

第 1 次课程作业

1、Please give a case about software fault that you like best. And give the reason why you are interested in. (请给出一个你最喜欢的软件缺陷案例，并解释原因。)

2、Please provide your good software practices to share with others. (请将你的好软件工程(开发、测试、管理等)经验详细描述出来，以便于和大家分享。)

3、What's the Software Quality Assurance? what components does SQA include? (软件质量保证的定义是什么？SQA 包含哪些类型的活动？)

参考答案：

软件质量保证是一组系统化的、有计划的活动，用来充分确认软件开发过程或维护过程符合功能性需求、技术性需求以及管理需求，从而保证软件产品在满足预算的情况下按时完成。SAQ 包括：项目前质量活动，全项目生命周期质量活动，错误预防和改进的基础设施方面的质量活动，软件质量管理活动，标准、认证以及 SQA 评估活动，SQA 的组织（人的因素）方面的质量活动。

4、Please compare SQA with SQC. (请比较软件质量保证和软件质量控制。)

参考答案：

1) 二者的目标不同：(1) 软件质量控制的主要目标是拒绝(抑制)质量不达标的产品。已开发完成的软件，只要还未交给用户，软件质量控制活动就会发生。(2) 软件质量保证的主要目标是最小化软件的成本，减少软件不能按时交付的比例。

2) 具体来讲，可以认为软件质量控制知识软件质量保证的一部分。

5、Please give the definitions of software quality and defect. (请给出软件质量和缺陷的定义。)

参考答案：

软件质量：软件满足用户需求(功能性及非功能性)的程度，一定程度上可以认为质量是软件对某些人的价值。当然，不同的人对质量的理解和要求不同。

软件缺陷：在某些涉众眼中，软件中存在的导致其质量下降的任何事情(如存在 bug、界面不美观、操作不便等)。

第 2 次课程作业

1、RUP 是什么？它包含哪些阶段？包含哪些核心规程？请详细解释其二维模型。

参考答案：

The Rational Unified Process (**RUP**) is a software engineering process framework that provides a disciplined yet flexible approach to assigning tasks and responsibilities within a software development organization.

RUP 包括 4 个阶段：初始、细化、构造、产品化。

RUP 包含 6 个核心环节(规程)：业务(商业)建模、需求、分析与设计、实现、测试、部署。

RUP 采用二维图示描述了一个软件项目的研发过程，横轴表示开发活动的阶段和迭代(动态或时间维度)；纵轴描述了与软件项目开发有关的活动(规程，静态维度)。在 RUP 中，一个软件项目分为 4 个阶段，每个阶段又可包含多次迭代。每个迭代产生一个主要里程碑，每次迭代产生一个次要里程碑。每次迭代，可能涉及多个环节(规程)，如需求、分析与设计、实现、测试等。

2、RUP 中包含哪些测试角色？简述每类角色的职责。

参考答案：

RUP 中包括 4 类测试角色：测试经理、测试分析师、测试设计师、测试员。

测试经理的主要职责：为整个测试工作负责、确定测试工作范围、分配测试任务、汇报测试整体工作（编写测试总结报告）。

测试分析师的主要职责：分析测试对象、设计测试用例、设计测试数据、分析测试结果。

测试设计师的主要职责：确定测试方法、设计自动化测试框架、开发自动化测试工具、配置测试环境等。

测试员的主要职责：执行具体的测试工作、记录测试结果和测试日志。

第 3 次课程作业

1、案例描述：某公司根据年终考核给员工发放年终奖，年终考核包括三个指标：本年度工作完成情况、工作能力、工作态度，根据这三个指标的得分计算出某位员工的总得分。其中：本年度工作完成情况占总成绩的 50%，本年度工作完成情况最高得分为 100 分；工作能力总分为 20 分；工作态度总分为 30 分。根据员工的总得分，将员工分为如下等级：A--[85,100]、B--[60,85)、C--[40,60)、D--[0, 40)。对于评分等级为 A 的员工，其年终奖额外奖励 10%；对于等级为 B 的员工，其年终奖正常得到；对于等级为 C 的员工，扣除 20%的年终奖；对于等级为 D 的员工，则进行辞退、不发年终奖。假定该公司给每位员工的基础年终奖都相同，为 Total。

问题：请结合课上所学路径测试相关知识设计测试用例。请给出本案例的（伪）代码，根据（伪）代码画出程序图，给出 DD 路径，然后针对逻辑覆盖给出的 6 种指标设计测试用例。

参考答案：

代码：

参数说明：work_task—[int 类型，工作完成情况的得分]，work_capacity—[int 类型，工作能力的得分]，work_attitude —[int 类型，工作态度的得分]，work_total —[int 类型，员工总得分]，work_bonus —[int 类型（也可为 float 类型），员工的年终奖]，rate —[String 类型，员工等级]

```
1. String calculateBonus(int work_task,int work_capacity,int work_attitude){  
2.     int work_total, work_bonus,Total=20000;  
3.     String rate;  
4.     If(work_task<0|| work_task>100)  
5.         return “工作完成情况的输入有误”;  
6.     If(work_capacity <0|| work_capacity >20)  
7.         return “工作能力的输入有误”;  
8.     If(work_attitude <0|| work_attitude >30)  
9.         return “工作态度的输入有误”;  
10.    work_total= (int)work_task*50%+ work_capacity+ work_attitude;  
11.    switch(work_total){  
12.        Case 1: 85 <= work_total <= 100  
13.            work_bonus = Total*(1+10%);  
14.            rate=”A”;  
15.            break;  
16.        Case 2: 60 <= work_total < 85  
17.            work_bonus = Total;
```

```

18.         rate="B";
19.         break;
20.     Case 3: 40 <= work_total < 60
21.         work_bonus = Total*(1-20%);
22.         rate="C";
23.         break;
24.     Case 4: 0 <= work_total < 40
25.         work_bonus = 0;
26.         rate="D";
27.         break;
28.     }
29.     return "该员工等级为"+rate+",年终奖为"+ work_bonus;
30. }

```

程序代码对应的程序图如下：

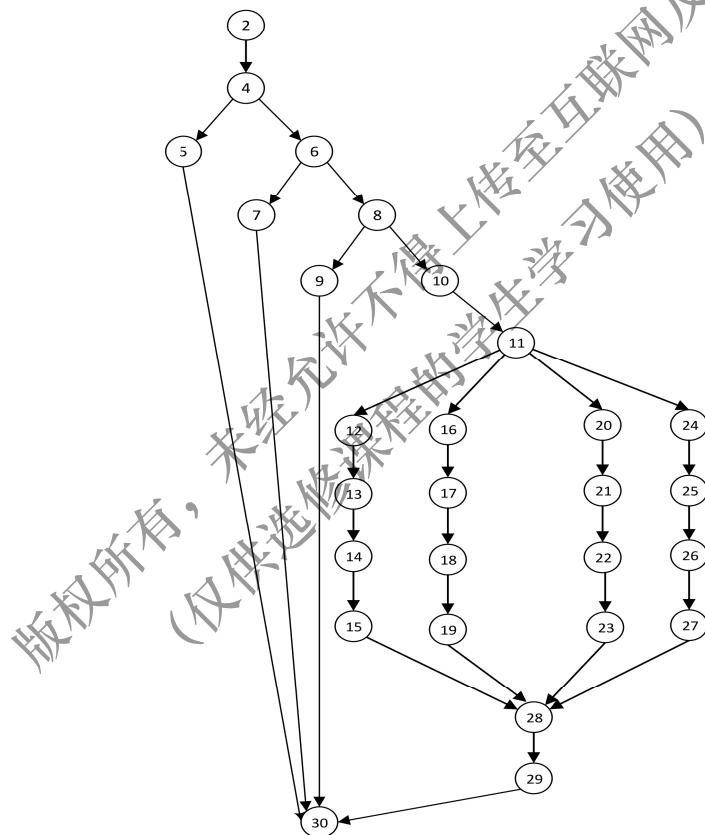
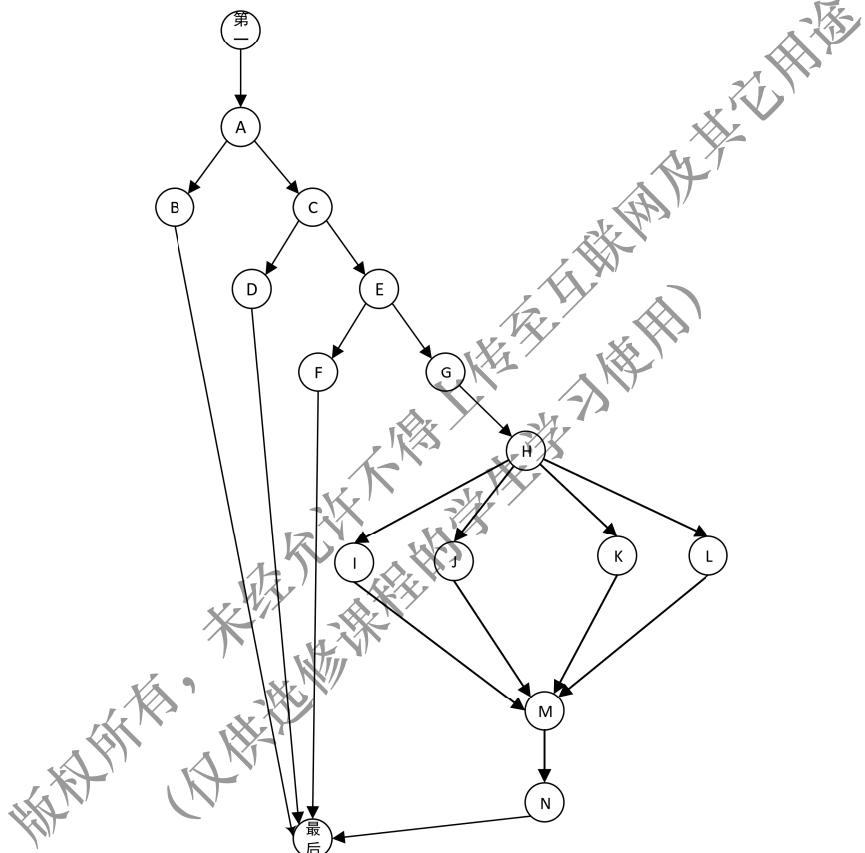


表 1 计算年终奖程序的 DD 路径

程序图节点	DD 路径名称	定义情况
2	第一	1
4	A	3
5	B	4
6	C	3
7	D	4
8	E	3
9	F	4

10	G	4
11	H	3
12~15	I	5
16~19	J	5
20~23	K	5
24~27	L	5
28	M	3
29	N	4
30	最后	2

由 DD 路径的表格，可得如下的 DD 路径图：



由 DD 路径图可得计算年终奖程序的可执行路径

表 2 计算年终奖程序的可执行路径

编号	可行路径
P1	第一-A-B-最后
P2	第一-A-C-D-最后
P3	第一-A-C-E-F-最后
P4	第一-A-C-E-G-H-I-M-N-最后
P5	第一-A-C-E-G-H-J-M-N-最后
P6	第一-A-C-E-G-H-K-M-N-最后
P7	第一-A-C-E-G-H-L-M-N-最后

那么根据计算年终奖程序的伪码，针对逻辑覆盖给出的 6 种指标，分别设计测试用例。

1) 语句覆盖

对于计算年终奖程序，为了满足语句覆盖，需要设计 7 个测试用例即可，如表 3 所示。

表 3 计算年终奖程序的语句覆盖测试用例

编号	输入			通过路径
	work_task	work_capacity	work_attitude	
1	110	20	15	P1
2	80	25	15	P2
3	80	15	-1	P3
4	95	20	15	P4
5	80	15	10	P5
6	60	15	5	P6
7	40	10	5	P7

2) 判定覆盖

计算年终奖程序总共包含 4 个的判定，分别为：

语句 4: $\text{work_task} < 0 \text{|| work_task} > 100$, 有 2 条分支, L1: True, R1: False;

语句 6: $\text{work_capacity} < 0 \text{|| work_capacity} > 20$, 有 2 条分支, L2: True, R2: False;

语句 8: $\text{work_attitude} < 0 \text{|| work_attitude} > 30$, 有 2 条分支, L3: True, R3: False;

语句 11: $\text{switch}(\text{work_total})$, 共有 4 条分支, S1: $85 \leq \text{work_total} \leq 100$, S2: $60 \leq \text{work_total} < 85$, S3: $40 \leq \text{work_total} < 60$, S4: $0 \leq \text{age} < 40$

所以针对判定覆盖给出的测试用例如表 4 所示：

表 4 计算年终奖程序的判定覆盖测试用例

编号	输入			通过路径	判定取值
	work_task	work_capacity	work_attitude		
1	110	20	15	P1	R1 L2 L3 —
2	80	25	15	P2	L1 R2 L3 —
3	80	15	-1	P3	L1 L2 R3 —
4	95	20	15	P4	L1 L2 L3 S1
5	80	15	10	P5	L1 L2 L3 S2
6	60	15	5	P6	L1 L2 L3 S3
7	40	10	5	P7	L1 L2 L3 S4

3) 条件覆盖

整个程序的条件整理：

(1) 语句 4 $\text{work_task} < 0 \text{|| work_task} > 100$:

条件 $\text{work_task} < 0$ 取真为 T1, 取假为 F1

条件 $\text{work_task} > 100$ 取真为 T2, 取假为 F2

(2) 语句 6 $\text{work_capacity} < 0 \text{|| work_capacity} > 20$:

条件 $\text{work_capacity} < 0$ 取真为 T3, 取假为 F3

条件 work_capacity > 20 取真为 T4, 取假为 F4

(3) 语句 8 work_attitude < 0 || work_attitude > 30:

条件 work_attitude < 0 取真为 T5, 取假为 F5

条件 work_attitude > 30 取真为 T6, 取假为 F6

(4) 语句 8 switch(work_total), 共有 4 个条件, S1: 85 <= work_total <= 100, S2:

60 <= work_total < 85, S3: 40 <= work_total < 60, S4: 0 <= age < 40

则满足条件覆盖的测试用例如表 5 所示:

表 5 计算年终奖程序的条件覆盖测试用例

编号	输入			通过路径	条件取值
	work_task	work_capacity	work_attitude		
1	110	20	15	P1	F1 T2 F3 F4 F5 F6
2	80	25	15	P2	F1 F2 F3 T4 F5 F6
3	80	15	-1	P3	F1 F2 F3 T4 F5 T6
4	-1	20	15	P4	T1 F2 F3 F4 F5 F6
5	80	-1	15	P2	T1 F2 T3 F4 F5 F6
6	80	15	-1	P3	T1 F2 T3 F4 T5 F6
7	95	20	15	P4	F1 F2 T3 F4 T5 F6 S1
8	80	15	10	P5	F1 F2 T3 F4 T5 F6 S2
9	60	15	5	P6	F1 F2 T3 F4 T5 F6 S3
10	40	10	5	P7	F1 F2 T3 F4 T5 F6 S4

4) 判定-条件覆盖是判定覆盖和条件覆盖的综合, 条件覆盖的测试用例满足判断-条件覆盖, 详见表 4。

5) 条件组合覆盖

条件组合覆盖需要使每个判断的所有可能的条件取值组合至少执行一次, 根据程序的实际情况, 得到如下的取值组合:

work_task = -1 作 T1F2

work_task = 110 作 F1T2

work_task = 80 作 F1F2

work_capacity = -1 作 T3F4

work_capacity = 25 作 F3T4

work_capacity = 15 作 F3F4

work_attitude = -1 作 T5F6

work_attitude = 35 作 F5T6

work_attitude = 25 作 F5F6

语句 8 switch(work_total), 共有 4 个条件, S1: 85 <= work_total <= 100, S2: 60 <=

`work_total < 85, S3: 40 <= work_total < 60, S4: 0 <= age < 40` 取各自的条件即可。

表 4 符合条件组合覆盖。

6) 路径覆盖

由于年终奖程序中不存在循环，共存在 7 条路径，语句覆盖的测试用例满足路径覆盖，详见表 3。

第 4 次课程作业

1、为什么需要进行确认构建稳定性，并改进测试资产？

参考答案：

1) 因为目前主流的软件项目开发模式为迭代式开发，迭代式开发模式频繁地产生不同版本的构建；如果某个构建不稳定，会浪费大量的测试工作。因此，在进行正式测试之前，需要对拟测试的构建进行稳定性验证；如果构建不稳定，可以不对它进行进一步的测试。

2) 在测试过程中，产生了大量的测试资产；随着软件研发的进展，有些测试资产不再适用，需要移除或者修改；对于有些新功能需要增加新的测试用例等资产。因此，需要对测试用例、测试数据、测试工具、测试报告等进行测试资产进行改进。