**1．简述质量控制 、质量保证的概念。简述V形模型中的软件测试类型及其主要内容。简述白盒测试和黑盒测试的概念及其主要方法及内容。**

**质量控制**：是通过检测和测试活动，验证产品是否符合既定的质量标准和规范的过程。其目的是发现和修正产品缺陷，确保最终产品的质量。强调产品的“结果”质量。主要活动包括测试、检查、审计等。关注实际生产中的缺陷检测和纠正。

**质量保证**：是通过建立合理的管理体系、制定规范和流程，确保产品在开发和生产过程中能够满足质量要求的过程。其目的是预防质量问题的发生。它强调过程的“预防”， 涉及管理制度的制定、规范的执行和过程改进，关注过程的有效性和效率。

**V形模型中软件测试类型**：① 单元测试：测试代码模块的功能和逻辑，通常由开发人员完成，验证单个模块是否满足设计需求，基于代码的白盒测试方法，如语句覆盖、分支覆盖；② 集成测试：测试多个模块之间的接口和交互行为，验证模块之间的集成是否正确，数据流和控制流是否正常。可以使用驱动程序（Driver）和桩（Stub）模块进行模拟测试；③ 系统测试：测试整个软件系统的功能、性能和可靠性，测试整个软件系统的功能、性能和可靠性，黑盒测试为主，包括功能测试、性能测试、兼容性测试等；④ 验收测试：由客户或用户进行的测试，验证系统是否满足业务需求，确保系统能在实际环境中正常运行，满足用户需求，基于需求的黑盒测试。

**白盒测试**：

* 概念：测试者基于软件的内部结构和实现细节进行测试，目的是验证代码逻辑的正确性。
* 主要方法：
  + 语句覆盖：确保每条语句至少执行一次
  + 分支覆盖：确保每个分支（如if-else）的每种可能都被执行一次。
  + 路径覆盖：确保所有可能的执行路径都被测试。
* 主要内容：
  + 验证代码逻辑的正确性。
  + 检测逻辑漏洞、边界问题和算法的正确性。

**黑盒测试：**

* 概念：测试者不关心软件的内部实现，只基于输入和输出验证软件功能。
* 主要方法：
  + 等价类划分：将输入划分为有效和无效的等价类，分别测试。
  + 边界值分析：测试输入数据的边界值及其附近的情况。
  + 决策表测试：基于输入条件和输出的组合关系构造测试用例。
  + 场景测试：模拟真实用户场景进行测试。
* 主要内容：
  + 验证软件功能是否符合需求。
  + 检测界面交互、功能实现、输入输出正确性等问题。

**2、针对下图简单方法，画出程序流程图，标记分支，同时画出控制流图。然后选择测试用例，分别完成语句覆盖、分支覆盖、条件覆盖、条件组合覆盖和路径覆盖测试。**

**要求针对不同的覆盖，给出执行过程和对应的测试用例，以及执行后的输出**

**void func(int a, int b, int c, int d) {**

**if(a>0 && b>100)**

System.***out.println("good");***

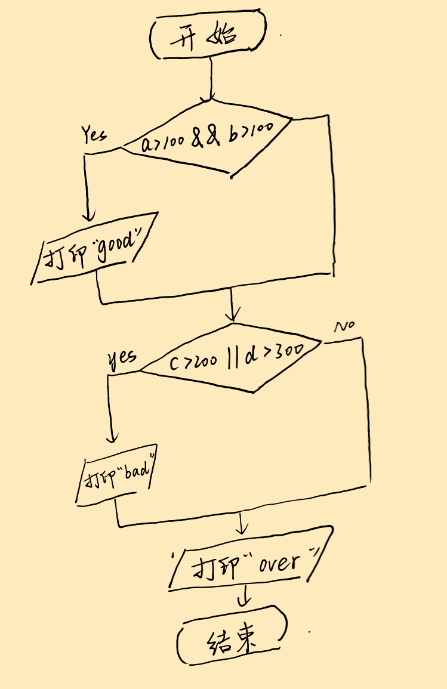
**if(c>200 || d>300)**

System.***out.println("bad");***

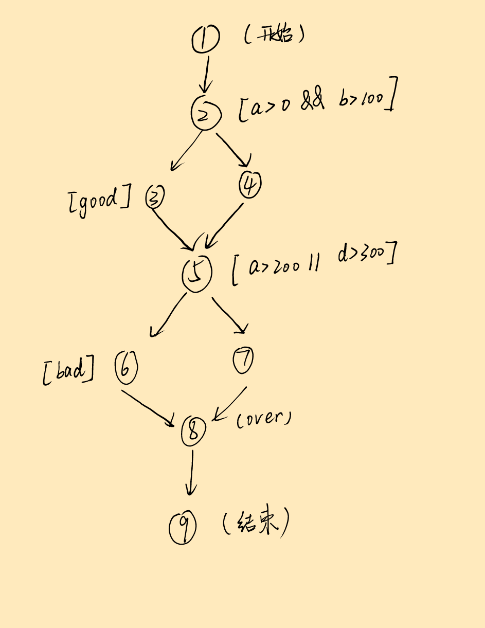
System.***out.println(“over");***

}

**程序流程图**



**控制流图**：



**语句覆盖:**

* 测试用例1: a=1, b=101, c=201, d=0
* 执行路径：开始→good→bad→over→结束
* 输出：good、bad、over

**分支覆盖：**

* 测试用例1: a=1, b=101, c=201, d=0 (第一个if为真，第二个if为真)
* 测试用例2: a=0, b=0, c=0, d=0 (第一个if为假，第二个if为假)
* 输出1：good、bad、over
* 输出2：over

**条件覆盖：**

* 测试用例1: a=1, b=101, c=201, d=0 (a>0真, b>100真, c>200真, d>300假)
* 测试用例2: a=0, b=50, c=100, d=301 (a>0假, b>100假, c>200假, d>300真)
* 输出1：good、bad、over
* 输出2：bad、over

**条件组合覆盖**：

* 第一个if(a>0 && b>100)的组合
  + 测试用例1: a=1, b=101, c=0, d=0 (T,T)
  + 测试用例2: a=0, b=101, c=0, d=0 (F,T)
  + 测试用例3: a=1, b=50, c=0, d=0 (T,F)
  + 测试用例4: a=0, b=50, c=0, d=0 (F,F)
* 第二个if(c>200 || d>300)的组合
  + 测试用例5: a=0, b=0, c=201, d=301 (T,T)
  + 测试用例6: a=0, b=0, c=201, d=0 (T,F)
  + 测试用例7: a=0, b=0, c=0, d=301 (F,T)
  + 测试用例8: a=0, b=0, c=0, d=0 (F,F)

**路径覆盖：**

* 路径1: a>0&&b>100为假 → c>200||d>300为假 → over  
  测试用例: a=0, b=50, c=100, d=100  
  输出: over
* 路径2: a>0&&b>100为假 → c>200||d>300为真 → bad → over  
  测试用例: a=0, b=50, c=201, d=100  
  输出: bad、over
* 路径3: a>0&&b>100为真 → good → c>200||d>300为假 → over  
  测试用例: a=1, b=101, c=100, d=100  
  输出: good、over
* 路径4: a>0&&b>100为真 → good → c>200||d>300为真 → bad → over  
  测试用例: a=1, b=101, c=201, d=100  
  输出: good、bad、over

3、

某保险公司系统需要输入年龄，体重，类型为整数型，年龄范围：45岁至59岁（45<=age<=59），体重范围：55到70公斤（55<=weight <=70 ），请结合等价类划分、边界值分析法设计测试用例。

要求：用表格给出等价类划分（给出有效和无效等价类），和边界值分析的结果。

合理设计用例，用表格给出最终的测试用例

* 等价类划分

|  |  |  |
| --- | --- | --- |
| 参数 | 年龄 | 体重 |
| 年龄 | C1: 45 ≤ age ≤ 59 | C2: age < 45  C3: age > 59 |
| 体重 | C4: 55 ≤ weight ≤ 70 | C5: weight < 55  C6: weight > 70 |

* 边界值分析：

|  |  |
| --- | --- |
| 参数 | 边界值 |
| 年龄 | 44 (最小值-1)  45 (最小值)  46 (最小值+1)  58 (最大值-1)  59 (最大值)  60 (最大值+1) |
| 体重 | 54 (最小值-1)  55 (最小值)  56 (最小值+1)  69 (最大值-1)  70 (最大值)  71 (最大值+1) |

* 最终测试用例的分析：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | 测试场景 | 年龄 | 体重 | 预期结果 | 测试依据 |
| 1 | 有效等价类组合 | 50 | 50 | 有效 | 有效等价类 |
| 2 | 年龄下边界 | 45 | 60 | 有效 | 边界值 |
| 3 | 年龄上边界 | 59 | 60 | 有效 | 边界值 |
| 4 | 年龄上边界 | 59 | 60 | 有效 | 边界值 |
| 5 | 体重下边界 | 50 | 55 | 有效 | 边界值 |
| 6 | 体重上边界 | 50 | 70 | 有效 | 边界值 |
| 7 | 年龄下边界-1 | 44 | 60 | 无效 | 边界值 |
| 8 | 年龄上边界+1 | 60 | 60 | 无效 | 边界值 |
| 9 | 体重下边界-1 | 50 | 54 | 无效 | 边界值 |
| 10 | 体重上边界+1 | 50 | 71 | 无效 | 边界值 |
| 11 | 年龄无效、体重无效 | 40 | 50 | 无效 | 无效等价类 |
| 12 | 年龄有效、体重无效 | 50 | 80 | 无效 | 无效等价类 |
| 13 | 年龄无效、体重有效 | 65 | 60 | 无效 | 无效等价类 |
| 14 | 两个边界最小值 | 45 | 55 | 有效 | 边界值组合 |
| 15 | 两个边界最大值 | 59 | 70 | 有效 | 边界值组合 |
| 16 | 年龄下边界+1、体重有效 | 46 | 60 | 有效 | 边界值 |
| 17 | 年龄有效、体重下边界+1 | 50 | 56 | 有效 | 边界值 |
| 18 | 年龄上边界-1、体重有效 | 58 | 60 | 有效 | 边界值 |
| 19 | 年龄有效、体重上边界-1 | 50 | 69 | 有效 | 边界值 |