

软件测试课程期末考试（复习大纲）

1 简答题（20 分，每题 1 分）

1 什么是软件测试？

软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计的一批测试用例，并利用这些测试用例运行程序以及发现错误的过程

2 软件测试具有哪些特性？

普遍性，不完全性，证错不证对，周期性，综合性，可拓展性

3 软件测试的对象包括哪些方面？

需求规格文档、概要设计规格说明、详细设计规格说明，源程序

4 软件测试的目标是什么？

以最少的时间和人力，系统地找出软件系统中潜在的各种错误和缺陷

5 写出你认为软件测试应该遵循的最重要的几条原则。

1.软件测试应追溯到用户需求。2.软件测试尽早开始，不断进行。3. 应彻底检查每个测试结果

6 软件测试方法可以分为哪几类方法？

1.软件测试的基本方法 2. 针对不同开发阶段的 3. 针对软件不同特性和方面的 4. 针对不同开发方式和应用场景的 5. 特殊的

7 简单描述软件测试的信息流模型，指出在该模型中最关键的因素是什么？

测试输入，得到测试结果，和预期结果比对并分析，错误则改正，回归测试，输出修正结果，正确则进行可靠性分析。**最关键因素：预期结果**

8 软件开发有哪些模型，软件测试分别处于什么位置？

大棒开发法，构建-修复模型(测试编码长期循环往复)，瀑布模型(在开发阶段)，快速原型模型(测试没有明确位置)，螺旋开发(每一阶段)

9 软件测试过程有哪些模型？这些模型各有什么特点？

V 模型：自顶向下开发，自底向上测试

W 模型：尽早测试、不断测试，文档是静态测试，软件是动态测试

X 模型：对单独程序片段相互分离的编码和测试，多次集成交接后再进行测试

H 模型：将测试活动完全独立

10 作为一种职业，你认为软件测试员应该具备哪些专业基础？

要系统地掌握软件测试的基本知识，善于不断尝试新的测试方法和测试工具(过硬的技术+丰富的经验)

11 在软件测试领域，最根本（关键）的问题在哪里？

1.测试由谁来执行？2.测试什么？3.什么时候进行测试？4. 怎样进行测试？

11 软件测试工具大致可以分为几种？

1. 测试管理工具 2. 自动化功能测试工具 3. 性能测试工具 4. 单元测试工具 5. 白盒测试工具 6. 测试用例设计工具

12 软件测试管理具有什么作用？

对软件测试全过程控制，对软件质量全方位管理，建立多层次的软件开发、管理体系

13 简述软件测试发展的六个发展阶段。

面向调试-面向演示-面向破坏-面向评价-面向预防-职业、教育和研究逐渐兴起

14 缺陷管理具体包括哪些功能？

缺陷预防 缺陷发现 缺陷记录和报告 缺陷分类和跟踪 缺陷处理 缺陷预测

15 什么是软件？

软件 = 程序 + 文档

16 什么是软件质量？

软件产品满足规定和隐含需求能力有关的所有特征和所有特性的总和

17 什么是软件缺陷？

程序员犯错，变为程序中的故障或缺陷，执行程序时导致了和预期行为之间的差异，出现软件失效

18 软件测试发展的未来趋势是什么？

软件测试自动化 测试技术智能化 测试方法工具化 测试工具服务化 测试理论系统化
测试手段多样化 测试过程标准化 测试人员专业化 测试部门独立化 测试管理全面化 测试对象精细化

19 软件测试有哪些停止准则？

超时,无措，通过特定测试，正面指出，出错速率

20 解释软件的故障模型 PIE。

Execution 执行：错误代码必须要执行到。

Infection 感染：触发了错误的中间状态。

Propagation 传播：错误的中间状态必须可以传播到最后的输出使其可以被观测

2 名词解释，并将以下名词术语相应编号填入表中进行分类，给出白盒动态测试技术之间的强弱关系，给出黑盒测试和白盒测试综合的测试策略。（10 分）

	1 黑盒测试	2 白盒测试
3 静态测试	5	6789
4 动态测试	19-24	10-18

1 黑盒测试：不考虑程序内部结构和处理过程的前提下，对程序接口进行的测试

2 白盒测试：允许利用程序内部的逻辑结构及有关信息，设计或选择测试用例

3 静态测试：不运行待测软件，不生成测试用例，按照一定步骤直接检查源代码或软件规格说明书

4 动态测试：生成测试用例，运行待测软件，根据结构元素覆盖、功能正确完整等准则进行测试

5 文档审查：对软件规格说明书进行阅读和检查

6 同行评审：好友评审，不正式，互相查看代码

7 桌面评审：单人进行代码检查，自己阅读检查自己的程序，对照错误检查

8 代码走查：程序员逐行逐功能阅读、解释代码，评审组听取并提问

9 代码审查：编写的程序员不能参与，专业的参与人员从不同角度进行评审、记录

10 语句覆盖：选择若干测试用例，使得每个语句至少执行一次

11 判定覆盖：使得每个判定的取真分支和取假 分支至少执行一次

12 条件覆盖：使得每个判定的条件变量的可能取值至少执行一 次

13 判定/条件覆盖：上面的组合

14 条件组合覆盖：使得每个判定中各个条件变量的所有取值组合 至少执行一次

15 修改决策条件覆盖：使得每个条件变量独立改变判定语句的真假值

16 路径覆盖：覆盖程序中所有可能的路径

17 LCSAJ 覆盖：覆盖每个线性代码序列跳转

18 数据流测试（定义/引用对覆盖）：通过追踪变量在程序中的定义-使用链，检测数据依赖缺陷

19 等价类划分：从每个等价类中只取一组数据作为代表性数 据用于测试

20 边界值分析：选取恰好等于、刚刚 大于或刚刚小于边界的值作为测试数据

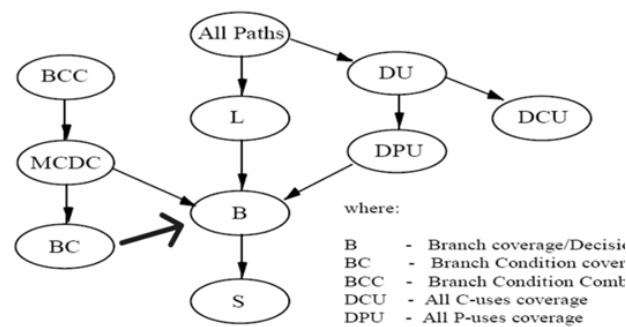
21 因果图与决策表：转换为因果图，得到条件和输入之间的依赖关系，得到条件和输入之间的依赖关系

22 错误猜测：列举出程序中可能有的错误和容易发生错误的特殊情况

23 状态转换测试：抽象为一个状态转换图，根据状态转换图设计测试用例，每个状态和转换都至少执行一次

24 语法测试：基于输入接口的语法变异生成测试用例

白盒动态测试技术之间的强弱关系：



黑盒测试和白盒测试综合的测试策略：有机结合黑盒测试方法，先进行静态白盒测试，再进行动态白盒测试，或两者结合，在黑盒测试基础上增加白盒测试。

3 简述软件开发过程中以下测试方法的特点

3.1 单元测试：验证行为，设计行为，编写文档的行为，具有回归性

3.2 集成测试：是单元到系统的过渡性测试，效果依赖于集成策略

3.3 系统测试：将整个软件系统视为一个整体来测试，测试各项功能是否满足用户要求

3.4 验收测试：保证软件满足所有功能要求，达到每个性能要求和其他预定要求等

3.5 冒烟测试：测试周期短，响应速度快，最小化集成风险，简化故障诊断

3.6 回归测试：验证已变更系统的完整性和正确性

3.7 α 测试：用户在受控制开发环境下进行的验收测试

3.8 β 测试：用户在开发者无法控制的实际使用环境下的验收测试

4 简述以下软件特性或方面的测试特点（what, why, how,...）（30 分，每题 1 分）

4.1 负载测试：让系统在压力极限范围内或临界状态连续运行，测试系统的稳定性

4.2 压力测试：持续不断给被测系统增加压力，直到被测系统被压垮，从而确定系统能承受的最大压力

4.3 性能测试：对软件的运行性能指标进行测试，判断系统集成后在实际使用环境下能否稳定、可靠运行

4.4 可靠性测试：通过系统性的测试方法评估软件的可靠性程度

4.5 容量测试：确定测试对象在给定时间内能够持续处理的最大负载或工作量

4.6 安全性测试：验证该软件系统中危险防止和处理设施的安全控制机制是否有效

4.7 安装测试：确保软件在正常和异常情况下都能成功安装的测试

4.8 可用性测试：评估软件的易用程度，确保用户使用时有满意的体验

4.9 稳定性测试：检测软件在使用过程中是否稳定，是否会崩溃

4.10 本地化和国际化测试：检测软件是否针对不同国家或地区进行适配和调整

4.11 可访问性测试：检测软件是否满足可访问性规范，为各类用户扫除使用障碍

- 4.12 授权测试：测试授权过程健壮性的方法
- 4.13 容错性测试：检查软件在异常条件下自身是否具有防护性的措施或某种灾难性恢复的手段
- 4.14 一致性测试：测试一个产品在效率或互通性是否符合某个指定的标准
- 4.15 配置测试：检测软件在不同构建和运行配置下是否能正常工作
- 4.16 文档测试：针对软件开发、维护的全生命周期中文档展开测试，验证用户文档的正确性
- 4.17 兼容性测试：测试被测软件和其他软件之间能否正确交互和信息共享
- 4.18 试玩：为发现游戏中潜在的错误邀请玩家试玩进行测试
- 4.19 可恢复性测试：检测软件系统从软件或硬件失效中恢复的能力
- 4.20 卸载测试：关注软件能否卸载，卸载是否干净
- 4.21 能力测试：逐条检查目标文档，判断程序是否满足每条语句定义的能力
- 4.22 健壮性测试：验证系统面对各种异常条件、错误输入时的稳定性和正确性
- 4.23 穿越测试：故意违反各种安全约束，验证和检查系统的鲁棒性，并评估系统的安全性
- 4.24 在线帮助测试：验证系统的实时在线帮助的可用性和正确性
- 4.25 数据转换测试：验证已有的数据转换是否有效
- 4.26 备份测试：验证系统在软件或硬件失效的事件中备份数据的能力
- 4.27 接口测试：对软件需求规格说明中接口需求逐项进行的测试
- 4.28 人机交互界面测试：对人机交互界面提供的操作和显示界面进行测试，检验是否满足用户需求
- 4.29 余量测试：检查软件是否达到需求规格说明中要求的余量的测试，通常留有 20%
- 4.30 协议测试：检查协议标准的正确性和有效性
- 4.31 内存泄漏测试：检测内存是否被正确释放，是否被过度消耗导致系统崩溃
- 4.32 存储测试：关注数据存储功能，确保数据能够正确写入、存储和读取
- 4.33 软件老化：长时间运行错误积累，资源消耗导致的性能下降，错误率上升
- 4.34 不稳定性测试：检测软件的偶发性错误，难以复现的错误行为

5 简答题（核心检测能力是什么？或者说特点是什么？）（30 分，每题 1 分）

- 5.1 组合测试：充分考虑各种因素及其相互作用的测试方法，直接检测各种影响因素及其组合对软件产生的影响
- 5.2 蜕变测试：不再是输出和预期输出对比来判断，而是一组输出之间的关系来判断
- 5.3 基于规格说明的软件测试：验证软件实现与规格说明的一致性
- 5.4 基于模型的软件测试：利用模型来描述系统的行为、特性和需求，设计测试用例
- 5.5 基于错误的软件测试：针对待测软件中可能存在的某种软件错误，设计测试用例
- 5.6 基于搜索的软件测试：利用元启发式搜索技术自动生成测试数据
- 5.7 统计测试：根据预定义的概率分布来随机选择测试用例
- 5.8 基于操作剖面的测试：按照人们对软件各项功能的使用频率来选择测试用例
- 5.9 变异测试：少有的主动评估测试用例有效性的测试
- 5.10 脆弱性测试：相同测试环境和对象下，测试结果时而成功时而失败，检测这一类偶发性错误
- 5.11 基于性质的软件测试方法：根据系统或程序的性质来生成测试用例
- 5.12 极限测试：先创建单元测试和验收测试，再创建代码库的测试形式
- 5.13 模糊测试：提供非预期的输入并监视异常结果来发现软件故障的方法
- 5.14 软件测试的控制论方法-自适应测试：是通过控制理论指导软件测试

- 5.15 导向性随机测试：将具体执行与符号执行相结合，通过生成测试输入来执行程序中的所有可行路径
- 5.16 图形用户界面测试：测试软件产品的图形用户界面，确保 GUI 符合用户要求或规格说明书
- 5.17 随机测试：在所有可能输入中，随机选择和产生测试输入，对软件进行测试
- 5.18 自适应随机测试：强调随机选择测试用例的过程中，测试用例之间尽可能均匀分散开
- 5.19 反随机测试：使测试用例最大限度的分散，尽可能发现更多的软件故障
- 5.20 结对测试：两个测试人员结对在一起测试软件的方法
- 5.21 在线测试：测试执行被结合到算法中，在运行过程中动态产生测试用例
- 5.22 探索性测试：测试工程师通过测试不断学习系统，改进测试过程
- 5.23 反模型测试：在测试过程中，从数据里抽象出模型
- 5.24 成分测试：将复杂大规模软件测试分解为测试该软件的各个组成部分
- 5.25 有限状态机测试：验证系统在不同输入下状态转移是否符合预期
- 5.26 基于 Petri 网的测试：利用 Petri 网理论进行建模和测试的方法
- 5.27 基于模型检查的测试：通过数学模型精确描述系统行为，系统地验证所有可能的状态和路径
- 5.28 TTCN 测试：使用 TTCN 定义测试用例、执行过程，确保测试对象符合其设计规范
- 5.29 布尔规格测试：将系统行为表示为布尔逻辑公式，验证其是否满足预定的逻辑条件
- 5.30 基于统一建模语言测试：应用待测试软件的 UML 模型来获得软件测试的需求和覆盖准则
- 5.31 差分测试：确保新版本软件相对于旧版本或不同实现之间的兼容性和一致性
- 5.32 故障注入测试：故意向系统中注入故障，验证系统在面对错误或异常情况下的反应和处理能力

6 简单分析评论题（软件的特点是什么？测试方法有哪些特殊性？）（10 分，每题 1 分）

- 6.1 面向对象软件的测试：封装、继承、多态、复杂交互、数据行为联动
- 6.2 面向方面的软件测试：容易维护 高度模块化且可重用 容易理解和演化
- 6.3 面向服务的软件测试：服务间交互验证 服务独立性 动态性扩展性 容错性与恢复能力 消息驱动与异步通信
- 6.4 构件软件测试：组件可观察性 组件可控制性 组件可追踪性 组件可理解性
- 6.5 Web 应用软件测试：Web 应用存在于网络上，和大量不同操作系统、浏览器、硬件平台等交互
- 6.6 普适计算环境下的软件测试：在一个无处不在且高度集中的计算环境中，对 软件系统测试
- 6.7 云测试：灵活性、可扩展性 成本效益 快速部署
- 6.8 物联网环境下的软件测试：对物联网设备及其系统性能、功能和安全性的评估过程
- 6.9 并行软件测试：在开发并行处理或多线程应用程序时，针对并行特性进行的测试
- 6.10 嵌入式软件测试：验证软件质量和嵌入式系统的可靠性要求
- 6.11 高可信软件测试：验证在外部干扰存在时，仍能提供连续的服务
- 6.12 网构软件测试：在互联网开放、动态和多变环境下软件系统基本形态的一种抽象
- 6.13 移动应用软件测试：确保移动应用质量和用户体验的重要过程
- 6.14 人工智能软件测试：不透明性 数据依赖性，验证数据有效 动态变化 评估模型的一致性 伦理和法律合规性

6.15 区块链软件系统测试：确保开发安全、可靠和高效区块链应用

6.16 元宇宙测试：沉浸感 社交互动 经济系统，虚拟货币 用户生成内容 跨平台兼容性