

2005年上半年软件设计师下午试题

试题一至试题四是必答题

试题一（15分）

阅读以下说明和数据流图，回答问题1至问题3，将解答题纸的对应栏内。

[说明]

学生住宿服务系统帮助学生在就学的城市内找以所需的住房，系统对出租的房屋信息、房主信息、需要租房的学生信息以及学生和房主的会面住处进行管理和维护。

房主信息包括姓名、地址、电话号码以及系统分配的唯一身份标识（ID）和密码；房屋信息包括房屋地址、类型（单间/套间）、适合住宿的人数、房租、房主的ID以及现在是否可以出租（例如由于装修原因，需等到装修后才可出租或者房屋已被租出）。每当房屋信息发生变化时，房主须通知系统，系统将更新房到文件以便学生能够获得准确的可租用房屋信息，房主向系统中加入可租用的房屋信息时，需交纳一定的费用，由系统自动给出费用信息。房主可随时更新房屋的各种属性。

学生可通过系统查询现有的可租用的房屋，但必须先在系统中注册。学生信息包括姓名、现住址、电话号码、出生日期、性别以及系统分配的唯一身份标识（ID）和密码。若学生希望租用某房屋，则需要发出租房请求，请求中包含房屋的详细信息，系统将安排学生与房主会面的时间和地点，并将会面信息通知学生和房主，会面信息包括会面时间、地点以及会面双方的基本信息，系统将记录会面信息。

学生住宿服务系统的顶层图如图1-1所示；学生住宿服务系统的第0层DFD图如图1-2所示，其中，加工3的细化图如图1-3所示。

[问题1]（6分）

(1) 数据流图1-1缺少了一条数据流（在图1-2中也未给出该数据流），请给出此数据流的起点和终点，并采用说明中的词汇给出此数据流名。

(2) 数据流图1-2中缺少了与“查询房屋”加工相关的数据流，请指出此数据流的起点和终点。

[问题2]（4分）

“安排会面”加工除需要写入会面文件外，还需要访问哪些文件？

[问题3]（5分）

请补齐下列数据字典条目：

登录信息=学生 ID+密码

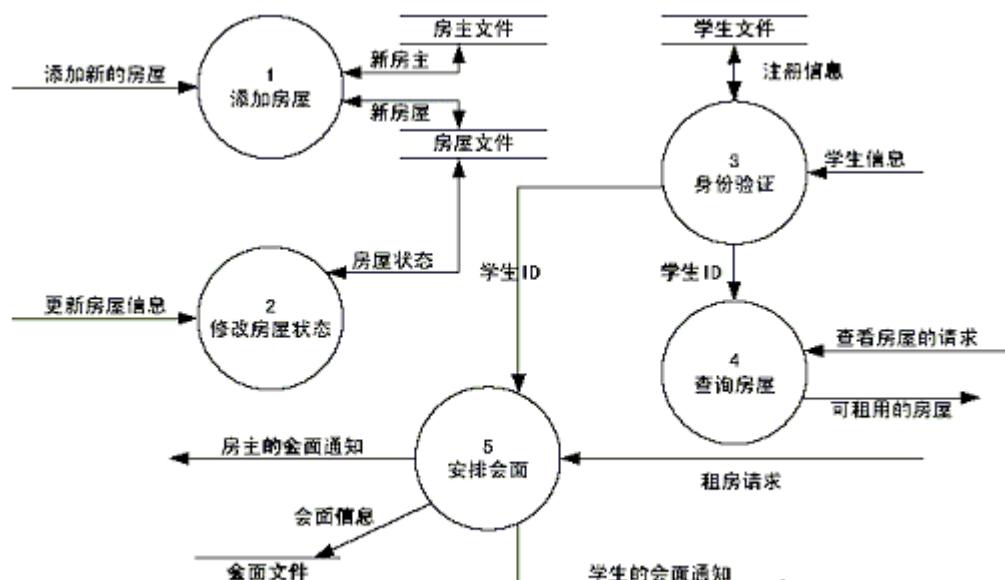
注册信息=

[数据流图 1-1]



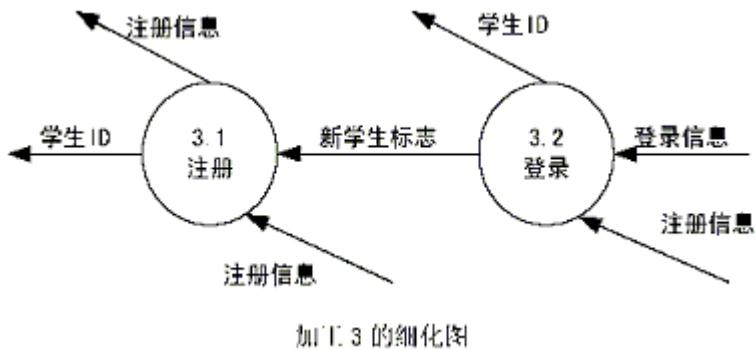
学生住宿服务顶层图

[数据流图 1-2]



学生住宿服务第 0 层 DFD 图

[数据流图 1-3]



试题二（15分）

阅读以下说明和表，回答问题1至问题4，将解答填入答题纸的对应栏内。

[说明]

某公司信息管理系统的部分关系模式设计结果描述如下：

1. 公司有多个部门，每个部门有一名负责人、一间办公室、一部电话、多名职员，每个职员最多数一个部门，负责人也是公司一名职员。
2. 公司职员的工资大于等于1000元且最小等于8000。
3. 数据库的部分关系模式设计如下：
 职员（职员号，职工姓名，月工资。部门号、办公室，电话）
 部门（部门号，部门名，负责人代码，任职时间）
 4.“职员”和“部门”的关系示例分别如表2-1和表2-2所示

[表2-1]

“职员”关系

职员号	职员姓名	月工资	部门号	办公室	电话
60801	汪俊华	1000	1	A座201	6883122
60802	杨晓军	3200	1	A座201	6883122
60803	王晓华	4300	2	B座202	6883123
60804	邢彦军	2800	2	B座202	6883123
60805	吕靖原	5300	3	A座301	6883124
60806	芦文峰	3200	3	A座301	6883124
60807	牟雪松	2800	3	A座301	6883124
60808	高亚南	1200	4	B座302	6883125

60810	周黎	3200	4	B 座 302	6883125
60820	姚应磊	1200	4	B 座 302	6883125
60821	程文驰	3200	5	B 座 303	6883126
60836	许俊坤	0	Nu11

[表 2-2]

"部门"关系

部门号	部门名	负责人代码	任职时间
1	财务部	60802	2001-8-5
2	市场部	60803	2002-6-3
3	研发部	60805	2002-6-3
4	生产部 1	60810	2003-8-1
5	生产部 2	60821	2004-6-3

[问题 1] (4 分)

根据上述说明，请给出

- (1) "职员"关系模式的主键和外键。
- (2) "部门"关系模式的主键和外键。

[问题 2] (4 分)

- (1) 用 SQL 定义"职员"关系模式，请在空缺处填入正确的内容。

```
Create Table 职员 (职员号 CHAR (5) __(a)__ ,
```

职员姓名 CHAR (8),

月工资 NUMBER (4),

部门号 CHAR (1),

办公室 CHAR (20)

电话 CHAR (8),

__(b)__(部门号),

CHECK (月工资 \geq 1000 AND 月工资 \leq 8000));

(2)针对人数大于等于 2 的部门创建视图 D_View(Dept, D_num, D_Totals, D_Avgpay),

其中, Dept 为部门号, D_num 为部门人数, D_Totals

为部门人数, D_Avgpay 为平均工资, 请在空缺处填入正确的内容。

```
Create View D_View (Dept, D_num, D_Totals, D_AvgPay) As
  (Select 部门号, __(c)__
   from 职员
   __(d)__ count (*) >=2 WHERE 部门号 IS NOT NULL):
```

[问题 3] (3 分)

对于表 2-1、表 2-2 所示的"职员"和"部门"关系, 请指出下列各行是否可以插入"职员"关系, 为什么?

1	60811	芦 峰	800	1	A 座 201	6883122
2	60802	李晓啸	3500	2	B 座 202	6883123
3	60812	高亚南	2600			

[问题 4] (4 分)

原来的"职员"关系模式存在什么问题? 在不增加新关系模式的前提下, 请给出修出改后的"职员"和"关系模式"。

试题三 (15 分)

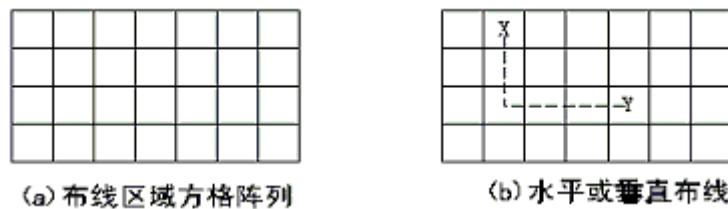
阅读以下说明和流程图, 从供选择的答案中选出应填入流程图 __(n)__ 处的字句写在答题纸的对应栏内。

[说明]

一个印刷电路板的布线区域可分成 $n \times m$ 个方格, 如图 3-1 (a) 所示, 现在需要确定电路板中给定的两个方格的中心点之间的最短布线方案。电路只能沿水平或垂直方向布线, 如图 3-1 (b) 中虚线所示。为了避免线路相交, 应将已布过线的方格作成封锁标记, 其他线

路不允许穿过被封锁的方格。

[图 3-1]



设给定印刷电路板的起始方格 x 与目的方格 y 尚未布线, 求这两个方格间最短布线方案的基本思路是: 从起始方格 x 开始, 先考查距离起始方格距离为 k 的某一个可达方格就是目标方格 y 时为止, 或者由于不存在从 x 到 y 的布线方案而终止。布线区域中的每一个方格与其相邻的上、下、左、右四个方格之间的距离为 1, 依次沿下、右、上、左这四个方向考查, 并用一个队列记录可达方格的位置。表 3-1 给出了沿这四个方向前进 1 步时相对于当前方格的相对偏移量。

[表 3-1]

搜索顺序 i	方向	行偏移量	列偏移量
0	上	-1	0
1	右	0	1
2	上	-1	0
3	左	0	-1

例如, 设印刷电路板的布线区域可划分为一个 6×8 的方格阵列, 如图 3-2 (a) 所示, 其中阴影表示已封锁方格。从起始方格 x (位置 [3, 2], 标记为 0) 出发, 按照下、右、上、左的方向依次考查, 所标记的可达方格如图 3-2 (a) 所示, 目标方格为 y (位置 [4, 7], 标记为 10), 相应的最短布线路径如图 3-2 (b) 虚线所示。

[图 3-2]

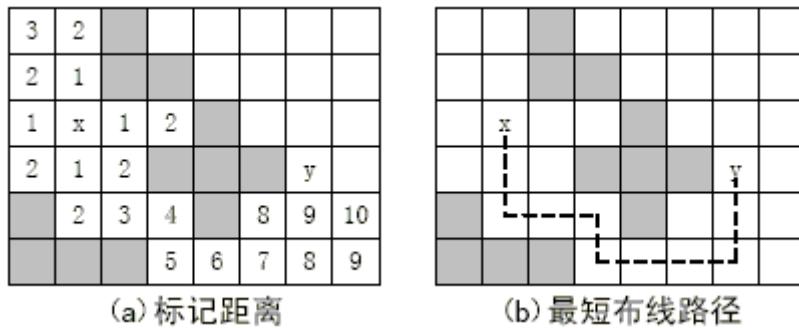
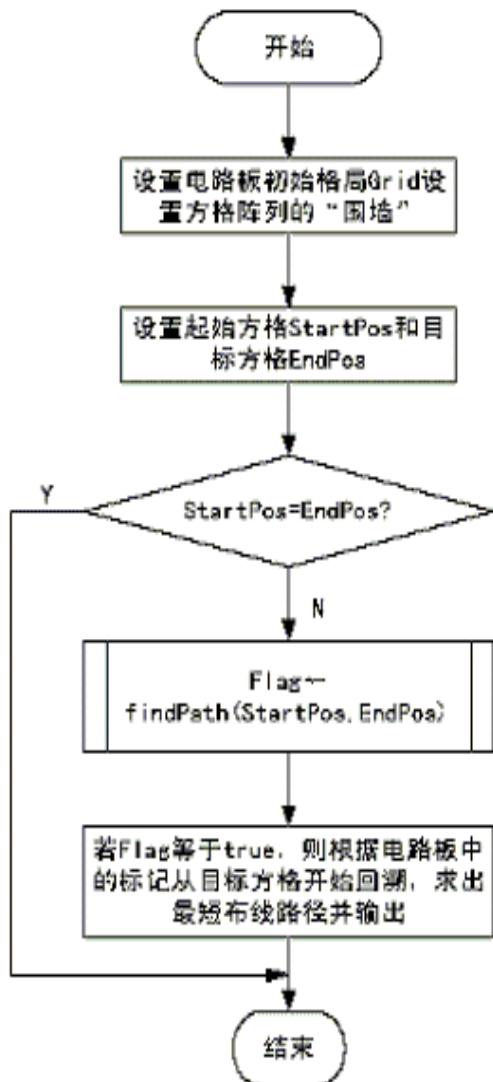


图 3-3 和图 3-4 所示的流程图即利用上述思想路，在电路板方格阵列中进行标记，图中使用的主要符号如表 3-2 所示。在图 3-4 中，设置电路板初始格局即将可布线方格置为数值-1、已布线方格（即封锁方格）置为-9。设置方格阵列“围墙”的目的是省略方格位置的边界条件判定，方法是在四周附加格，并将其标记为-9（与封锁标记相同）。

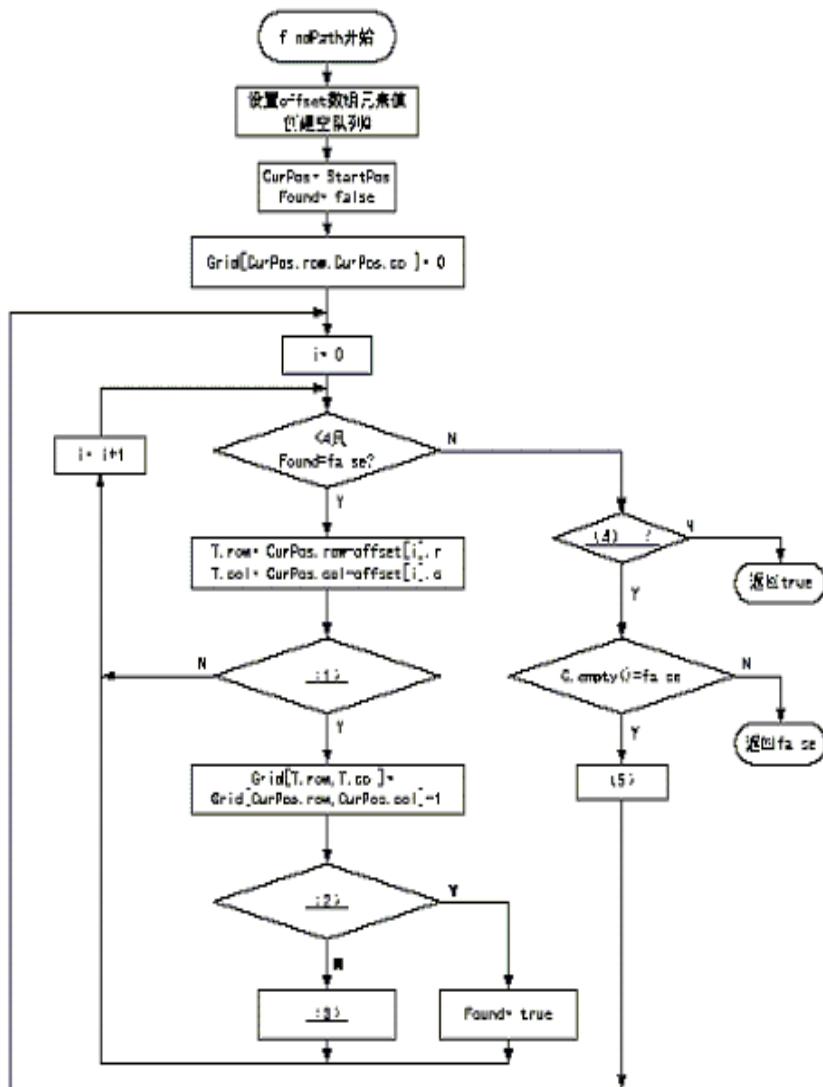
[表 3-2]

符号	含义
Grid	全局二维数组 Grid[N+2,M+2], 表示电路板方格阵列，初始时数组元素 Grid[i,j] 的值为 -1 表示当前方格可布线，为 -9 表示前方格不可布线。
Offset	一维数组 offset[4]:offset[i](0≤i≤3) 的分量为 r (行偏移量) 和 c(列偏移量)，按照表 3-1 的内容设置其值。
Startpos、 Endpos、 Curpos、T	分别表示起始方格、目标方格、当前方格和临时方格，其位置用分量度 row 和 col 确定。
Q.insert(s)	将方格 s 的位置信息加入队列
Q.delete()	删除非空队列的队头元素，并返回该元素。
Q.empty()	若队列 Q 为空，则返回 true；否则返回 false。

[图 3-3]



[图 3-4]



●供选择的答案

- | | |
|-----------------------------|-----------------------------|
| [a] Found ≠ true | [b] Found = true |
| [c] T = EndPos | [d] Q.insert(T) |
| [e] T ← Q.delete() | [f] CurPos = EndPos |
| [g] i ≥ 4 | [h] CurPos ← Q.delete() |
| [i] Grid[T.row, T.col] = -1 | [j] Grid[T.row, T.col] ≠ -1 |

试题四（15 分）

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

[说明]

假设需要将 N 个任务分配给 N 个工人同时去完成，每个人都能承担这 N 个任务，但费用不同。下面的程序用回溯法计算总费用最小的一种工作分配方案，在该方案中，为每个人本试题及答案仅供参考，疏漏之处不断改进

分配 1 个不同的任务。

程序中，N 个任务从 0 开始依次编号，N 个工人也从 0 开始依次编号，主要的变量说明如下：

c[i][j]:将任务 i 分配给工人 j 的费用；

task[i]:值为 0 表示任务 i 未分配，值为 j 表示任务 i 分配给工人 j；

worker[k]:值为 0 表示工人 k 未分配任务，值为 1 表示工人 k 已分配任务；

mincost:最小总费用。

[C 程序]

```
# include <stdio.h>

#define N 8           /*N 表示任务数和工人数*/

int c[N][N];

unsigned int mincost = 65535;    /*设置的初始值，大于可能的费用*/

int task[N], temp[N], worker[N];

void plan(int k,unsigned int cost)

{ int i;

  if ( __(1)__ && cost<mincost) {

    mincost = cost;

    for(i=0; i<N; i++) temp[i]=task[i]; }

  else{

    for(i=0; i<N; i++)          /*分配任务 k*/

    if (worker[i]==0 && __(2)__){

      worker[i]=1; task[k]=__(3__);

      Plan(__(4__), cost+c[k][i]);

      __(5__); task[k] = 0;

    }/*if*/}

}

/*Plan*/



void main()

{ int i,j;
```

```

for(i=0; i<N; i++){           /*设置每个人任务由不同工人承担时的费用及全局数组的
初值*/
    worker[i]=0; task[i]=0; temp[i]=0;
    for(j=0; j<n; j++)
        scanf("%d",&c[i][j]);
}
Plan(0, 0); /*从任务 0 开始分配*/
printf("\n 最小差用=%d\n", mincost);
for(i=0; i<N; i++)
printf("Task% is assigned to Worker%d\n",i,temp[i]);
}           /*main*/

```

从下列的 3 道试题（试题五至试题七）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。

试题五（15 分）

阅读以下说明和 C++ 码，将应填入 __(n)___ 处的字名写在答题纸的对应栏内。

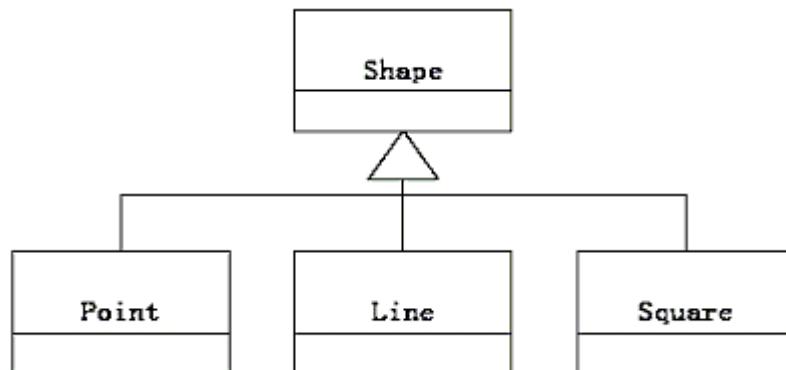
[说明]

某绘图系统存在 Point、Line、Square 三种图元，它们具有 Shape 接口，图元的类图关系如图 5-1 所示。现要将 Circle 图元加入此绘图系统以实现功能扩充。已知某第三方库已经提供了 XCircle 不是由 Shape 派生而来，它提供的接口不被系统直接使用。代码 5-1 既使用了 XCircle 又遵循了 Shape 规定的接口，即避免了从头开发一个新的 Circle 类，又可以不修改绘图系统中已经定义的接口。代码 5-2 根据用户指定的参数生成特定的图元实例，并对之进行显示操作。

绘图系统定义的接口与 XCircle 提供的显示接口及其功能如下表所示：

Shape	XCircle	功能
display()	DisplayIt()	显示图元

[图 5-1]



[代码 5-1]

```
class Circle: public __(1)__{
    Private:
        __(2)__ m_circle;
    public:
        void display () {
            __(3)__
        }
};
```

[代码 5-2]

```
class Factory{
    public:
        __(4)__ getShapeInstance(int type){      //生成特定类实例
            switch(type){
                case 0:return new Point;
                case 1:return new Rectangle;
                case 2:return new Line;
                case 3: return new Circle;
                default : return NULL;
            }
}
```

```

    }

}

};

void main(int argc,char *argv[]){
    if(argc !=2){
        cout<<"error parameters ! "<<endl;
        return;
    }
    int type=atoi(argv[1]);
    Factory factory;
    Shape *s;
    S=factory._(5);
    if(s==NULL){
        cout<<"Error get the instance!"<<endl;
        return;
    }
    s->display();
    _(6) ;
    return;
}

```

试题六（15 分）

阅读以下说明和 Java 代码，将应填入 _(n)_ 处的字名写在答题纸的对应栏内。

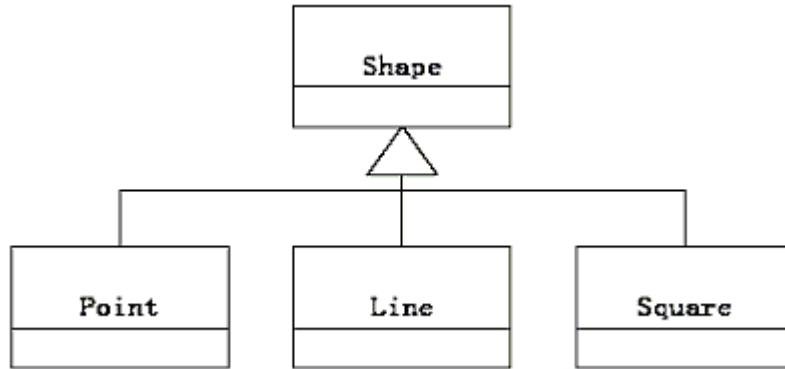
[说明]

某绘图系统存在 Point、Line、Square 三种图元，它们具有 Shape 接口，图元的类图关系如图 6-1 所示。现要将 Circle 图元加入此绘图系统以实现功能扩充。已知某第三方库已经提供了 XCircle 类，且完全满足系统新增的 Circle 图元所需的功能，但 XCircle 又遵循了 Shape 规定的接口。既避免了从头开发一个新的 Circle 类，又可以不修改绘图系统中已经定义的接口，代码 6-2 根据用户指定的参数生成特定的图元实例，并对之进行显示操作。

绘图系统定义的接口与 XCircle 提供的显示接口及其功能如下表所示：

Shape	XCircle	功能
display()	displayit()	显示图元

[图 6-1]



[代码 6-1]

```

class Circle {
    private pxc;
    public Circle () { pxc = new (3);
    }
    public void display(){
        pxc. (4);
    }
}
  
```

[代码 6-2]

```

public class Factory{
    public (5) getShapeInstance(int type){ //生成特定类实例
        Switch(type){
            case 0: return new point();
            case 1: return new Rectangle();
            case 2: return new Line();
            case 3: return new Circle();
        }
    }
}
  
```

```
    default: return null;
}
};

public class App{
    public static void main(String argv[]){
        if(argv.length !=1){
            System.out.println("error parameters!");
            return;
        }
        int type=(new Integer(argv[0])).intValue();
        Factory factory=new Factory();
        if(s==null){
            System.out.println("Error get instance!");
            Return;
        }
        s.display();
        return;
    }
}
```

试题七（15 分钟）

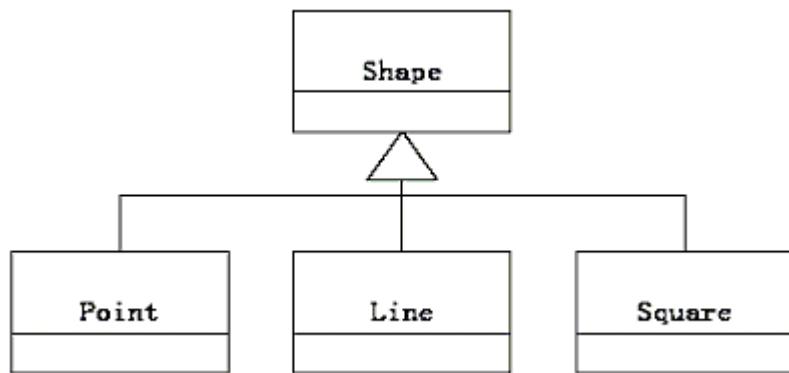
阅读以下说明和 Visual Basic 代码，将应填入 __(n)___ 处的字名写在答题纸的对应栏内。

[说明]

某绘图系统定义了一个抽象类 IShape，现有三个类 CPoint，CLine 和 CCircle，它们都具有 IShape 界面。相应的类图关系如图 7-1 所示。

已知某第三方库已经提供了 XCircle 类，且完全满足 CCircle 图元显示时所需的功能。代码 7-1 是抽象类 IShape 类模块内容，代码 7-2 实现了类 CCircle 的 IShape 界面，并使用了 XCircle 提供的显示功能。XCircle 提供的显示功能方法接口为 displayIt。

[图 7-1]



[代码 7-1]

```
Public Color As Long
```

```
Sub draw()
```

'方法体不包括可执行语句

```
End Sub
```

```
Sub move(stepx As Single, stepy As Single)
```

'方法体不包括可执行语句

```
End Sub
```

[代码 7-2]

```
__(1)___
```

```
Private color As Long
```

... '其他定义省略

```
Private ridged As __(2)___
```

```
Private Sub Class_Initialize()
```

```
    Set bridged= __(3)___
```

```
End Sub
```

Private Property __(4)__ ()As Long

IShape_Color=color

End Property

Private Property __(5)__ (ByVal newColor As Long)

Color=newColor

End Property

Private Sub IShape_draw() '使用 XCircle 提供的显示功能

__(6)___

End Sub

Private Sub IShape_move (stepx As Single, stepy As Single)

... '省略描述

End Sub