3．工资计算系统中的一个子系统有如下功能：

（ 1 ）计算扣除部分—由基本工资计算出应扣除（比如水电费、缺勤）的部分；

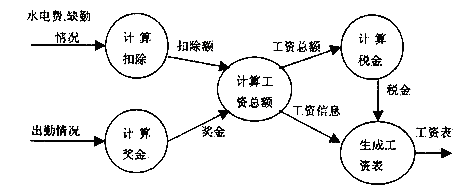
（ 2 ）计算奖金部分—根据职工的出勤情况计算出奖励金；

（ 3 ）计算工资总额部分—根据输入的扣除额及奖金计算出总额；

（ 4 ）计算税金部分—由工资总额中计算出应扣除各种税金；

（ 5 ）生成工资表—根据计算总额部分和计算税金部分传递来的有关职工工资的详细信息生成工资表。

试根据要求画出该问题的数据流程图。



4一个考试录取统计分数子系统有如下功能：

(1) 计算标准分：根据考生原始分计算，得到标准分，存入考生分数文件；

(2) 计算录取线分：根据标准分、招生计划文件中的招生人数，计算录取线，存入录取线文件。

试根据要求画出该系统的数据流程图。



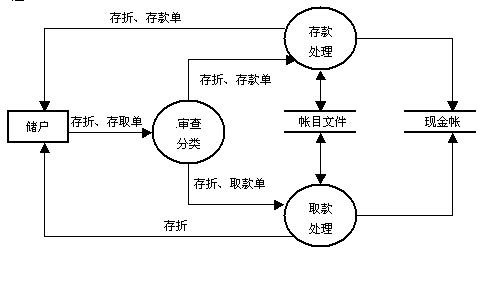
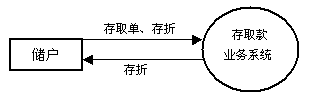
11欲开发一个银行的活期存取款业务的处理系统：储户将填好的存/取款单和存折交给银行工作人员，然后由系统作以下处理；

　　(1)业务分类处理：系统首先根据储户所填的存/取款单，确定本次业务的性质，并将存/取款单和存折交下一步处理；

　　(2)存款处理：系统将存款单上的存款金额分别记录在存折和帐目文件中，并将现金存入现金库；最后将存折还给储户；

(3)取款处理：系统将取款单上的取款金额分别记录在存折和帐目文件中，并从现金库提取现金；最后将现金和存折还给储户。

该系统的总体图如下图所示，请画出该系统的零级图。



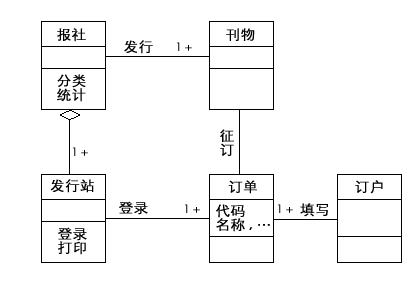
3.某报社采用面向对象技术实现报刊征订的计算机管理系统，该系统基本需求如下：

  （1）报社发行多种刊物，每种刊物通过订单来征订，订单中有代码，名称，订期，单价，份数等项目，订户通过填写订单来订阅报刊。

  （2）报社下属多个发行站每个站负责收集登陆订单打印收款凭证等事务。

（3）报社负责分类并统计各个发行站送来的报刊订阅信息。

  请就此需求建立对象模型。（10分）



1.某“调整工资”处理模块接受一个“职称”的变量，根据职称的不同（助教，讲师，副教授，教授）作不同的处理，其中若是助教还必须输入工龄，只有工龄超过两年才能调整工资。请用等价类划分法设计测试用例。（7分）

[〔答案〕](http://student.zjzk.cn/course_ware/software/test/test_show.asp?filename=test_2.txt&fname=模拟试卷二&bgp=title2.gif#51#51)

1.划分等价类：

|  |  |  |
| --- | --- | --- |
| 输入条件 | 合理等价类 | 不合理等价类 |
| 职称 | |  | | --- | | ①教授 | | ②副教授 | | ③讲师 | | ⑤四种职称之外任意一种 |
| 职称兼工龄 | ④助教兼工龄大于2年 | |  | | --- | | ⑥助教兼工龄等于两年 | | ⑦助教兼工龄小于两年 | |

设计测试用例：

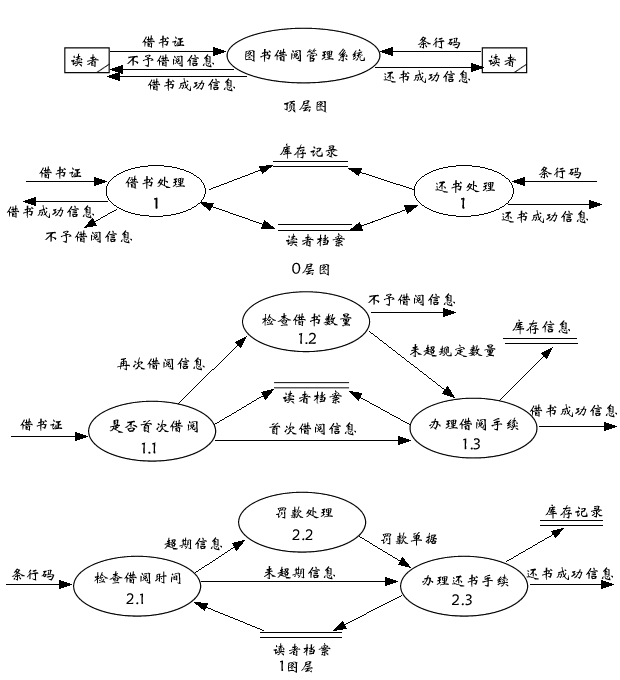
|  |  |  |
| --- | --- | --- |
| 输入数据 | 预期结果 | 覆盖范围 |
| 教授 | 输入有效，进行调整工资处理 | ① |
| 副教授 | 输入有效，进行调整工资处理 | ② |
| 讲师 | 输入有效，进行调整工资处理 | ③ |
| 助教  3 | 输入有效，进行调整工资处理 | ⑤ |
| 助教   2 | 输入有效，不调整工资处理 | ⑥ |
| 助教   1 | 输入有效，不调整工资处理 | ⑦ |
| 工程师 | 输入无效 | ④ |

2.某图书馆借阅系统有以下功能：

（1）借书：根据读者的借书证查询读者档案，若借书数目未超过规定数量，则办理借阅手续（修改库存记录及读者档案），超过规定数量者不予借阅。对于第一次借阅者则直接办理借阅手续。

（2）还书：根据读者书中的条形码，修改库存记录及读者档案，若借阅时间超过规定期限则罚款。

  请对以上问题，画出分层数据流图。（8分）



4下面是一段求最大值的程序，其中datalist是数据表，n是datalist的长度。

int GetMax(int n, int datalist[ ])  {       int k=0;

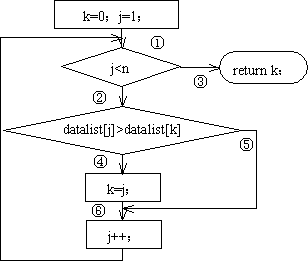
               for ( int j=1; j<n; j++ )

                        if ( datalist[j] > datalist[k] ) k=j;

                return k; }

（1） 画出该程序的控制流图，并计算其McCabe环路复杂性。

**控制流图如下，McCabe环路复杂性为3。**



（2） 用基本路径覆盖法给出测试路径。

**测试路径：Path1： ①→③ Path2： ①→②→④→⑥→… Path3： ①→②→⑤→⑥→…**

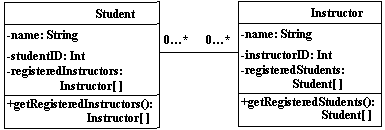
（3） 为各测试路径设计测试用例。

**测试用例： Path1： 取n=1，datalist[0] = 1， 预期结果：k=0**

**Path2： 取n=2，datalist[0] = 1，datalist[1] = 0， 预期结果：k=0**

**Path3： 取n=2，datalist[0] = 0，datalist[1] = 1， 预期结果：k=1**

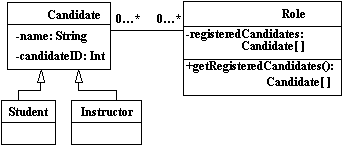
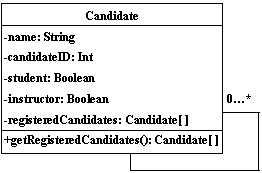
5下图显示了某个学校课程管理系统的部分类图，其中一个学生（student）可以知道所有注册课程的教师（instructor），一个教师也可以知道所有注册课程的学生。



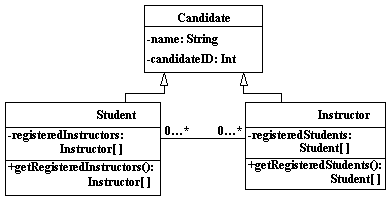
       现在提出一个新的需求："一个教师也可以是某些课程的学生"，那么下面设计A－C中哪一个是最好的？为什么？

**设计B是最好的。学生和教员均可以从Candidate类继承而来；抽象出Role类，使Candidate类与Role类之间形成多对多的关联关系，实现了"一个人既是教师又是某门课的学生"这个需求。**

       设计A：  设计B：



       设计C：



５、根据下面给出的规格说明，利用等价类划分的方法，给出足够的测试用例。

　　"一个程序读入3个整数，它们分别代表一个三角形的3个边长。该程序判断所输入的整数是否构成一个三角形，以及该三角形是一般的、等腰的或等边的，并将结果打印出来。"

要求：设三角形的3条边分别为A、B、C，并且

（1） 列出等价类表，格式如下：

|  |  |  |
| --- | --- | --- |
| 输入条件 | 有效等价类 | 无效等价类 |
| 是否构成一个三角形 | （1）A＞0且B＞0且C＞0且 A＋B＞C且B＋C＞A且 A＋C＞B。 | （2）A≤0或B≤0或C≤0 （3）A＋B≤C或A＋C≤B 或B＋C≤A |
| 是否等腰三角形 | （4）A＝B或A＝C或B＝C | （5）A≠B且A≠C且B≠C |
| 是否等边三角形 | （6）A＝B且A＝C且B＝C | （7）A≠B或A≠C或B≠C |

（2） 设计测试用例，格式如下：

用例n：输入【A，B，C】覆盖等价类……（列出等价类序号），输出结果为……。

**用例1：输入【3，4，5】覆盖等价类（1，2，3，4，5，6），输出结果为构成一般三角形。**

**用例2：三者取一输入【0，1，2】覆盖等价类（2），输出结果为不构成三角形。**

**输入【1，0，2】覆盖等价类（2），输出结果为不构成三角形。**

**输入【1，2，0】覆盖等价类（2），输出结果为不构成三角形。**

**用例3：三者取一输入【1，2，3】覆盖等价类（3），输出结果为不构成三角形。**

**输入【1，3，2】覆盖等价类（3），输出结果为不构成三角形。**

**输入【3，1，2】覆盖等价类（3），输出结果为不构成三角形。**

**用例4：三者取一输入【3，3，4】覆盖等价类（1）（4），输出结果为等腰三角形。**

**输入【3，4，4】覆盖等价类（1）（4），输出结果为等腰三角形。**

**输入【3，4，3】覆盖等价类（1）（4），输出结果为等腰三角形。**

**用例5：输入【3，4，5】覆盖等价类（1）（5），输出结果为不是等腰三角形。**

**用例6：输入【3，3，3】覆盖等价类（1）（6），输出结果为等边三角形。**

**用例7：三者取一输入【3，4，4】覆盖等价类（1）（4）（7），输出结果为不是等边三角形。**

**输入【3，4，3】覆盖等价类（1）（4）（7），输出结果为不是等边三角形。**

**输入【3，3，4】覆盖等价类（1）（4）（7），输出结果为不是等边三角形。**

4.（共20分）下面是一段插入排序的程序，将R[k+1]插入到R[1…k]的适当位置。

　　R[0] = R[k+1];

j = k;

　　while (R[j] > R[0]){

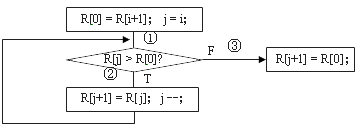
　　　　R[j+1] = R[j];

　　　　j--;}

　　R[j+1] = R[0];

用路径覆盖方法为它设计足够的测试用例（while循环次数为0、1、2次）。

画出该程序的流程图：



测试用例设计

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 循环 次数 | 输 入 数 据 | | | | |  | 预 期 结 果 | | | | | 覆 盖 路 径 | |
| j | R[i-2] | R[i-1] | R[i] | R[i+1] | R[0] | j | R[i-2] | R[i-1] | R[i] | R[i+1] | 约束 | 路 径 |
| 0 | i | - | - | 1 | 2 | 2 | i | - | - | 1 | 2 | < | ①③ |
|  | i | - | - | 1 | 1 | 1 | i | - | - | 1 | 1 | = | ①③ |
| 1 | i | - | 1 | 3 | 2 | 2 | i-1 | - | 1 | 2 | 3 | >< | ①②③ |
|  | i | - | 2 | 3 | 2 | 2 | i-1 | - | 1 | 2 | 3 | >= | ①②③ |
| 2 | i | 1 | 3 | 4 | 2 | 2 | i-2 | 1 | 2 | 3 | 4 | >>< | ①②②③ |
|  | i | 2 | 3 | 4 | 2 | 2 | i-2 | 2 | 2 | 3 | 4 | >>= | 1. ②③ |

6首先说出什么是基本途径测试，然后为以下程序流程图设计基本途径测试的测试用例，并标明路径。

**基本途径测试是指覆盖基本途径集合的试验用例将使程序中的每条语句至少执行一次。**

**测试用例如下：x=3，y=0，z=3（覆盖x>2，y=0，x=3，z>1，通过路径abcde）；**

**x=1，y=1，z=1（覆盖x≤1，y≠0，x≠2，z≤1，通过路径ace）。**

