

# 软件工程

(课程代码 02333)

## 注意事项:

1. 本试卷分为两部分, 第一部分为选择题, 第二部分为非选择题。
2. 应考者必须按试题顺序在答题卡(纸)指定位置上作答, 答在试卷上无效。
3. 涂写部分、画图部分必须使用 2B 铅笔, 书写部分必须使用黑色字迹签字笔。

## 第一部分 选择题

一、单项选择题: 本大题共 15 小题, 每小题 2 分, 共 30 分。在每小题列出的备选项中只有一项是最符合题目要求的, 请将其选出。

1. 在 UML 中, 用于描述系统的功能需求, 并通过场景来展示系统功能的使用方式的图是  
A. 类图  
B. 用例图  
C. 序列图  
D. 组件图
2. 以下在单元测试之后进行的测试层次是  
A. 系统测试  
B. 验收测试  
C. 集成测试  
D. 回归测试
3. 软件开发的本质在于通过抽象和建模来解决问题, 最准确地描述这一过程的是  
A. 软件开发是将现实世界的问题转化为计算机程序的过程  
B. 软件开发是编写代码以实现算法的过程  
C. 软件开发是使用设计模式来提高代码质量的过程  
D. 软件开发是优化硬件资源以提升软件性能的过程
4. 在软件建模中, 模型的作用是  
A. 模型是软件开发的最终产品, 用户可以直接使用  
B. 模型是对现实世界的简化表示, 用于分析和设计软件系统  
C. 模型是用于测试软件系统性能的工具  
D. 模型是用于编写软件文档的框架

5. 需求规格说明书(SRS)中的非功能需求通常描述了软件的  
A. 界面设计和用户体验  
B. 性能、可靠性和安全性  
C. 功能模块和操作流程  
D. 开发工具和环境要求
6. 在结构化需求分析中, 以下哪个步骤是用于识别和定义系统必须处理的数据对象?  
A. 数据流分析  
B. 数据字典创建  
C. 数据建模  
D. 功能分解
7. 在结构化设计中, 下列选项不是模块划分的标准的是  
A. 功能性  
B. 数据流  
C. 控制流  
D. 时间复杂度
8. 关于结构化设计中的“自顶向下”设计方法, 描述正确的是  
A. 从系统的整体目标开始, 逐步分解到具体的模块  
B. 从系统的细节开始设计, 逐步构建到整个系统  
C. 首先设计系统的数据结构, 然后设计功能模块  
D. 首先设计用户界面, 然后设计系统的内部逻辑
9. UML 中的哪个概念元素用于描述对象可能具有的状态序列以及引起状态转移的事件?  
A. 用例图  
B. 序列图  
C. 状态图  
D. 活动图
10. 以下关于 UML 用例图的描述, 正确的是  
A. 用例图主要用于描述系统的静态结构  
B. 用例图中, 参与者(Actor)表示系统中的一个用例  
C. 用例图中, 用例(Use Case)表示系统的一个功能或行为  
D. 用例图不能表示用例之间的关系
11. 关于 UML 中的依赖关系, 描述正确的是  
A. 依赖关系表示两个类之间的继承关系  
B. 依赖关系表示一个类是另一个类的成员  
C. 依赖关系表示两个类在运行时具有动态关联  
D. 依赖关系表示一个类的实现依赖于另一个类的接口
12. 以下关于 RUP(Rational Unified Process)的描述, 正确的是  
A. RUP 是一个瀑布模型的过程  
B. RUP 是一个线性的软件开发过程  
C. RUP 是一个迭代的、增量的软件开发过程  
D. RUP 不支持需求变更管理

13. 以下关于软件生存周期过程的描述, 正确的是
- A. 软件生存周期是指软件产品从概念到废弃的整个时期
  - B. 软件生存周期只包括编码和测试两个阶段
  - C. 软件生存周期过程是指软件从设计到部署的时期
  - D. 软件生存周期过程不包括维护阶段
14. 增量模型适用于
- A. 需求非常明确的项目
  - B. 项目需求可能会频繁变化
  - C. 项目有严格的截止日期
  - D. 项目团队规模较小
15. 在 CMMI 的组织成熟度级别中, 以下哪个级别表明过程已量化管理并持续改进?
- A. 初始级
  - B. 已管理级
  - C. 已定义级
  - D. 持续优化级

## 第二部分 非选择题

二、填空题: 本大题共 10 小题, 共 20 个空, 每空 1 分, 共 20 分。

16. 增量模型的特点是软件被划分为一系列的\_\_\_\_\_, 每个增量都是可交付的, 并且每个增量都添加了更多的功能, 直到软件完全完成; 增量模型的突出优点是第一个版本交付所需要时间和成本\_\_\_\_\_。
17. 所谓成熟度等级, 是指达到预先定义的一组过程域所有目标的一种\_\_\_\_\_等级。在 CMMI 中, 应用于一个组织过程改善的成熟度等级有\_\_\_\_\_个。
18. 单元测试一般采用\_\_\_\_\_技术。单元测试通常考虑模块的以下 4 个特征: 模块接口、\_\_\_\_\_、重要的执行路径、错误执行路径。
19. 等价类划分方法是指把软件所有可能的输入数据, 即软件的\_\_\_\_\_划分成若干部分, 形成一个等价类, 即在一个部分中各个输入数据对于发现软件中错误的概率是一样的, 然后从每一部分中选取数据作为\_\_\_\_\_, 进行软件测试。
20. 软件评估可分为\_\_\_\_\_评估与动态评估。评审、走查和形式化证明等是静态评估技术, 而\_\_\_\_\_是一种常用的动态评估技术, 通过执行程序发现其中的错误。
21. 在 UML 状态图中, 事件有四种类型, 分别为: 信号事件、调用事件、\_\_\_\_\_、变化事件; 信号事件是一种\_\_\_\_\_事件, 调用事件通常是一个同步事件。
22. 为了表示事物之间的各类关系, UML 给出了 4 个术语, 分别是: 关联、\_\_\_\_\_、\_\_\_\_\_、细化。
23. 在计算机软件领域中, 一般把数据定义为对客观事物的一种表示; \_\_\_\_\_定义为具有特定语义的数据。\_\_\_\_\_是指数据的流动。

24. 模块是执行一个特殊任务的一个过程以及相关的\_\_\_\_\_。模块通常由两部分组成, 一部分是接口, 给出可由其他模块或例程访问的常量、变量、函数等; 模块的另一部分是\_\_\_\_\_, 是接口的实现。
25. 在 RUP 中, 规定了四个开发阶段: 初始阶段、\_\_\_\_\_、构造阶段和移交阶段。在 RUP 的每次迭代中都要经历一个核心 workflow, 即需求、分析、设计、\_\_\_\_\_和测试。

三、简答题: 本大题共 6 小题, 每小题 5 分, 共 30 分。

26. 简述演化模型的主要特征。
27. 简述语句覆盖、分支覆盖、条件组合覆盖、路径覆盖的含义及它们之间的关系。
28. 简述 RUP 的核心工作中, 需求获取的用况模型与需求分析模型的主要异同点。
29. 为了表达客观世界中的各类事务, UML 给出了 8 个术语, 请给出这 8 个术语, 并简述其基本含义。
30. 简述模块耦合的主要类型。
31. 简述软件危机的基本含义。

四、综合应用题: 本大题共 2 小题, 每小题 10 分, 共 20 分。

32. 等价类划分是一种测试设计技术, 它将输入数据的集合划分为若干个等价类, 从每个等价类中选取代表性的值作为测试用例。以下为一个简单的 Java 方法, 该方法用于检查用户输入的密码是否有效。

```
public class PasswordValidator {
    public static boolean isValidPassword(String password) {
        if (password == null || password.length() < 8 || password.length() > 12) {
            return false;
        }
        boolean hasUpperCase = false;
        boolean hasLowerCase = false;
        boolean hasDigit = false;

        for (int i = 0; i < password.length(); i++) {
            char ch = password.charAt(i);
            if (Character.isUpperCase(ch)) {
                hasUpperCase = true;
            } else if (Character.isLowerCase(ch)) {
```

```
        hasLowerCase = true;
    } else if (Character.isDigit(ch)) {
        hasDigit = true;
    }
}

return hasUpperCase && hasLowerCase && hasDigit;
}
}
```

请用文字说明密码有效的规则是什么？然后给出密码无效类说明及对应的测试用例设计。

33. 以下是一个简单的算法，用于从数组中找出最大值，使用伪代码描述：

算法名称: FindMaxValue

输入: 一个数组 arr, 其长度为 n ( $n > 0$ )

输出: 数组中的最大值 max\_value

伪代码:

- 1) . 初始化 max\_value 为 arr[0] // (a)
- 2) . 对于 i 从 1 到 n-1 做循环: // (b)
  - a. 如果 arr[i] > max\_value, 则 // (c)
    - i. 将 max\_value 设置为 arr[i] // (d)
- 3) . 结束循环
- 4) . 返回 max\_value // 返回找到的最大值。

请在伪代码中的位置 (a)、(b)、(c)、(d) 处给出相应伪代码的注释；并用一个程序流程图来描述该伪代码描述的算法。

