

软件工程-同等学力计算机综合真题及答案(回忆版)
(2018)

2018 年软工

第三部分 软件工程

(共 30 分)

一、单项选择题(每小题 1 分,共 5 分)

1. E-R 模型中包含的基本成分是()

- 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、协作

S 答: C E-R 模型中包含的基本成分是实体、联系、属性

来源: 2008 年上半年信息系统监理师上午真题

2. 正式技术评审的目的是()

- A. 发现软件中的错误 B. 评价程序员的工作效率
C. 发现和改正程序中的错误 D. 记录程序员给出错情况与绩效挂钩

S 答: C 正式技术评审目的: (1) 发现软件在功能、逻辑、实现上的错误; (2) 验证软件符合它的需求规格; (3) 确认软件符合预先定义的开发规范和标准; (4) 保证软件在统一的模式下进行开发; (5) 便于项目管理。此外, 正式技术评审为新手提供软件分析、设计和实现的培训途经, 后备、后续开发人员也可通过正式技术评审熟悉他人开发的软件。

3. 在 UML 中,() 关系描述了两个对象类之间的一般化/特殊性关系, 它可以使子对象类共享父对象类的属性和方法。

- A. 依赖 B. 泛化 C. 关联 D. 实现

S 答: B

4. 某模块内有两个处理 A 和 B, 分别对数据区 X 写数据和读数据, 则该模块的内聚类型属于()

- A. 逻辑内聚 B. 过程内聚 C. 通信内聚 D. 内容内聚

S 答: C 通信内聚-模块内各个组成部分都使用相同的数据或产生相同的数据结构。

来源: 2016 年下半年软件设计师考试第 36 题

5. 关于增量开发模型的叙述, 错误的是()

- A. 不必等到整个系统开发完成就可以使用
B. 可以使用较早开发的增量构件来构建稍后开发的增量构件
C. 优先级最高的服务先交付, 这样最重要的服务能接受最多的测试
D. 有利于较好的模块划分

S 答: D 来源: 2016 年 5 月软件设计师考试第 29 题

二、判断题(每小题 1 分,共 5 分。如果正确,用“√”表示,否则,用“×”表示)

1. 软件的开发成本不但需要考虑开发的人力消耗, 还要考虑期间的其他经常性消耗。()

S 答: √ 直接成本、间接成本、人力成本、非人力成本都要考虑

2. 模块的独立程度可以使用两个标准来衡量。这两个标准分别是模块的内聚度和模块之间的耦合度。它们属于定性的标准。()

S 答: √ 模块的独立程度可以由两个定性标准度量, 这两个标准分别称为内聚和耦合

3. 描述一个模块内的处理流程时, 一种改进的方法是使用 N-S 图(盒图), 与最常用的程序流程图相比, 它的优点是完全避免了 GOTO 转移, 彻底遵循了结构化程序设计思想。()

S 答: × 程序流程图属于结构化程序设计工具的一种, 它也限制使用 goto 语句。

注: 结构化程序设计思想包括: 自顶向下、逐步求精、模块化、限制使用 goto 语句。

主张使用顺序、选择、循环三种基本结构来嵌套连结成具有复杂层次的“结构化程序”, 严格控制 GOTO 语句的使用。

为了提高程序的清晰性和可靠性, 结构化程序设计中限制使用 goto 语句。结构化程序设计常用三种工具: 图形、表格和语言。

图形: 程序流程图、N-S 图、PAD 图 表格: 判定表 语言: 过程设计语言(PDL)

4. 描述一个模块内的处理流程时, 一种改进的方法是使用 PAD 图(盒图), 与最常用的程序流程图相比, 它的优点是完全避免了 GOTO 转移, 彻底遵循了结构化程序设计思想。()

S 答: × 同上

5. 在描述系统功能时常使用用例建模，但也需要辅之以规格说明，即用例实现的场景。场景从用户角度描述每一个功能处理的事件序列。()

S 答：× 场景从用户角度描述每一个功能处理的事件如何执行。

注：业务用例、业务用例场景与业务用例实现之间的关系

所谓的业务用例场景即为了实现业务用例所采取的不同的实现方式或做法，因此一个业务用例才有多个业务用例场景的出现。而业务用例场景和业务用例实现的关系通常为一个业务用例场景对应一个业务用例实现。业务用例场景规定了业务如何执行，而业务用例实现则描述如何通过计算机来实现这个业务。

举一个简单的例子供大家参考：假如用户想缴纳手机话费，我们可以吧缴纳手机话费当做一个业务用例，众所周知，缴纳话费可以去营业厅中缴纳，也可以网上缴费(如通过微信，支付宝)，这两种方式都能达到缴纳手机费用的业务目标。因此缴纳手机话费这个业务用例就存在两种不同的业务用例场景，一个是通过营业厅的营业员代理缴纳，一个是自己通过互联网缴费，它们均对应一个业务用例实现，而这两种业务用例场景的最终实现都需要依托计算机，也即计算机如何实现这个业务，也即我们上面提到的业务用例实现。

三、简答题（每小题 4 分，共 12 分）

1. 在承包软件项目之前为什么要需要可行性研究？软件项目的可行性研究主要研究哪几个方面的可行性？

S 答：可行性研究的目的，就是用最小的代价在尽可能短的时间内确定问题是否能够解决。它压缩简化了系统分析和设计的过程，也就是在较高层次上以较抽象的方式进行系统分析的过程。如果在项目开始进行前，没有进行可行性研究，当开发时遇到问题且该问题根本得不到解决，那么之前在这项工程上所有的花费都只能是浪费。

可行性研究最根本的任务是对以后的行动方针提出建议，如果问题没有可行的解，分析员应该建议停止这项开发工程，避免时间资源人力金钱的浪费，如果问题值得去解决，那么应该推荐一个较好的解决方案，并且为工程制定一个初步的计划。

(1)经济可行性：进行成本/效益分析。从经济角度判断系统开发是否“合算”。

(2)技术可行性：进行技术风险评价。从开发者的技术实力、以往的工作基础、问题的复杂性等出发，判断系统开发在时间、费用等限制条件下成功的可能性。

(3)法律可行性：确定系统开发可能导致的任何侵权、妨碍和责任。

(4)方案的选择：评价系统或产品开发的几个可能的候选方案，最后给出结论意见

2. 事件驱动风格的体系结构在软件体系结构分类中属于控制模型，它是通过外部生成的事件来驱动的系统。典型的事件驱动风格的体系结构有哪两种类型？简述它们的控制机制。

S 答：

事件驱动模式的基本原理是构件并不直接调用过程，而是触发一个或多个事件。系统中的其他构件可以注册相关的事件，触发一个事件时，系统会自动调用注册了该事件的构件过程，即触发事件会导致另一构件中过程的调用。

事件驱动模式的主要特点是事件的触发者并不知道哪些构件会受到事件的影响,且不能假定构件的处理顺序，甚至不知道会调用哪些过程，使用事件驱动模式的典型系统包括各种图形界面工具。

特征：系统中有许多目标，利用消息协作的方式来工作。主子系统负责整个系统的运行，每个元素拥有事件接收和处理机制

优点：易于完成并发任务，良好扩展性，良好交互性

缺点：对系统计算控制能力弱，难以共享数据，对象间逻辑复杂

举例：各种 IDE；MVC 框架；EJB 的 JMS 和 MessageDriven Bean

两种类型：

1) MVC 全名是 Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC 被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

2) 集成开发环境 (IDE, Integrated Development Environment) 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。集成了代码编写功能、分析功能、编译功能、调试功能等一体化的开发软件服务套。所有具备这一特性的软件或者软件套 (组) 都可以叫集成开发环境。如微软的 Visual Studio 系列，Borland 的 C++ Builder、Delphi 系列等。

该程序可以独立运行，也可以和其它程序并用。IDE 多被用于开发 HTML 应用软件。例如，许多人在设计网站时使用 IDE (如 HomeSite、DreamWeaver 等)，因为很多项任务会自动生成。

注：

体系结构的风格有哪些？

- ．管道-过滤器风格
- ．面向对象风格
- ．事件驱动风格
- ．分层风格
- ．数据共享风格
- ．解释器风格
- ．反馈控制环风格

常见架构风格举例总结

1. 管道-过滤器风格

特征：每个功能构件都有一组输入输出，过滤器独立完成自身的功能，相互之间无须进行状态交互；各过滤器无须知道输入管道与输出管道所连接的过滤器的存在，仅仅保证自己的输入输出，不必理会其余过滤器。

优点：减少构件之间的耦合；容易维护和扩展；易于分析。

缺点：缺乏交互性。

举例：Linux Shell 的管线；数字通信系统。

2. 面向对象风格

特征：封装，继承，多态

优点：高模块化、代码封装、代码共享、易维护、良好扩展性

缺点：调用者必须知道被调对象的引用，当对象引用改变时，必须通知所有可能调用其的对象，系统内高耦合

举例：Java，C#开发的系统

3. 事件驱动风格

特征：系统中有许多目标，利用消息协作的方式来工作。主子系统负责整个系统的运行，每个元素拥有事件接收和处理机制

优点：易于完成并发任务，良好扩展性，良好交互性

缺点：对系统计算控制能力弱，难以共享数据，对象间逻辑复杂

举例：各种 IDE；MVC 框架；EJB 的 JMS 和 MessageDriven Bean

4. 分层风格：

特征：每层只为上一层服务，同时接收来自下层的服务。

优点：抽象化，软件重用，易扩展

缺点：方法间接调用，影响性能

举例：TCP/IP 协议；J2EE 的表现层、领域层、数据控制层、数据库。

5. 数据共享风格（库风格）

特征：中央的数据单元被共享，为一些单元提供存储和访问服务

优点：知识库扩展性，解决具体领域的问题

缺点：

举例：专家系统、自然语言处理、模式识别

6. 解释器风格

特征：固定的结构、伪代码、解释器引擎

优点：

缺点：

举例：Java 虚拟机

7. 反馈控制环风格

特征：通过其学习构件和决策者构件的运用，能利用学习和信息更新增强自身的功能

优点：

缺点：

举例：数据挖掘、机器学习系统

8. 两层 C/S 风格

特征：服务器管理应用程序，客户机运行应用程序。现已比较少用。

9. 三层 C/S 风格

特征：客户端运行表现层。服务器运行功能层和数据层

优点：BS 所缺乏的

缺点：客户端难以升级维护

举例：QQ，MSN，需安装的网游

10. B/S 风格

特征：客户端采用浏览器。

优点：容易维护升级

缺点：交互性较差，响应较慢

举例：各种网站

3. 软件生命周期中可能执行的活动可分为 5 个基本过程，这 5 个基本过程是什么？每一个基本过程与软件项目的哪一方相关？

S 答：GB/T 8566-2007《信息技术 软件生存周期过程》标准把软件生存周期中可能执行的活动分为 5 个基本过程、9 个支持过程和 7 个组织过程，每一生存周期过程划分为一组活动，每一活动进一步划分为任务。

5 个基本过程：

- 1、获取过程：为需方而定义的活动，启动，招标，合同，对供方监督，验收等
- 2、供应过程：为供方而定义的活动，启动，准备投标，签订合同，编制计划，执行，交付和完成
- 3、开发过程：为开发方而定义的活动：需求、设计、编码、测试、安装、验收
- 4、运作过程：为操作方而定义的活动：运行测试，系统运行，用户支持
- 5、维护过程：为维护方而定义的活动：问题和修改分析，修改实现，维护评审/验收，迁移，软件退役

四、建模题（共 8 分）

问题陈述：下面是在某慕课教育平台上一个在线作业批改系统的简化陈述：

◆ 学生

- 系统登录，提交学生标识（学号、密码）；
- 在线提交作业（作业题量、完成题目、答案、提交时间）；
- 获取作业已批改的通知（学号、批改时间、评分、评价（可选））；
- 查看已批改的作业。

◆ 教师

- 教师登录，提交教师标识（教师号、密码）；
- 上传作业信息（作业题量、布置题目、标准答案、评分标准、最后期限）；
- 获取有作业提交的通知（学号、提交时间）；
- 下载学生提交的作业（在屏幕上显示）；
- 批改后上传已批改作业（学号、批改时间、评分、评价（可选））；
- 获取作业抽检意见（学号、教师号、建议）。

◆ 教学督导

- 教师登录，提交督导标识（教师号，密码）；
- 抽取并下载作业样本（学号、教师号、批改时间、评分、评价（可选））；
- 抽检并提交抽检意见（学号、教师号、建议）。

试回答：

- 1.（4 分）结构化分析方法给出该系统的顶层 DFD 中，请补充图中缺失的 4 个数据流；
- 2.（4 分）写出该数据流图中的数据字典。