2017 年下半年软件设计师下午真题及答案解析

试题一 (15 分)

某公司拟开发一个共享单车系统，采用北斗定位系统进行单车定位，提供针对用户的 APP 以及微信小程序、基于 Web 的管理与监控系统。该共享单车系统的主要功能如下。

1）用户注册登录。用户在 APP 端输入手机号并获取验证码后进行注册，将用户信息进行存储。用户登录后显示用户所在位置周围的单车。

2）使用单车。

①扫码/手动开锁。通过扫描二维码或手动输入编码获取开锁密码，系统发送开锁指令进行开锁，系统修改单车状态，新建单车行程。

②骑行单车。单车定时上传位置，更新行程。

③锁车结账。用户停止使用或手动锁车并结束行程后，系统根据已设置好的计费规则及使用时间自动结算，更新本次骑行的费用并显示给用户，用户确认支付后，记录行程的支付状态。系统还将重置单车的开锁密码和单车状态。

3）辅助管理。

①查询。用户可以查看行程列表和行程详细信息。

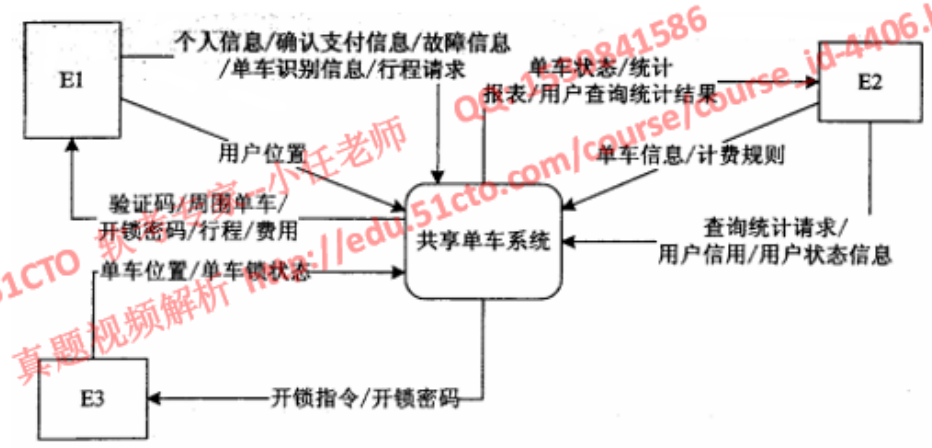
②报修。用户上报所在位置或单车位置以及单车故障信息并进行记录。

4）管理与监控。

①单车管理及计费规则设置。商家对单车基础信息、状态等进行管理，对计费规则进行设置并存储。

②单车监控。对单车、故障、行程等进行查询统计。

③用户管理。管理用户信用与状态信息，对用户进行查询统计。 现采用结构化方法对共享单车系统进行分析与设计，获得如图 1-1 所示的上下文数据流图和图 1-2 所示的 0 层数据流图。



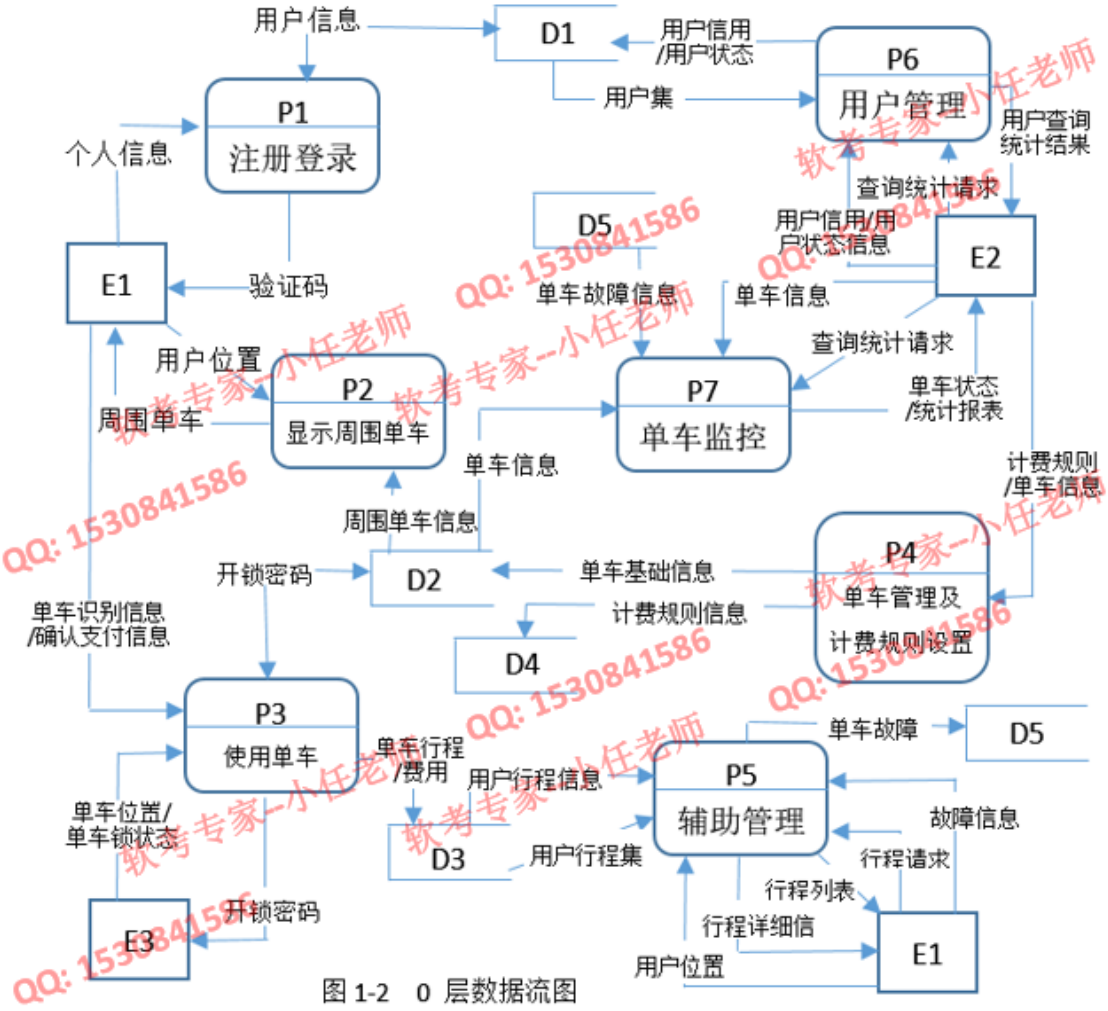


图 1-2　0 层数据流图

【问题 1】（3 分）
使用说明中的词语，给出图 1-1 中的实体 E1~E3 的名称。

【问题 2】（5 分）
使用说明中的词语，给出图 1-2 中的数据存储 D1~D5 的名称。

【问题 3】（5 分）
根据说明和图中术语及符号，补充图 1-2 中缺失的数据流及其起点和终点。

【问题 4】（2 分）
根据说明中术语，说明“使用单车”可以分解为哪些子加工？

试题二（15 分）
M 公司为了便于开展和管理各项业务活动，提高公司的知名度和影响力，拟构建一个基于网络的会议策划系统。
【需求分析结果】
该系统的部分功能及初步需求分析的结果如下 ：

（1）M 公司旗下有业务部、策划部和其他部门。部门信息包括部门号、部门名、主管、联系电话和邮箱号；每个部门只有一名主管，只负责管理本部门的工作，且主管参照员工关系的员工号；一个部门有多名员工，每名员工属于且仅属于一个部门。

（2）员工信息包括员工号、姓名、职位、联系方式和薪资。职位包括主管、业务员、 策划员等。业务员负责受理用户申请，设置受理标志。一名业务员可以受理多个用户申请，但一个用户申请只能由一名业务员受理。

（3 用户信息包括用户号、用户名、银行账号、电话、联系地址。用户号唯一标识用户信息中的每一个元组。

（4）用户申请信息包括申请号、用户号、会议日期、天数、参会人数、地点、预算和受理标志。申请号唯一标识用户申请信息中的每一个元组，且一个用户可以提交多个申请，但一个用户申请只对应一个用户号。

（5）策划部主管为已受理的用户申请制定会议策划任务。策划任务包括申请号、任务明细和要求完成时间。申请号唯一标识策划任务的每一个元组。一个策划任务只对应一个已受理的用户申请，但一个策划任务可由多名策划员参与执行，且一名策划员可以参与执行，且在项策划任务。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图（不完整）如图 2-1 所示。

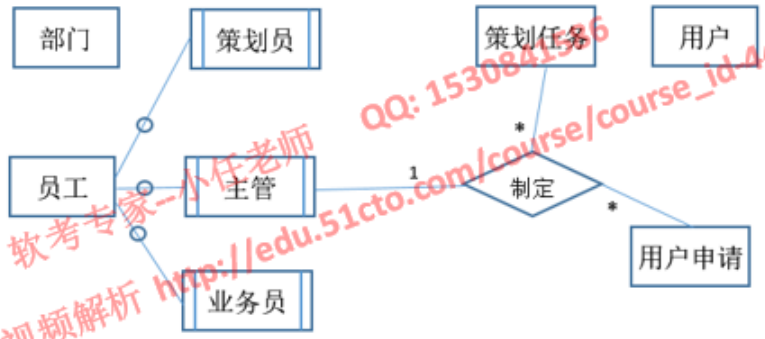


图 2-1 实体联系图

【关系模型设计】

部门（部门号，部门名，部门主管，联系电话，邮箱号）

员工（员工号，姓名，（ 1 ），联系方式，薪资）

用户（用户名，（ 2 ），电话，联系地址）

用户申请（申请号，用户号，会议日期，天数，参会人数，地点，受理标志，（ 3 ））

执行（申请号，任务明细，（ 4 ））

【问题 1】（5 分）

根据问题描述，补充五个联系，完善图 2-1 的实体联系图。联系名可用联系 1、联系 2、联系 3、联系 4 和联系 5，联系的类型为 1:1、1:n 和 m:n（或 1:1、1:*和*:*）。

【问题 2】（4 分）

根据题意，将关系模型中的空（a）~（d）补充完整，并填入答题纸对应的位置上。

【问题 3】（4 分）

给出“用户申请”和“策划任务”关系模式的主键和外键。

【问题 4】（2 分）
请问“执行”关系模式的主键为全码的说法正确吗？为什么？

试题三 (15 分)

某大学拟开发一个用于管理学术出版物（Publication）的数字图书馆系统(System)，用户可以从该系统查询或下载己发表的学术出版物。系统的主要功能如下：

1.登录系统。系统的用户(User)仅限于该大学的学生(Student)、教师(Faculty)和其他工作人员(Staff)。在访问系统之前，用户必须使用其校园账户和密码登录系统。

2. 查询某位作者(Author)的所有出版物。系统中保存了会议文章(ConfPaper)、期刊文章(JournalArticle)和校内技术报告(TechReport)等学术出版物的信息，如题目、作者以及出版年份等。除此之外，系统还存储了不同类型出版物的一些特有信息：

（1）对于会议文章，系统还记录了会议名称、召开时间以及召开地点；

（2）对于期刊文章，系统还记录了期刊名称、出版月份、期号以及主办单位；

（3）对于校内技术报告，系统记录了由学校分配的唯一 ID。

3. 查询指定会议集(Proceedings)或某个期刊特定期(Edition)的所有文章。会议集包含了发表在该会议（在某个特定时间段、特定地点召开）上的所有文章。期刊的每一期在特定时间发行，其中包含若干篇文章。

4.下载出版物。系统记录每个出版物被下载的次数。

5.查询引用了某篇出版物的所有出版物。在学术出版物中引用他人或早期的文献作为相关工作或背景资料是很常见的现象。用户也可以在系统中为某篇出版物注册引用通知，若有新的出版物引用了该出版物，系统将发送电子邮件通知该用户。

现在采用面向对象方法对该系统进行开发，得到系统的初始设计类图如图 3-1 所示。

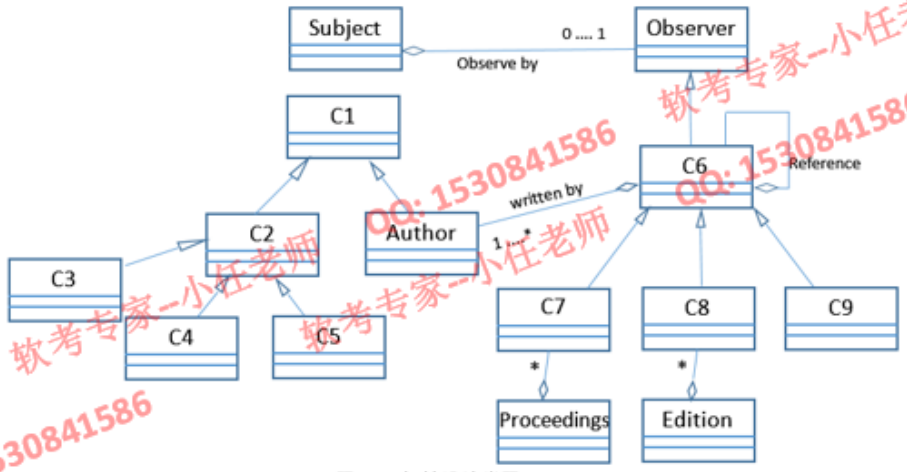


图 3-1 初始设计类图

【问题 1】（9 分）
根据说明中的描述，给出图 3-1 中 C1~C9 所对应的类名。

【问题 2】（4 分）
根据说明中的描述，给出图 3-1 中类 C6~C9 的属性。

【问题 3】（2 分）
图 3-1 中包含了哪种设计模式？实现的是该系统的哪个功能？

试题四 (15 分)

一个无向连通图 G 点上的哈密尔顿(Hamiltion)回路是指从图 G 上的某个顶点出发，经过图上所有其他顶点一次且仅一次，最后回到该顶点的路径。一种求解无向图上哈密尔顿回路算法的基础思路如下：

假设图 G 存在一个从顶点 V_0 出发的哈密尔顿回路 V_1--V_2--V_3--..--V_{n-1}--V_0。算法从顶点 V_0 出发，访问该顶点的一个未被访问的邻接顶点 V_1，接着从顶点 V_1 出发，访问 V_1 一个未被访问的邻接顶点 V_2，....；对顶点 V_i，重复进行以下操作：访问 V_i 的一个未被访问的邻接接点 V_{i+1}；若 V_i 的所有邻接顶点均已被访问，则返回到顶点 V_{i-1}，考虑 V_{i-1} 的下一个未被访问的邻接顶点，仍记为 V_i；直到找到一条哈密尔顿回路或者找不到哈密尔顿回路，算法结束。

【C 代码】

下面是算法的 C 语言实现。

（1）常量和变量说明

n ：图 G 中的顶点数

c[][]：图 G 的邻接矩阵

K：统计变量，当期已经访问的顶点数为 k+1

x[k]：第 k 个访问的顶点编号，从 0 开始

Visited[x[k]]：第 k 个顶点的访问标志，0 表示未访问，1 表示已访问

⑵C 程序

```
#include <stido.h>
#include <stidb.h>
#define MAX 100

Void Hamilton(int n,int x,MAX,int c[MAX][MAX]) {
int i ;
int visited[MAX];
int k;
 /*初始化 x 数组和 visited 数组*/
for (i=0:i<n;i++) {
   x[i]=0;
   visited [i]=0;
}
/*访问起始顶点*/
k=0
（  1  ）;
x[0]=0
K=k+1
/*访问其他顶点*/
while(k>=0) {
   x[k]=x[k]+1;
   while(x[k]<n) {
 if （  2  ）&& c[x-[k-1]][x[k]]==1）{   /*邻接顶点 x[k]未被访问过*/
    break ;
    } else {
```

```
    x[k] = x[k] +1
    }
  }
  if（x[k] <n-1&&（ 3    ）{ /*找到一条哈密尔顿回路*/
    for(k=0;k<n;k++) {
      prinf("%d--",x[k] ); /*输出哈密尔顿回路*/
    }
    prinf("%d--",x[0]) ;
    return ;
  } else if ( x[k]<n&& k<n-1)  {/*设置当期顶点的访问标志，继续下一个顶点*/
    （ 4   ）;
    k=k+1;
  } else { /*没有未被访问过的邻接顶点，回退到上一个顶点*/
     x[k]=0 ;
     visited x[k]=0 ;
     （  5  ）;
  }
}
}
```

【问题 1】（10 分）

根据题干说明。填充 C 代码中的空（1）~（5）

【问题 2】（5 分）

根据题干说明和 C 代码，算法采用的设计策略为（6），该方法在遍历图的顶点时，采用的是（7）方法（深度优先或广度优先）。

试题五 (15 分)

某图像预览程序要求能够查看 BMP 、JPEG 和 GIF 三种格式的文件，且能够 Windows 和 Linux 两种操作系统上运行.程序需具有较好的扩展性以支持新的文件格式和操作系统。为满足上述需求并减少所需生成的子类数目，现采用桥接（Bridge）模式进行设计，得到如图 5-1 所示的类图。

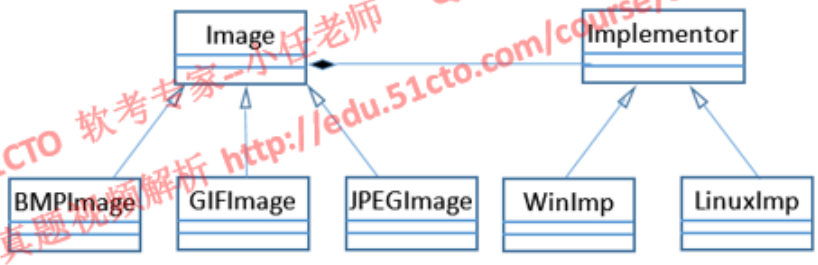


图 5-1 类图

【C++代码】

```
#include <iostream>
#include<:string> ;
```

```
Using namespace std;

class Matrix｛//各种格式的文件最终都被转化为像素矩阵
    //此处代码省略
};
class Implementor ｛
    public:
    （ 1 ）； //显示像素矩阵 m
} ;
class WinImp:public Implementor ｛
public:
   Viod doPaint（Matrix m）｛  /*调用 Windows 系统的绘制函数绘制像素矩阵*/
} ;
class LinuxImp:public Implementor ｛/*调用 Linux 系统的绘制函数绘制像素矩阵*/｝
} ;
class Image ｛
public:
   viod setImp（Implement*imp）? ｛this->imp=imp;｝
   Virtual viod parseFile（string fileName）=0
   protected:
   Implementor*imp;
} ;
class BMPImage:public Image ｛
      //此处省略代码
} ;
class GIFImage:public Image ｛
Public:
    viod   parseFile（string fileName）  ｛
       //此处解析 GIF 文件并获得一个像素矩阵对象 m
      （  2 ）；   //显示像素矩阵 m
     }
} ;
class JPEGImage:public Image ｛
   //此处代码省略
} ;

int main（）｛
   //在 linux 操作系统上查看 demo.gif 图像文件
   Image*image=（  3 ）；
   Implementor*imageImp=（ 4 ）；
   （ 5 ）
   Image->parseFile（"demo.gif"）；
   return 0;
}
```

试题六 (15 分)

某图像预览程序要求能够查看 BMP 、JPEG 和 GIF 三种格式的文件，且能够在 Windows 和 Linux 两种操作系统上运行。程序需具有较好的扩展性以支持新的文件格式和操作系统。为满足上述需求并减少所需生成的子类数目，现采用桥接模式进行设计，得到如图 6-1 所示的类图。

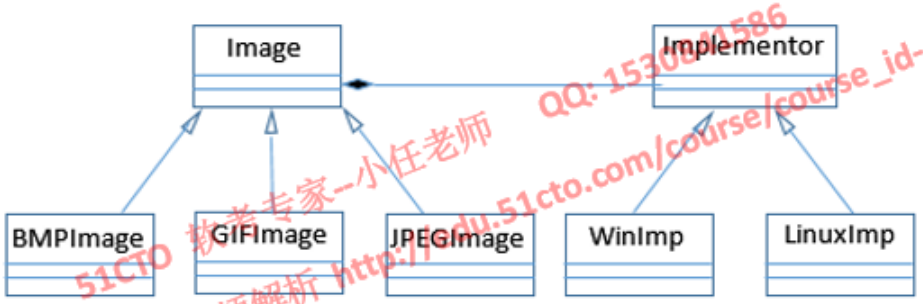


图 6-1 类图

【Jave 代码】

```
Import java.util.*;

class Matrix{ //各种格式的文件最终都被转化为像素矩阵
//此处代码省略
};
abstract class Implementor {
   Public（   1   ）;  //显示像素矩阵 m
};
class WinImp extends Implementor {
   public void doPaint（Matrix m）{ //调用 Windows 系统的绘制函数绘制像素矩阵
   }
};
class LinuxImp extends Implementor{
   public void doPaint（Matrix m）{    //调用 Linux 系统的绘制函数绘制像素矩阵
   }
};
abstract class Image {
   public void setImp（Implementor imp）{ this.imp= imp; }
   public abstract void parseFile（String fileName）;
   protected Implementor imp;
} ;
class BMPImage extends Image {
     //此处代码省略
};
class GIFImage extends Image {
    public void parseFile（String fileName）  {
      //此处解析 BMP 文件并获得一个像素矩阵对象 m
```

```
    （  2 ）；//显示像素矩阵 m
   }
};
Class Main {
    Public static viod main（String[]args）{
     //在 Linux 操作系统上查看 demo.gif 图像文件
      Image image=（  3  ）
      Implementor imageImp=（ 4 ）
      （  5 ）
        Image.parseFile（"demo.gif"）；
      }
}
```

2017 年下半年软件设计师下午答案及解析

试题一

问题 1　　单击此链接可看真题解析视频 http://edu.51cto.com/course/course_id-4406.html

E1:用户　　E2:商家　　E3:单车

问题 2

D1: 用户信息表　D2: 单车信息表　　D3:行程信息表

D4: 单车计费规则信息表　　D5: 单车故障信息表　　（或文件均可）

问题 3

起点：P3　　终点：E1　　数据流名称：开锁密码

起点：P3　　终点：E1　　数据流名称：费用

起点：P3　　终点：E3　　数据流名称：开锁指令

起点：P3　　终点：D2　　数据流名称：单车状态

起点：D4　　终点：P3　　数据流名称：计费规则

起点：D3　　终点：P7　　数据流名称：行程信息

问题 4

可以分解为：

1.扫码/手动开锁子加工

2.骑行单车子加工

3.锁车结账子加工

试题二

问题 1

1.联系 1：部门和员工，1:n

2.联系 2：业务员和用户申请，1:n

3.联系 3：用户和用户申请，1:n

4.联系 4：策划员和策划任务，m:n

5.联系 5：策划任务和用户申请，1:1

问题 2

1. 职位，部门号

2. 用户号，银行帐号

3. 预算费用，员工号

4. 要求完成时间

问题 3

用户申请：主键：申请号； 外键：用户号，员工号

策划任务：主键：申请号； 外键：申请号

问题 4

策划任务的属性包括申请号、任务明细和要求完成时间，只有申请号是候选码，而任务明细

和要求完成时间为非主属性，即候选码不包括全部属性，所以不是全码。
解析：。
若关系中只有一个候选码，且这个候选码中包含全部属性，则该候选码为全码。
若关系中的一个属性或属性组的值能够唯一地标识一个元组，且他的真子集不能唯一的标识一个元组，则称这个属性或属性组做候选码

试题三

问题 1

C1:system	C2:User	C3:Student
C4:Faculty	C5:Staff	C6:Publication
C7:ConfPaper	C8:JournalArticle	C9:TechReport

问题 2
C6:题目，作者，出版年份，出版物类型，下载次数，作者电子邮箱
C7:会议名称，召开时间，召开地点
C8:期刊名称，出版月份，期号，主办单位
C9:ID

问题 3
观察者模式，实现：引用他人学术出版物发送电子邮件通知该用户。
策略模式，实现：查询作者的三类不同出版物：会议文章，期刊文章，校内技术报告。

试题四

问题 1
1. visited[0] = 1
2. visited[x[k]] == 0
3. visited[x[k]] == 1
4. visited[x[k]] = 1
5. k = k - 1

问题 2：
6.回溯法
7.深度优先

试题五

1.virtual void doPaint(Matrix m) = 0
2.imp->doPaint(m)
3.new GIFImage()
4.new LinuxImp()
5.image->setImp(imageImp)

试题六

1.abstract void doPaint(Matrix m)
2.imp.doPaint(m)
3.new GIFImage()

4.new LinuxImp()

5.image.setImp(imageImp)

一、小任老师软件设计师视频

1、软件设计师基础知识视频精讲 http://edu.51cto.com/course/course_id-4033.html

2、软件设计师上午历年真题解析视频 http://edu.51cto.com/course/course_id-5827.html

3、软件设计师下午历年真题解析视频 http://edu.51cto.com/course/course_id-4406.html

二、小任老师高级系统分析师视频课程

1、系统分析师综合知识视频课程 http://edu.51cto.com/course/course_id-2422.html

2、系统分析师下午案例视频课程 http://edu.51cto.com/course/course_id-2968.html

3、系统分析师论文写作视频课程 http://edu.51cto.com/course/course_id-3069.html