

# 全国计算机技术与软件专业技术资格（水平）考试

## 2006 年下半年 程序员 下午试卷

（考试时间 14:00~16:30 共 150 分钟）

### 请按下述要求正确填写答题纸

1. 在答题纸的指定位置填写你所在的省、自治区、直辖市、计划单列市的名称。
2. 在答题纸的指定位置填写准考证号、出生年月日和姓名。
3. 答题纸上除填写上述内容外只能写解答。
4. 本试卷共 8 道题，试题一至试题三是必答题，试题四至试题五选答 1 道，试题六至试题八选答 1 道。每题 15 分，满分 75 分。

试题号	一~三	四~五	六~八
选择方法	必答题	选答 1 题	选答 1 题

5. 解答时字迹务必清楚，字迹不清时，将不评分。
6. 仿照下面例题，将解答写在答题纸的对应栏内。

### 例题

2006 年下半年全国计算机技术与软件专业技术资格（水平）考试日期是 (1) 月 (2) 日。

因为正确的解答是“11 月 4 日”，故在答题纸的对应栏内写上“11”和“4”（参看下表）。

例题	解答栏
(1)	11
(2)	4

**试题一（共 15 分）**

阅读以下说明和算法，完善算法并回答问题，将解答写在答题纸的对应栏内。

**[说明]**

假设以二维数组  $G[1..m, 1..n]$  表示一幅图像各像素的颜色，则  $G[i, j]$  表示区域中点  $(i, j)$  处的颜色，颜色值为 0 到  $k$  的整数。

下面的算法将指定点  $(i_0, j_0)$  所在的同色邻接区域的颜色置换为给定的颜色值。约定所有与点  $(i_0, j_0)$  同色的上、下、左、右可连通的点组成同色邻接区域。

例如，一幅  $8 \times 9$  像素的图像如图 1-1 所示。设用户指定点  $(3, 5)$ ，其颜色值为 0，此时其上方  $(2, 5)$ 、下方  $(4, 5)$ 、右方  $(3, 6)$  邻接点的颜色值都为 0，因此这些点属于点  $(3, 5)$  所在的同色邻接区域，再从上、下、左、右四个方向进行扩展，可得出该同色邻接区域的其他点（见图 1-1 中的阴影部分）。将上述同色区域的颜色替换为颜色值 7 所得的新图像如图 1-2 所示。

	1	2	3	4	5	6	7	8	9
1	5	4	5	4	3	1	5	1	2
2	2	5	5	3	0	1	3	2	1
3	0	3	2	3	0	0	2	3	1
4	2	0	1	0	0	0	0	2	0
5	1	0	0	0	0	3	2	0	1
6	0	1	0	2	0	0	2	2	1
7	6	5	5	0	1	0	2	1	0
8	6	3	3	4	0	0	7	4	5

图 1-1

	1	2	3	4	5	6	7	8	9
1	5	4	5	4	3	1	5	1	2
2	2	5	5	3	7	1	3	2	1
3	0	3	2	3	7	7	2	3	1
4	2	7	1	7	7	7	7	2	0
5	1	7	7	7	7	3	2	0	1
6	0	1	7	2	7	7	2	2	1
7	6	5	5	0	1	7	2	1	0
8	6	3	3	4	7	7	7	4	5

图 1-2

**[算法]**

输入：矩阵  $G$ ，点的坐标  $(i_0, j_0)$ ，新颜色值  $newcolor$ 。

输出：点  $(i_0, j_0)$  所在同色邻接区域的颜色置换为  $newcolor$  之后的矩阵  $G$ 。

算法步骤（为规范算法，规定该算法只在第七步后结束）：

- 第一步：若点  $(i_0, j_0)$  的颜色值与新颜色值  $newcolor$  相同，则\_\_\_\_(1)\_\_\_\_；
- 第二步：点  $(i_0, j_0)$  的颜色值  $\rightarrow oldcolor$ ；创建栈  $S$ ，并将点坐标  $(i_0, j_0)$  入栈；
- 第三步：若\_\_\_\_(2)\_\_\_\_，则转第七步；
- 第四步：栈顶元素出栈  $\rightarrow (x, y)$ ，并\_\_\_\_(3)\_\_\_\_；
- 第五步：
  - 1) 若点  $(x, y-1)$  在图像中且  $G[x, y-1]$  等于  $oldcolor$ ，则  $(x, y-1)$  入栈  $S$ ；
  - 2) 若点  $(x, y+1)$  在图像中且  $G[x, y+1]$  等于  $oldcolor$ ，则  $(x, y+1)$  入栈  $S$ ；
  - 3) 若点  $(x-1, y)$  在图像中且  $G[x-1, y]$  等于  $oldcolor$ ，则  $(x-1, y)$  入栈  $S$ ；
  - 4) 若点  $(x+1, y)$  在图像中且  $G[x+1, y]$  等于  $oldcolor$ ，则  $(x+1, y)$  入栈  $S$ ；
- 第六步：转\_\_\_\_(4)\_\_\_\_；
- 第七步：算法结束。

**[问题]**

是否可以将算法中的栈换成队列？回答：\_\_\_\_(5)\_\_\_\_。

## 试题二（共 15 分）

阅读以下说明和 C 函数，将应填入     (n) 处的字句写在答题纸的对应栏内。

### [说明]

某单位准备进行一次选举，对指定的 n 名 (n<80) 候选人进行投票。为了简化选举工作，事先印制了写有这 n 名候选人姓名的选票，投票者只需将选中者名下的圆圈涂黑即可。规定每张选票上被涂黑的圆圈数不得超过 3 个，也不得少于 1 个，否则视为无效选票。投票结束后，所有选票经计算机扫描处理，形成了描述选票状态的文本文件。例如，n=8 时所形成的文件如下：

```
01011000
10010100
10011000
11111111
00000000
00111000
...
```

其中，每行表示一张选票的数据，每列代表一位候选者的得票情况。第 i 行第 j 列为 1，表示第 i 张选票上投了第 j 名候选人 1 票。

函数 statistic( ) 的功能是读入选票数据，并统计每位候选者的得票数，函数返回有效选票数。

### [C 函数]

```
int statistic(FILE *fp, int candidate[], int n)
{ /* 从文件中读入选票数据，n 为候选人数量 (n<80)，从有效选票中统计每位候选者的得票数并存入 candidate[]，函数返回有效选票数 */
```

```
    char str[80]; /* str 保存读入的一张选票数据 */
    int i, tag = 0; /* tag 用于判断每张选票是否有效 */
    int q = 0; /* q 用于计算有效选票数 */

    for(i = 0; i < n; i++)
        candidate[i] = 0;

    while (____(1)____) {
        fgets(str, 80, fp); /* 读入一张选票数据 */
        for(tag = 0, i = 0; _____(2)____; i++)
            if (str[i] == '1') tag++;

        if (____(3)____) { /* 若是有效选票，则进行统计 */
            _____(4)____;
            for(i = 0; i < n; i++)
                if (str[i] == '1') _____(5)____;
        }
    }

    return q;
} /* statistic */
```

**试题三（共 15 分）**

阅读以下说明和 C 函数，将应填入     (n)     处的字句写在答题纸的对应栏内。

**[说明]**

若一个矩阵中的非零元素数目很少且分布没有规律，则称之为稀疏矩阵。对于 m 行 n 列的稀疏矩阵 M，进行转置运算后得到 n 行 m 列的矩阵 MT，如图 3-1 所示。

$$M_{4 \times 5} = \begin{bmatrix} 0 & -3 & 0 & 0 & 5 \\ 0 & 0 & 0 & 10 & 0 \\ 12 & 0 & 0 & 0 & 0 \\ 0 & 14 & 0 & 0 & -7 \end{bmatrix} \quad MT_{5 \times 4} = \begin{bmatrix} 0 & 0 & 12 & 0 \\ -3 & 0 & 0 & 14 \\ 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 5 & 0 & 0 & -7 \end{bmatrix}$$

图 3-1 稀疏矩阵 M 及其转置矩阵 MT

为了压缩稀疏矩阵的存储空间，用三元组（即元素所在的行号、列号和元素值）表示稀疏矩阵中的一个非零元素，再用一维数组逐行存储稀疏矩阵中的所有非零元素（也称为三元组顺序表）。例如，图 3-1 所示的矩阵 M 相应的三元组顺序表如表 3-1 所示，其转置矩阵 MT 的三元组顺序表如表 3-2 所示。

表 3-1

矩阵 M		
行号	列号	元素值
0	1	-3
0	4	5
1	3	10
2	0	12
3	1	14
3	4	-7

表 3-2

M 的转置矩阵 MT		
行号	列号	元素值
0	2	12
1	0	-3
1	3	14
3	1	10
4	0	5
4	3	-7

函数 TransposeMatrix(Matrix M) 的功能是对用三元组顺序表表示的稀疏矩阵 M 进行转置运算。

对 M 实施转置运算时，为了将 M 中的每个非零元素直接存入其转置矩阵 MT 三元组顺序表的相应位置，需先计算 M 中每一列非零元素的数目（即 MT 中每一行非零元素的数目），并记录在向量 num 中；然后根据以下关系，计算出矩阵 M 中每列的第一个非零元素在转置矩阵 MT 三元组顺序表中的位置：

```
cpot[0] = 0
cpot[j] = cpot[j-1] + num[j-1]    /* j 为列号 */
```

类型 ElemType、Triple 和 Matrix 定义如下：

```
typedef int ElemType;
typedef struct { /* 三元组类型 */
    int r, c; /* 矩阵元素的行号、列号 */
    ElemType e; /* 矩阵元素的值 */
}Triple;
typedef struct { /* 矩阵的三元组顺序表存储结构 */
    int rows, cols, elements; /* 矩阵的行数、列数和非零元素数目 */
    Triple data[MAXSIZE];
}Matrix;
```

### [C 函数]

```
int TransposeMatrix(Matrix M)
{
    int j, q, t;
    int *num, *cpot;
    Matrix MT;    /* MT 是 M 的转置矩阵 */
    num = (int *)malloc(M.cols*sizeof(int));
    cpot = (int *)malloc(M.cols*sizeof(int));
    if (!num || !cpot)
        return ERROR;
    MT.rows = (1);    /* 设置转置矩阵 MT 行数、列数和非零元数目 */
    MT.cols = (2);
    MT.elements = M.elements;
    if (M.elements > 0) {

        for(q = 0; q < M.cols; q++)
            num[q] = 0;

        for(t = 0; t < M.elements; ++t) /* 计算矩阵 M 中每一列非零元素数目 */
            num[M.data[t].c]++;

        /* 计算矩阵 M 中每列第一个非零元素在其转置矩阵三元组顺序表中的位置 */
        (3);
        for(j = 1; j < M.cols; j++)
            cpot[j] = (4);

        /* 以下代码完成转置矩阵 MT 三元组顺序表元素的设置 */
        for(t = 0; t < M.elements; t++) {
            j = (5);    /* 取矩阵 M 的一个非零元素的列号存入 j */

            /* q 为该非零元素在转置矩阵 MT 三元组顺序表中的位置（下标）*/
            q = cpot[j];
            MT.data[q].r = M.data[t].c;
            MT.data[q].c = M.data[t].r;
            MT.data[q].e = M.data[t].e;
            ++cpot[j];    /* 计算 M 中第 j 列的下一个非零元素的目的位置 */
        }/* for */

    }/* if */
    free(num); free(cpot);

    /*此处输出矩阵元素，代码省略*/

    return OK;
}/* TransposeMatrix */
```

从下列的 2 道试题（试题四至试题五）中任选 1 道解答。  
如果解答的试题数超过 1 道，则题号小的 1 道解答有效。

#### 试题四（共 15 分）

阅读以下应用说明以及用 Visual Basic 编写的程序代码，将应填入     (n)     处的字句写在答题纸的对应栏内。

##### [应用 4.1]

设应用程序的运行窗口内有一个文字标签 (Label) 以及一个框架，其中有三个复选框 (chk1, chk2, chk3)，各个复选框单击事件过程的程序代码如下：

```
Private Sub chk1_Click()  
    Label.FontBold = chk1.Value  
End Sub  
Private Sub chk2_Click()  
    Label.FontItalic = chk2.Value  
End Sub  
Private Sub chk3_Click()  
    Label.FontUnderline = chk3.Value  
End Sub
```

三个复选框 chk1、chk2、chk3 的功能分别是：     (1)    。

##### [应用 4.2]

设应用程序的运行窗口内有两个文本框 Txt1 和 Txt2，其初始内容为空。在 Txt1 文本框中输入一个数值，当光标离开此文本框（例如进入文本框 Txt2）时，执行的程序代码如下：

```
Private Sub Txt1_LostFocus()  
    dim x as double  
    x = Val(Txt1.Text)  
    If x < 0 Or x > 100 Then  
        Txt1.Text = ""  
        MsgBox$ ("请重新输入!")  
        Txt1.SetFocus  
    Else  
        Txt2.Text = Txt1.Text  
    End If  
End Sub
```

该程序代码的功能是：若在文本框 Txt1 中输入的数值小于 0 或大于 100，当光标离开此文本框时，     (2)    ；否则，将其值复制到文本框 Txt2 中。

##### [应用 4.3]

在下面的应用中，当窗口内发生 Click 事件时，窗口内将显示如图 4-1 所示的杨辉三

角形（每一行都是二项式展开的系数）。请完善程序代码。

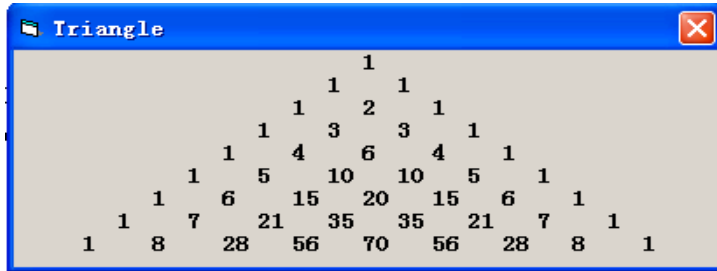


图 4-1

```
Private Sub Form_Click()  
    Dim i, j, c As Integer, StrTemp As String  
    Dim a(9) As Integer  
    a(0) = 0: a(1) = 1: StrTemp = Str(a(1)) + Space(3)  
    CurrentX = (ScaleWidth - TextWidth(StrTemp)) / 2  
    Print StrTemp  
    For j = 2 To 9  
        a(j) = 1  
        For c = j-1 To 2 Step -1  
            a(c) = (3)  
        Next  
        (4) = ""  
        For c = 1 To j  
            StrTemp = StrTemp & Str((5)) & Space(5 - Len(Str(a(c))))  
        Next  
        CurrentX = (ScaleWidth - TextWidth(StrTemp)) / 2  
        Print StrTemp  
    Next  
End Sub
```

### 试题五 (共 15 分)

阅读以下说明和 C 函数，将应填入     (n)     处的字句写在答题纸的对应栏内。

#### [说明]

某班级有 N 名学生，他们可根据自己的情况选修名称和数量不尽相同的课程。设 N 等于 6，学生信息、所选课程及成绩用链表结构存储，如图 5-1 所示。

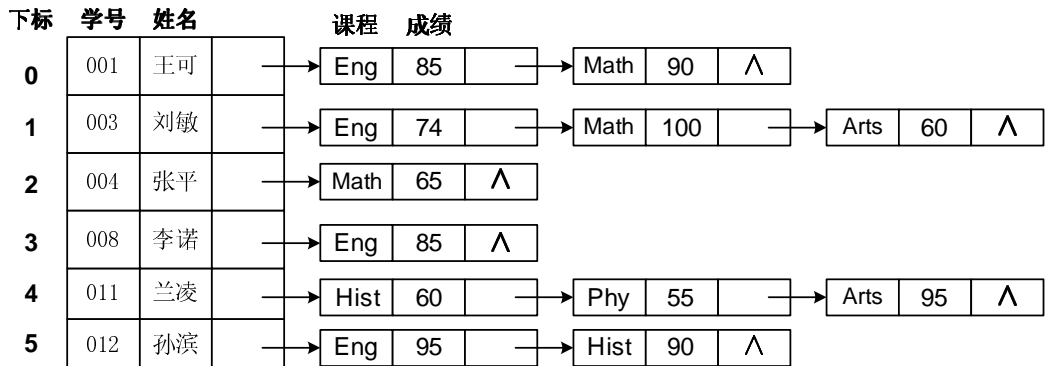


图 5-1

程序中相应的类型定义如下：

```
#define N 6
struct node{
    char cname[5];        /*课程名*/
    int grade;           /*成绩*/
    struct node *next;    /*指针，指示某学生选修的下一门课程及成绩*/
};

struct student{
    char xh[5];          /*学号*/
    char name[20];       /*姓名*/
    struct node *link;   /*指针，指示出选修的课程及成绩链表*/
}stud_info[N];
```

stud\_info[] 为一个全局数组。

函数 func (char kc[], int \*num) 的功能是统计选修了课程名为 kc 的学生的人数，并返回该课程的平均成绩 (若无人选修该课程，则平均成绩为 0)，参数 num 带回选修课程 kc 的学生人数。

**[C 函数]**

```
double func (char kc[], int *num)
{
    int i, count = 0, sum = 0; /*count 用于记录选修课程名为 kc 的学生的人数*/
    double avg = 0.0;
    struct node *p;

    for(i = 0; i < N; i++){
        p = _____(1)_____;          /*取第 i 个学生所修课程链表的头指针*/
        while (p) {
            if (_____(2)_____) {
                sum = _____(3)_____;
                count++;
                break;;
            }/*if*/
            p = p->next;
        }/*while*/
    }
    _____(4)_____;
    if (_____(5)_____)
        avg = (double)sum / count;          /* 计算平均成绩 */
    return avg;
}/*func*/
```

从下列的 3 道试题（试题六至试题八）中任选 1 道解答。  
如果解答的试题数超过 1 道，则题号小的 1 道解答有效。

### 试题六（共 15 分）

阅读以下说明和 C++ 程序代码，将应填入\_\_ (n) \_\_处的字句写在答题纸的对应栏内。

#### [说明]

在下面的 C++ 代码中，类 SalesTicket 能够完成打印票据正文的功能，类 HeadDecorator 与 FootDecorator 分别完成打印票据的台头和脚注的功能。

已知该程序运行后的输出结果如下所示，请填补该程序代码中的空缺。

这是票据的台头！

这是票据正文！

这是票据的脚注！

-----

这是票据的台头！

这是票据的脚注！

#### [C++ 程序代码]

```
#include <iostream>
using namespace std;

class SalesTicket {
public:
    _____(1)_____ printTicket() { cout << "这是票据正文!" << endl;}
};

class Decorator : public SalesTicket{
    SalesTicket *ticket;
public:
    Decorator(SalesTicket *t){ ticket = t; }
    void printTicket() {
        if(ticket != NULL)
            ticket->printTicket();
    }
};

class HeadDecorator : public Decorator{
public:
    HeadDecorator(SalesTicket *t): _____(2)_____ { }
    void printTicket() {
```

```

        cout << "这是票据的台头!" << endl;
        Decorator::printTicket();
    }
};

class FootDecorator :public Decorator{
public:
    FootDecorator(SalesTicket *t):__(3)_{ }
    void printTicket() {
        Decorator::printTicket();
        cout << "这是票据的脚注!" << endl;
    }
};

void main(void) {
    SalesTicket t;
    FootDecorator f(&t);
    HeadDecorator h(__(4));
    h.printTicket();
    cout << "-----" << endl;
    FootDecorator a(NULL);
    HeadDecorator b(__(5));
    b.printTicket();
}

```

### 试题七（共 15 分）

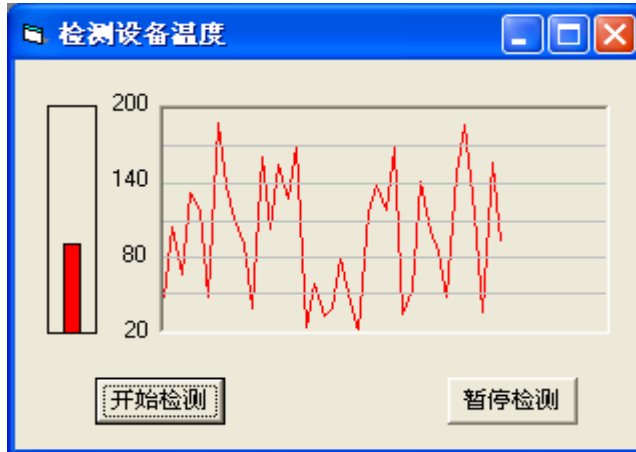
阅读以下应用说明以及用 Visual Basic 编写的程序代码，将应填入\_\_\_\_(n)\_\_\_\_处的字句写在答题纸的对应栏内。

#### [应用说明]

某应用程序用于监测某种设备的工作温度（20~200 度），其运行窗口中，包括一个温度计（矩形形状 shpMeter）以及其中指示当前设备温度的水银柱（矩形形状 shpT），文字标签标记了温度刻度；另有一个图片框 picCurve，用于动态描述检测到的温度曲线（用户见到的曲线与水银柱等高变化）；命令按钮“开始检测”（cmdStart）用于启动温度检测，命令按钮“暂停检测”（cmdStop）用于暂停检测。

矩形形状 shpT(水银柱)属性 visible 初始设置为不可见，属性 Filltype 设置为 solid(实心),FillColor 设置为红色;图片框 picCurve 的属性 AutoRedraw 设置为 True;再创建一个定时器 TimT,属性 Enabled 初始设置为 False(不起作用),属性 Interval(定时间隔)设置为 500 毫秒。

为模拟设备温度的检测，程序中利用了(0, 1)之间均匀分布的伪随机数获得[20, 200]之间的随机温度 T。为了便于在图片框 picCurve 中绘制曲线，程序中对该图片框建立了坐标系统，左上角为原点 (0, 0)，水平向右方向为 X 轴，垂直向下方向为 Y 轴，右下角坐标为 (50, 180)。为了便于观察记录的溫度值，图片框中从上到下创建了 5 条水平线  $L_s(i), i=0, 1, \dots, 4$ ，并在程序中按等间隔排列进行位置设置。程序中每隔半秒算出曲线点 (x, y)，其中  $x=0, 1, 2, \dots$ ，再用直线段连接各相邻曲线点形成温度曲线。



#### [Visual Basic 程序代码]

```
Dim ____ (1) ____ As Integer          ' 声明全局变量
Private Sub CmdStart_Click( )
    TimT.Enabled = True
    ShpT.Visible = True
End Sub
```

```

Private Sub CmdStop_Click( )
    TimT.Enabled = False
End Sub

Private Sub Form_Load( )
    Dim i, H As Integer
    PicCurve.Scale (0, 0)-(50, 180) ' 设置图片框坐标系: 左上角-右下角
    H = 30 ' H 等于图片框高度的六分之一
    For i = 0 To 4 ' 设置 5 条水平线 Ls(i) 的位置
        Ls(i).X1 = 0 ' Ls(i) 起点横坐标
        Ls(i).Y1 = H *         (2)         ' Ls(i) 起点纵坐标
        Ls(i).X2 = 50 ' Ls(i) 终点横坐标
        Ls(i).Y2 = Ls(i).Y1 ' Ls(i) 终点纵坐标
        Ls(i).BorderColor = &HCOCOCO ' 设置水平线颜色
    Next i
    x = 0 ' 设置曲线坐标初值
End Sub

Private Sub timT_Timer()
    Dim T, H As Integer ' T 为即时温度, H 为图片框中温度点显示高度
    T = Int(Rnd * 181) + 20 ' 模拟随机产生设备温度 (20~200 度)
    ' 按当前温度显示水银柱
    H = ShpMeter.Height *         (3)         ' 算出水银柱的高度
    ShpT.Top =         (4)         - H ' 设置水银柱顶部位置
    ShpT.Height = H ' 设置水银柱的高度
    ' 绘制温度曲线
    y =         (5)         ' 算出曲线上当前点的纵坐标
    If x = 51 Then ' 当超出图片框时
        PicCurve.Cls ' 清除图片框内以前画的曲线
        x = 0 ' 设置重画曲线的初值
    ElseIf x > 0 Then ' 除左边点外
        PicCurve.Line (x-1, Lasty)-(x, y), vbRed ' 由前 1 点到当前点画红色线段
    End If
    x = x + 1 ' 准备下一点坐标
    Lasty = y ' 保存当前坐标供下次使用
End Sub

```

### 试题八（共 15 分）

阅读以下说明和 Java 程序代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

#### [说明]

在下面的 Java 程序代码中，类 SalesTicket 能够完成打印票据正文的功能，类 HeadDecorator 与 FootDecorator 分别完成打印票据的台头和脚注的功能。

已知该程序运行后的输出结果如下所示，请填补该程序代码中的空缺。

这是票据的台头！

这是票据正文！

这是票据的脚注！

-----  
这是票据的台头！

这是票据的脚注！

#### [Java 程序代码]

```
public class SalesTicket {
    public void printTicket() {
        System.out.println("这是票据正文!");
    }
}

public class Decorator extends SalesTicket{
    SalesTicket ticket;
    public Decorator(SalesTicket t) {
        ticket = t;
    }
    public void printTicket() {
        if(ticket != null)
            ticket.printTicket();
    }
}

public class HeadDecorator extends Decorator{
    public HeadDecorator(SalesTicket t) {
        _____(1)_____;
    }
    public void printTicket() {
        System.out.println("这是票据的台头! ");
        super.printTicket();
    }
}
```

```
public class FootDecorator extends Decorator{
    public FootDecorator(SalesTicket t) {
        _____(2)_____;
    }
    public void printTicket() {
        super.printTicket();
        System.out.println("这是票据的脚注！");
    }
}
public class Main {
    public static void main(String[] args) {
        T = new HeadDecorator(____(3)____);
        T.____(4)____;
        System.out.println("-----");
        T = new FootDecorator(____(5)____);
        T.printTicket();
    }
}
```