

# 《软件工程》复习说明

## 一、 单项选择题（40 分）

30×课内题+10×课外题

## 二、 应用题（60 分）：

每 1 道大题可能需要 2 种不同的分析方法，将手工绘图借助软件（扫描全能王、Microsoft Office Lens 甚至内置图库）扫描为 PDF 文档并上传至[课堂派](#)。

### 1. 数据流图（第 50 页）

E-R 图不用画，提到的内容必须要画出来。如果题目要求单独划分存储过程和处理过程，则有两张数据流图。

矩形为外部实体，箭头为数据流，圆形为数据交换/处理，上下双线为数据存储。

### 示例：图书订购系统DFD

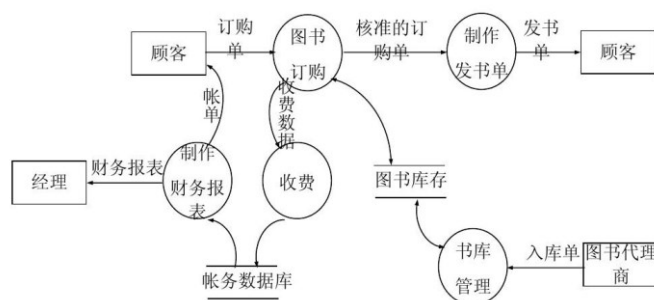


图 1 数据流图（Yourdon 表示法）

### 2. 判定表（第 56 页）

不要求化到最简模式，只要正常列出所有判定条件+正确绘制表格即可。

决策规则号		1	2	3	4	5	6
条件	A	Y	Y	Y	N	N	N
	B	N	N	N	Y	Y	Y
条件	$N \leq 50$	Y	N	N	Y	N	N
	$50 < N \leq 100$	Y	N	N	N	Y	N
	$100 < N \leq 150$	N	Y	N	N	N	Y
	$N > 150$	N	N	Y	N	N	Y
奖励	小于或等于100	Y					
	大于100小于等于150		Y				
	大于150			Y			
	小于或等于50				Y		
	大于50小于等于100					Y	
	大于100						Y

图2 判定表

### 3. 用例图（第 110 页）

在 UML 中，参与者使用人形符号表示，并且具有唯一的名称；用例使用椭圆表示，也具有唯一的名称。参与者和用例之间使用带箭头的实线连接，由参与者指向用例。

参与者之间可以存在泛化关系，类似的参与者可以组成一个层级结构。用三角空心箭头表示，由泛化参与者指向基础参与者（“会员”是“游客”的泛化，由“会员”指向“游客”）。

用例之间的关系有包含（include）、扩展（extend）和泛化（generalization）3 种。“包含”与“扩展”使用带箭头虚线表示，并在线上标注，由包含/扩展用例指向被包含/扩展用例；“泛化”使用空心三角形箭头表示，由子用例指向父用例。

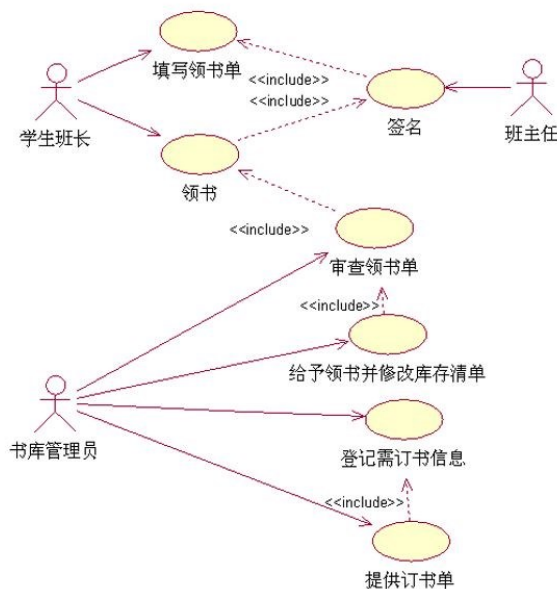


图3 用例图

### 4. 类图（第 110 页）

在 UML 中，类图用具有 3 个分隔线的矩形表示。顶层分隔表示类和对象的名称，中间表示属性，底层表示操作。类名使用大驼峰写法，首字母大写；对象名、属性名、方法名使用小驼峰写法，首字母小写。需

要在属性和方法上标注可见性，可见性等级顺序为：Private (-) < Protected (#) < Package (~) < Public (+)。

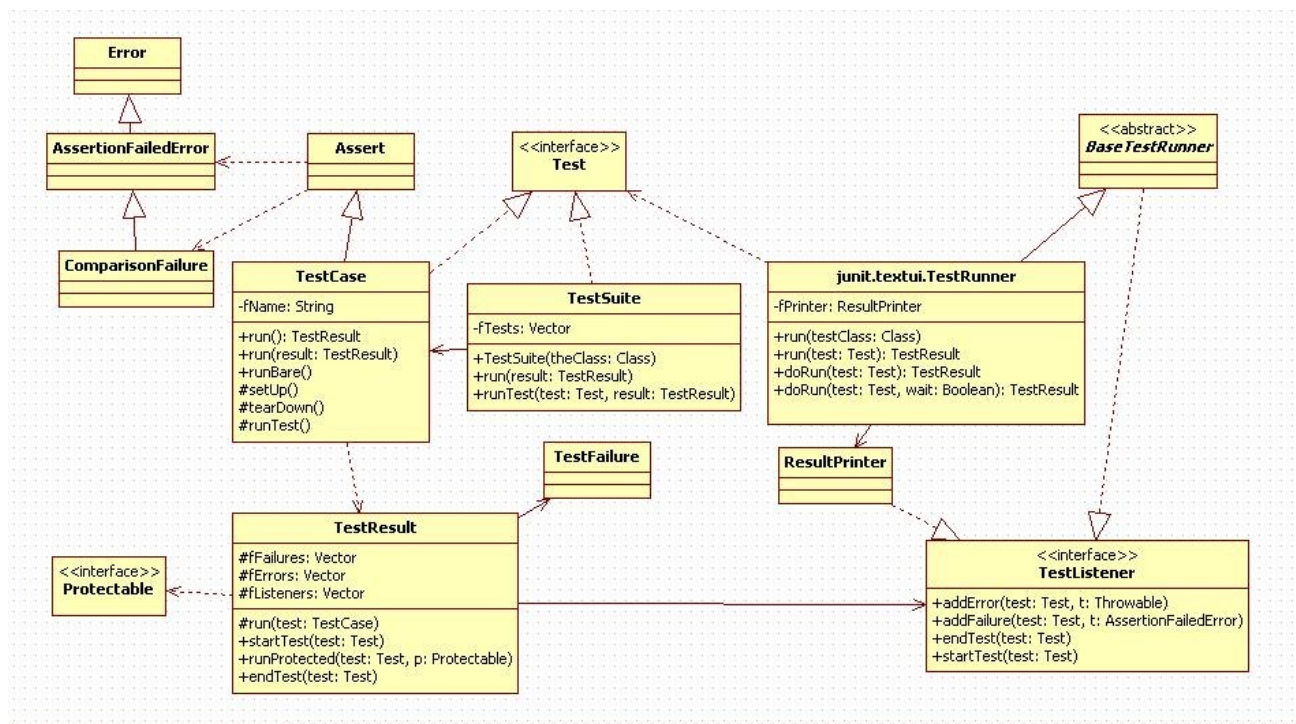


图 4 类图

关联关系细分为二元关联、多元关联、受限关联、聚集和组合。

- ◇ 二元关联：实线，两端标记重数（0、1、\*）
- ◇ 多元关联：菱形在中间，实线，类一端标记重数
- ◇ 受限关联：与二元/多元相同，多重端添加限定对象
- ◇ 聚合：实线，两端标记重数，整体端添加空心菱形箭头
- ◇ 组合：与聚合相似，使用实心菱形

若元素 X 的变化会引起元素 Y 的变化，则 Y 依赖于 X，X 为提供者、Y 为客户。依赖关系使用指向提供者的虚线箭头表示。

泛化即继承（Extends），以空心三角形箭头实线从子类指向父类。

实现（Implements）与接口相关，类似于泛化，使用虚线，从实现类指向接口。

## 5. 顺序图（第 116 页）

顺序图由对象（参与者的实例也是对象）、生命线、控制焦点、消息和销毁记号组成，使用直尺作图。

对象是位于图像顶部的矩形，按照参与者→边界类→控制类→实体类的顺序从左至右排列；生命线是一条垂直的虚线，表示对象的存在时间，虚线末端添加以×表示的销毁记号；控制焦点是一个细长的矩形，表示对象执行一个操作经历的时间段；消息是作用于控制焦点上的一条水平带箭头的实线，表示消息传递，同时注明消息的用途。

简单消息以实线箭头表示，返回以虚线箭头表示，同步消息在实线箭头上添加×标记，异步消息以无箭

头实线表示。

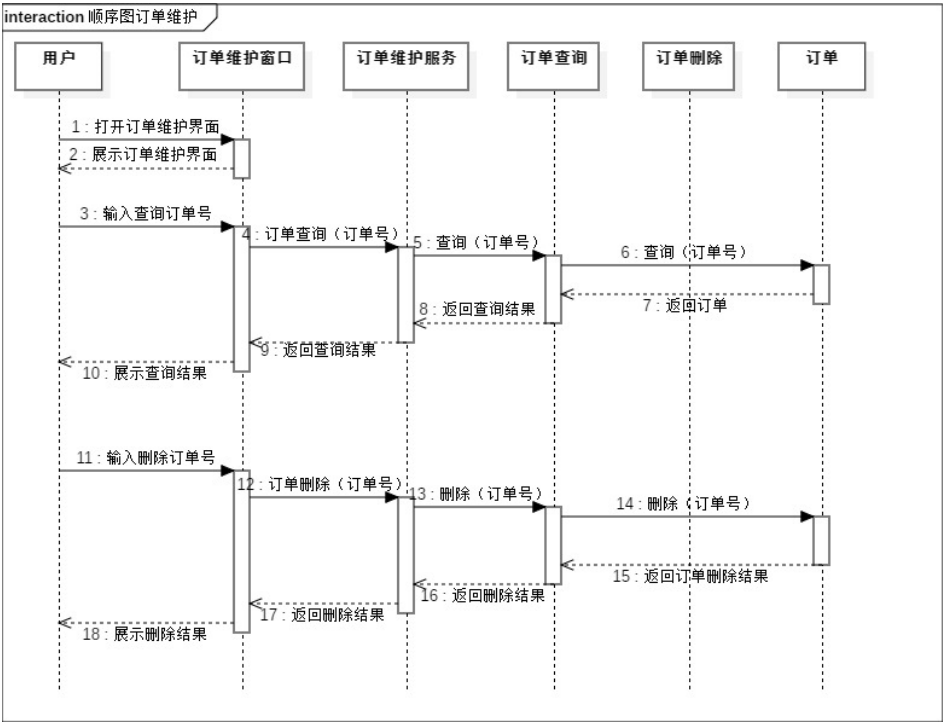


图 5 顺序图

6. 状态图（第 53、118 页）

在状态转换图中定义的状态主要有：初态（即初始状态）、终态（即最终状态）和中间状态。

初态用一个黑圆点表示；终态在黑圆点外加一个圆；中间状态用圆角矩形表示（用 2 条横线分为 3 个部分，顶部必须为状态名称，中部可选为状态变量的名称和值，底部可选为活动表）；状态转换用带箭头实线表示，并标注触发事件的名称。

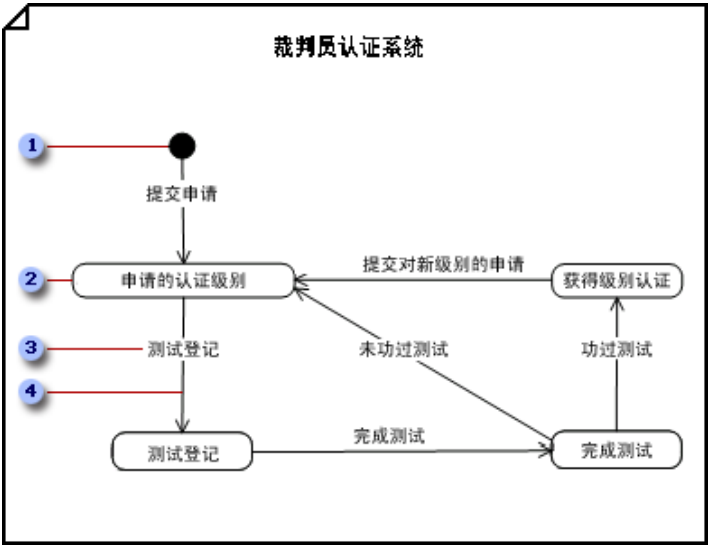


图 6 状态图

7. 逻辑覆盖法（第 233 页）

- ✓ 语句覆盖：输入若干测试用例，使程序中的所有执行语句至少执行 1 次
- ✓ 分支覆盖：输入若干测试用例，使程序中每个条件判断的真值分支和假值分支至少运行 1 次
- ✓ 条件覆盖：输入若干测试用例，使每个判断的所有逻辑条件的每种可能取值至少执行 1 次
- ✓ 分支-条件覆盖：同时满足分支覆盖和条件覆盖的要求，用例取两者并集
- ✓ 条件组合覆盖：输入若干测试用例，使每个判断语句的所有逻辑条件的可能取值组合至少执行 1 次
- ✓ 路径覆盖：覆盖所有可能的路径

## 8. 基本路径测试法（第 235 页）

- ✓ 绘制控制流图：每个圆圈都是节点，代表 1 个或多个无分支的（源代码）语句；箭头成为边或连接，代表控制流

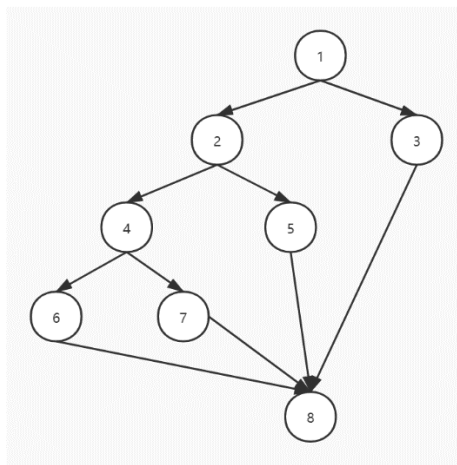


图 7 控制流图

- ✓ 计算环路复杂度（平面图的面数）： $F = E - V + 2$ （几何欧拉公式）
- ✓ 产生线性无关路径集
- ✓ 设计测试用例：每个用例分别对应一条线性无关路径集