正瓶颈
- B、应以提高程序的全局效率为主，局部效率为辅
- C、要优先改进耗时最多的部分
- D、任何优化都不能破坏代码的正确性

答案：C

二.判断题（共9题,36.0分）

【判断题】可以用基于流图的环形复杂度描述测试一个单元或构件所需的工作量。

答案：√

【判断题】等价类划分法的主要思想是首先将所有的输入数据划分成若干个有效等价类，然后设计一个新的测试用例，使其仅覆盖一个尚未被覆盖的等价类，重复这一步，直到所有的等价类都被覆盖为止。

答案：×

【判断题】白盒测试又叫做功能测试或数据驱动测试，黑盒测试又称为结构测试或逻辑驱动测试。

答案：×

【判断题】软件测试不仅能表明软件中存在错误，也能说明软件中不存在错误。

答案：×

【判断题】集成测试是测试和组装软件的系统化技术，进行集成测试时可以采用非渐增式测试和渐增式测试，其中渐增式测试实质上是同时完成单元测试和集成测试。

答案：√

【判断题】判定覆盖法则一定可以查出在判定中某些条件中关系运算符使用有误时产生的错误。

答案：×

【判断题】程序的McCabe环形复杂度决定了程序中独立路径的数量，而且这个数是确保程序中所有语句至少被执行一次所需的最少测试用例数量。

答案：×

【判断题】软件测试的目的是尽可能地发现程序中的错误和缺陷，详细严密的测试过程可以证明软件的正确性。

答案：×

【判断题】测试分析报告应把每个模块实际测试的结果，与软件需求规格说明书和概要设计说明书中规定的要求进行对照并作出结论。

答案：×

八、 软件维护

一.单选题（共11题,72.6分）

【单选题】以下关于文档的叙述中，不正确的是（ ）。

- A、文档也是软件产品的一部分，没有文档的软件就不能称之为软件。
- B、软件文档的编制在软件开发工作中占有突出的地位和相当大的工作量。
- C、文档仅仅描述和规定了软件的使用范围及相关的操作命令。
- D、高质量文档对于发挥软件产品的效益有着重要的意义。

答案：C

【单选题】在软件使用过程中，为了解决原来程序中遗漏处理文件中最后一个记录的问题而对软件进行的修改，称为（ ）。

- A、适应性维护
- B、改正性维护
- C、完善性维护
- D、预防性维护

答案：B

【单选题】为了提高软件的可靠性、可维护性等，为以后进一步改进软件打下良好基础而对软件进行的修改，称为（ ）。

- A、预防性维护
- B、改正性维护
- C、完善性维护
- D、适应性维护

答案：A

【单选题】（ ）是从现有的程序代码中抽取有关数据、体系结构和处理过程的设计信息，以便恢复设计结果。

- A、正向工程
- B、数据重构
- C、逆向工程
- D、代码重构

答案：C

【单选题】下面的（ ）不是软件再工程活动。

- A、程序结构改善
- B、增加新的功能
- C、数据再工程
- D、逆向工程

答案：B

【单选题】符合下列5个叙述的软件特性名称及顺序是（ ）。

- ①软件从一个计算机系统或环境转移到另一个计算系统或环境的容易程度。
 - ②软件在需要它投入使用时能实现其指定的功能的概率。
 - ③软件使不同的系统约束条件和用户需求得到满足的容易程度。
 - ④在规定的条件下和规定的一段期间内，实现所指定的功能的概率。
 - ⑤尽管有不合法的输入，软件仍能继续正常工作的能力。
- A、可移植性 可使用性 兼容性 可靠性 容错性
 - B、可移植性 可靠性 可修改性 可使用性 容错性
 - C、一致性 可靠性 可移植性 可使用性 可修改性
 - D、可测试性 可靠性 可理解性 可修改性 兼容性

答案：A

【单选题】在软件使用过程中如果因为安装了新的硬件或操作系统而需要对软件进行的修改，称为（ ）。

- A、完善性维护
- B、适应性维护
- C、预防性维护
- D、改正性维护

答案：B

【单选题】以下关于提高软件可维护性的措施中，错误的是（ ）。

- A、尽可能利用硬件特点以提高程序效率。
- B、在分析用户需求时同时考虑维护问题。
- C、不要修补不好的程序，要重新编写。也不要一味地追求代码的复用，要重新组织。
- D、把与硬件及操作系统有关的代码放到某些特定的程序模块中。

答案：A

【单选题】以下关于软件可维护性的叙述中不正确的是（ ）。

- A、可维护性可以从可理解性、可靠性、可测试性、可重用性、可移植性等方面进行度量。
- B、可维护性不受软件开发文档的影响。
- C、可维护性是软件开发阶段各个时期的关键目标。
- D、可维护性是衡量软件质量的一个重要特性。

答案：B

【单选题】某搜索引擎在使用过程中，若要增加接受语音输入的功能，使得用户可以通过语音输入来进行搜索，此时应对系统进行（ ）维护。

- A、完善性维护
- B、改正性维护
- C、适应性维护
- D、预防性维护

答案：A

【单选题】下面关于软件维护的描述中，错误的是（ ）。

- A、在软件产品刚刚进入使用阶段的初期，改正性维护的要求比较多。
- B、一般情况下，在几种维护活动中，完善性维护所占的比重最大，即大部分维护工作是改变和加强软件，而不是纠错。
- C、软件维护大多是救火式的紧急维修，一少部分可以有计划的一种再开发活动。
- D、软件维护活动所花费的工作占整个软件生存期工作量的70%以上，因此在软件开发的各个阶段都要努力提高软件的可维护性。

答案：C

二.判断题（共4题,27.4分）

【判断题】提高可维护性是在软件维护阶段考虑的问题。

答案：×

【判断题】为了提高软件日后的可用性和可维护性，软件组织应该为自己的所有遗留软件进行软件再工程。

答案：×

【判断题】代码重构是最常见的再工程活动，关注个体模块中存在的问题并完善软件的整体架构。

答案：×

【判断题】数据重构发生在较低的抽象层次上，是一种全范围的再工程活动。

答案：√

九、软件项目管理

一.单选题 (共12题,60.0分)

【单选题】假设一个人单独开发软件，生产率是5000行/人年。若4个人组成一个小组共同开发这个软件，在每条通信路径上耗费的工作量是250行/人年。则组中每人的生产率降低为（ ）。

- A、4200
- B、4625
- C、4350
- D、4375

答案：B

【单选题】在软件开发的各种资源中，（ ）是最重要的资源。

- A、硬件环境
- B、开发工具
- C、人员
- D、方法

答案：C

【单选题】软件项目估算时，若计划人员对每个功能分别按最佳的、可能的、悲观的三种情况给出功能点估计值，记作a, m, b，则估算的期望值E的公式为（ ）。

- A、 $E = (2a+3m+4b)/9$
- B、 $E = (a+m+b)/3$
- C、 $E = (a+4m+b)/6$
- D、 $E = (a+2m+b)/4$

答案：C

【单选题】配置管理贯穿软件开发的整个过程。以下内容中，不属于配置管理的是（ ）。

- A、版本控制
- B、风险管理
- C、变更管理
- D、配置状态报告

答案：B

【单选题】下面的（ ）是有效的软件配置项。

①文档 ②可执行程序 ③测试数据

- A、①文档
- B、①②③
- C、②可执行程序
- D、①②
- E、③测试数据

答案：B

【单选题】民主式结构团队的特点是（ ）。

- A、以主程序员为核心，团队其他人员的职能进行专业化分工。
- B、技术经理负责技术决策，项目经理负责非技术性事务的管理决策和绩效评价。
- C、开发人员以志愿者形式参加，每个人参与自己感兴趣的项目，完全无人管理。
- D、团队成员完全平等，享有充分的民主，成员之间通过协商做出决策。

答案：D

【单选题】软件配置管理的目的是（ ）。

- A、提高正确率
- B、提高软件开发效率
- C、减少混乱
- D、控制软件修改
- E、降低开发成本

答案：D

【单选题】以下关于进度管理工具Gantt图的叙述中，不正确的是（ ）。

- A、不能清晰地确定任务之间的依赖关系。
- B、能清晰地表达每个任务的开始时间、结束时间和持续时间。
- C、能清晰地确定影响进度的关键任务。
- D、能清晰地表达任务之间的并行关系。

答案：C

【单选题】关于工作分解结构，下列（ ）说法是错误的。

- A、工作分解结构的最高层，是项目的可交付成果
- B、工作分解结构不包括的工作就不在项目的范围之内
- C、工作分解结构表现形式可以是图表也可以是文字大纲
- D、工作分解结构是自顶向下逐层构建的

答案：A

【单选题】以下（ ）最不适宜采用民主制程序员组的组织形式。

- A、确定性较小的项目
- B、采用新技术的项目
- C、大规模项目
- D、开发人数少（如3~4人）的项目

答案：C

【单选题】以下关于关键路径的叙述中，错误的是（ ）。

- A、关键路径是一个工程网络图中最长（指持续时间最长，而非经过的活动最多）的路径。
- B、由最早时刻和最迟时刻都相同的事件构成的路径就是关键路径。
- C、现实中关键路径上每个活动的空闲时间不一定为零，但该路径的空闲时间总和最小。
- D、使用关键路径法可以得出完成项目所需的最少时间。

答案：B

【单选题】在软件配置管理中，基线的目的是（ ）。

- A、保证配置项的依赖性
- B、合理分配存取权限
- C、合理控制变更
- D、保证配置项的完整与正确

答案：C

二.判断题（共8题,40.0分）

【判断题】软件开发工作量是软件规模（KLOC或FP）的函数，其单位通常是人月（PM）。

答案：√

【判断题】对于相同的代码行数或功能点数，使用不同模型估算将得到不同的结果。主要原因是模型多数都是仅根据若干应用领域中有限个项目的经验数据推导出来的，适用范围有限。

答案：√

【判断题】如何组织项目组是一个重要的管理问题，对于大项目或是周期固定、较短的项目，适合采用集中式的人员组织方式。

答案：√

【判断题】估算软件规模有代码行技术和功能点技术两种方法，其中代码行方法估计实现一个功能所需要的源程序行数，其不依赖项目所使用的开发语言，并且具有较好的直观性。

答案：×

【判断题】配置管理在软件项目启动时就开始，一直持续到软件退役后才终止。

答案：√

【判断题】采用的编码方法和语言在表达和解决问题方面效率越高，用代码行方法计算的生产率就越高。

答案：×

【判断题】一个工程网络图中的关键路径有且只能有一条。

答案：×

【判断题】在工程网络图中，如果在由起始事件至终止事件所构成的一条路径上，所有事件的最早时刻和最迟时刻都相同，那么这条路径就是关键路径。

答案：×

十、面向对象建模基础

一.单选题（共10题,50.0分）

【单选题】在使用用例进行需求建模时，可以将常规动作放在一个用例中，将非常规动作放在另一个用例中，那么这两个用例之间的关系是（ ）。

- A、关联关系
- B、扩展关系
- C、包含关系
- D、泛化关系

答案：B

【单选题】在使用用例描述一个图书租售系统的需求时，以下（ ）不能作为参与者。

- A、张三
- B、顾客
- C、信用卡支付系统
- D、图书管理员

答案：A

【单选题】UML的设计视图包含了类、接口和协作。其中，设计视图的静态方面由类图和对象图表现，而动态方面由交互图、（ ）表现。

- A、类图和顺序图
- B、对象图 and 状态图
- C、状态图 and 活动图
- D、状态图 and 类图

答案：C

【单选题】UML中关联的多重度是指（ ）。

- A、一个类的实例能够与另一个类的多少个实例相链接
- B、一个类的某个方法被另一个类调用的次数
- C、一个类中被另一个类调用的方法的个数

D、两个类所具有的相同的方法和属性

答案：A

【单选题】以下关于分析类的叙述中，错误的是（ ）。

A、按照职责可以将分析类划分为：实体类、边界类和控制类。

B、“分析类”相当粗略，并且不要往其中添加技术细节。

C、边界类描述系统外部环境与内部运作之间的交互，为此要确定系统对用户的响应或消息，并要对界面的可视方面建模。

D、“分析类”是概念层的内容，从它们可捕获系统对象模型的雏形。

答案：C

【单选题】UML中的4种基本关系包括（ ）。

A、包含关系、扩展关系、泛化关系、关联关系

B、继承关系、多态关系、重载关系、关联关系

C、关联关系、依赖关系、泛化关系、实现关系

D、调用关系、泛化关系、实现关系、重载关系

答案：C

【单选题】当一个对象具有特征（ ）时，应该将其建模为属性，而不是类。

A、除了非常简单的取 / 赋值操作（set，get等），不具备更多其他行为。

B、有可能被多类对象共享或传递。

C、具有比较复杂的自身行为。

D、具有独立标识。

答案：A

【单选题】在一个“订单输入子系统”中，创建新订单和更新订单都必须核查用户帐号是否正确。那么，用例“创建新订单”、“更新订单”与用例“核查用户帐号”之间是（ ）关系。

A、泛化关系

B、等价关系

C、扩展关系

D、包含关系

答案：D

【单选题】在进行面向对象分析的过程中可以把“分析类”划分成几类，其中用（ ）描述系统必须存储的信息以及与这些信息直接相关的操作。

A、实体类

B、控制类

C、边界类

D、参与类

答案：A

【单选题】在UML中，（ ）用于描述系统与外部系统及用户之间的交互。

A、类图

B、用例图

C、对象图

D、协作图

答案：B

二.判断题（共10题,50.0分）

【判断题】在UML用例图中，参与者之间可以有泛化关系，表示子参与者的实例可以与其父参与者对应的用例实例进行交互。

答案：√

【判断题】用例是捕获需求的好方法，对于功能性需求和非功能性需求都可以准确获取和刻画。

答案：×

【判断题】聚合关系（aggregation）是一种特殊的组合关系(composition)，聚合关系具有很强的归属性，而且整体与部分的对象生存周期是必须一致的。

答案：×

【判断题】部署图是描述一个系统运行时的硬件节点、在这些节点上运行的软件构件将在何处物理运行以及它们将如何彼此通信的动态视图。

答案：×

【判断题】在面向对象分析与设计中，控制类是应用领域中的核心类，一般是系统内对象和系统外参与者的联系媒介；边界类用于保存系统中的信息以及提供针对这些信息的相关处理行为；实体类主要是协调上述两种类或对象之间的交互。

答案：×

【判断题】一个类与其自身不可以存在关联关系，两个不同的类之间也不可以存在多条同向的关联。

答案：×

【判断题】在面向对象分析中仅仅关注把概念组织成分析类之间的泛化关系，而不能用复用的形式来解释。

答案：√

【判断题】现拟要求开发一个企业信息管理系统，要求与原来已有的一商品库存系统相连接。在用用例图刻画该目标系统功能时，商品库存系统应该表示成为一个用例。

答案：×

【判断题】从用例图中的每一对参与者和用例，可以确定一个边界类，从每一个用例可以确定一个（或多个）控制类。

答案：√

【判断题】状态图详细说明了由事件序列引起的状态序列。在面向对象分析中要针对每个类创建状态图，以刻画每一个类的动态行为特性。

答案：×
