全国计算机技术与软件专业技术资格（水平）考试

2016 年上半年 软件设计师 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 6 道题，试题一至试题四是必答题，试题五至试题六选答 1 道。每题 15 分，满分 75 分。

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

2015 年上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“5 月 20 日”，故在答题纸的对应栏内写上“5”和“20”（参看下表）。

例题	解答栏
（1）	5
（2）	20

试题一至试题四是必答题

试题一（共 15 分）

阅读下列说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】

某会议中心提供举办会议的场地设施和各种设备，供公司与各类组织机构租用。场地包括一个大型报告厅、一个小型报告厅以及诸多会议室。这些报告厅和会议室可提供的设备有投影仪、白板、视频播放/回放设备、计算机等。为了加强管理，该中心欲开发一会议预订系统，系统的主要功能如下。

（1）检查可用性。客户提交预订请求后，检查预订表，判定所申请的场地是否在申请日期内可用；如果不可用，返回不可用信息。

（2）临时预订。会议中心管理员收到客户预定请求的通知之后，提交确认。系统生成新临时预订存入预订表，并对新客户创建一条客户信息记录加以保存。根据客户记录给客户发送临时预订确认信息和支付定金要求。

（3）分配设施与设备。根据临时预订或变更预定的设备和设施需求，分配所需设备（均能满足用户要求）和设施，更新相应的表和预订表。

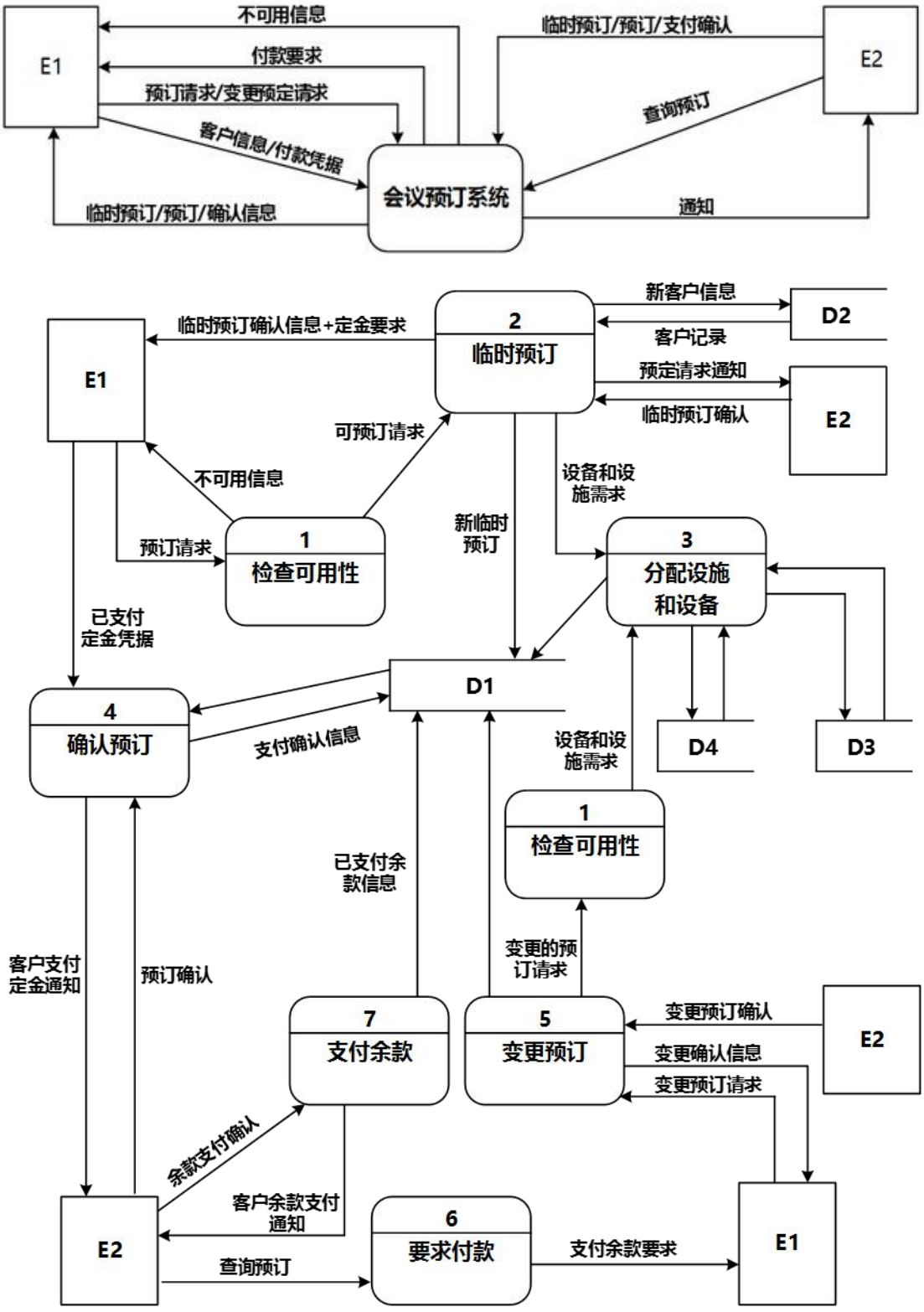
（4）确认预订。管理员收到客户支付定金的通知后，检查确认，更新预订表，根据客户记录给客户发送预订确认信息。

（5）变更预订。客户还可以在支付余款前提交变更预订请求，对变更的预订请求检查可用性，如果可用，分配设施和设各；如果不可用，返回不可用信息。管理员确认变更后，根据客户记录给客户发送确认信息。

（6）要求付款。管理员从预订表中查询距预订的会议时间两周内的预定，根据客户记录给满足条件的客户发送支付余款要求。

（7）支付余款。管理员收到客户余款支付的通知后，检查确认，更新预订表中的已支付余款信息。

现采用结构化方法对会议预定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（不完整）。



【问题 1】（2 分）

使用说明中的词语，给出图 1-1 中的实体 E1～E2 的名称。

【问题 2】（4 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1～D4 的名称。

【问题 3】（6 分）

根据说明和图中术语，补充图 1-2 之中缺失的数据流及其起点和终点。

【问题 4】（3 分）

如果发送给客户的确认信息是通过 Email 系统向客户信息中的电子邮件地址进行发送的，那么需要对图 1-1 和 1-2 进行哪些修改？用 150 字以内文字加以说明。

试题二（共 15 分）

阅读下列说明，回答问题 1 至问题 3；将解答填入答题纸的对应栏内。

【说明】

某销售公司当前的销售业务为商城实体店销售。现该公司拟开展网络销售业务，需要开发一个信息化管理系统。请根据公司现有业务及需求完成该系统的数据库设计。

【需求描述】

（1）记录公司所有员工的信息。员工信息包括工号、身份证号、姓名、性别、出生日期和电话，并只登记一部电话。

（2）记录所有商品的信息。商品信息包括商品名称、生产厂家、销售价格和商品介绍。系统内部用商品条码唯一区别每种商品。

（3）记录所有顾客的信息。顾客信息包括顾客姓名、身份证号、登录名、登录密码、和电话号码。一位顾客只能提供一个电话号码。系统自动生成唯一的顾客编号。

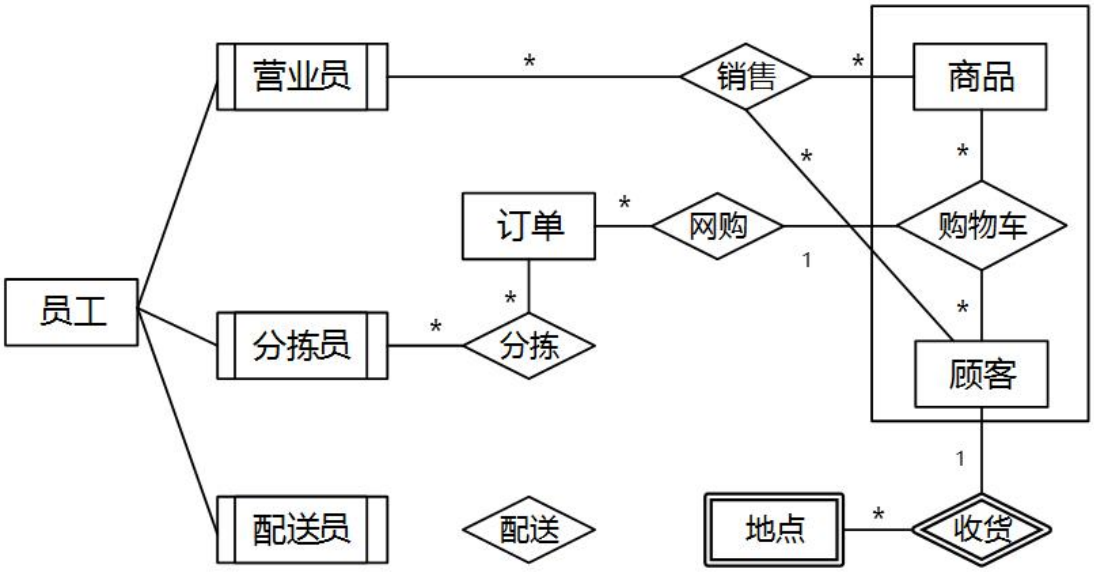
（4）顾客登录系统之后，在网上商城购买商品。顾客可将选购的商品置入虚拟的购物车内，购物车可长期存放顾客选购的所有商品。顾客可在购物车内选择商品、修改商品数量后生成网购订单。订单生成后，由顾客选择系统提供的备选第三方支付平台进行电子支付，支付成功后系统需要记录唯一的支付凭证编号，然后由商城根据订单进行线下配送。

（5）所有的配送商品均由仓库统一出库。为方便顾客，允许每位顾客在系统中提供多组收货地址、收货人及联系电话。一份订单所含的多个商品可能由多名分检员根据商品所在仓库信息从仓库中进行分拣操作，分拣后的商品交由配送员根据配送单上的收货地址进行配送。

（6）新设计的系统要求记录实体店的每笔销售信息，包括营业员、顾客、所售商品及其数量。

【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。



【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式（不完整）：

员工（工号，身份证号，姓名，性别，出生日期，电话）

商品（商品条码，商品名称，生产厂家，销售价格，商品介绍，　（a）　）

顾客（顾客编号，姓名，身份证号，登录名，登录密码，电话）

收货地点（收货 ID，顾客编号，收货地址，收货人，联系电话）

购物车（顾客编号，商品条码，商品数量）

订单（订单 ID，顾客编号，商品条码，商品数量，　（b）　）

分检（分拣 ID，分拣员工号，　（c）　，分拣时间）

配送（配送 ID，分拣 ID，配送员工号，收货 ID，配送时间，签收时间，签收快照）

销售（销售 ID，营业员工号，顾客编号，商品条码，商品数量）

【问题 1】（4 分）

补充图 2-1 中的“配送”联系所关联的对象及联系类型。

【问题 2】（6 分）

补充逻辑结构设计中的（a）、（b）和（c）三处空缺。

【问题 3】（5 分）

对于实体店销售，若要增加送货上门服务，由营业员在系统中下订单，与网购的订单进行后续的统一管理。请根据该需求，对图 2-1 进行补充，并修改订单关系模式。

试题三（共 15 分）

阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某软件公司欲设计实现一个虚拟世界仿真系统。系统中的虚拟世界用于模拟现实世界中的不同环境（由用户设置并创建），用户通过操作仿真系统中的 1~2 个机器人来探索虚拟世界。机器人维护着两个变量 b1 和 b2，用来保存从虚拟世界中读取的字符。

该系统的主要功能描述如下：

（1）机器人探索虚拟世界（Run Robots）。用户使用编辑器（Editor）编写文件以设置想要模拟的环境，将文件导入系统（Load File）从而在仿真系统中建立虚拟世界（Setup World）。机器人在虚拟世界中的行为也在文件中进行定义，建立机器人的探索行为程序（Setup Program）。机器人在虚拟世界中探索时（Run Program），有 2 种运行模式：

①自动控制（Run）：事先编排好机器人的动作序列（指令（Instruction）），执行指令，使机器人可以连续动作。若干条指令构成机器人的指令集（Instruction Set）。

②单步控制（Step）：自动控制方式的一种特殊形式，只执行指定指令中的一个动作。

（2）手动控制机器人（Manipulate Robots）。选定 1 个机器人后（Select Robot），可以采用手动方式控制它。手动控制有 4 种方式：

①Move：机器人朝着正前方移动一个交叉点。

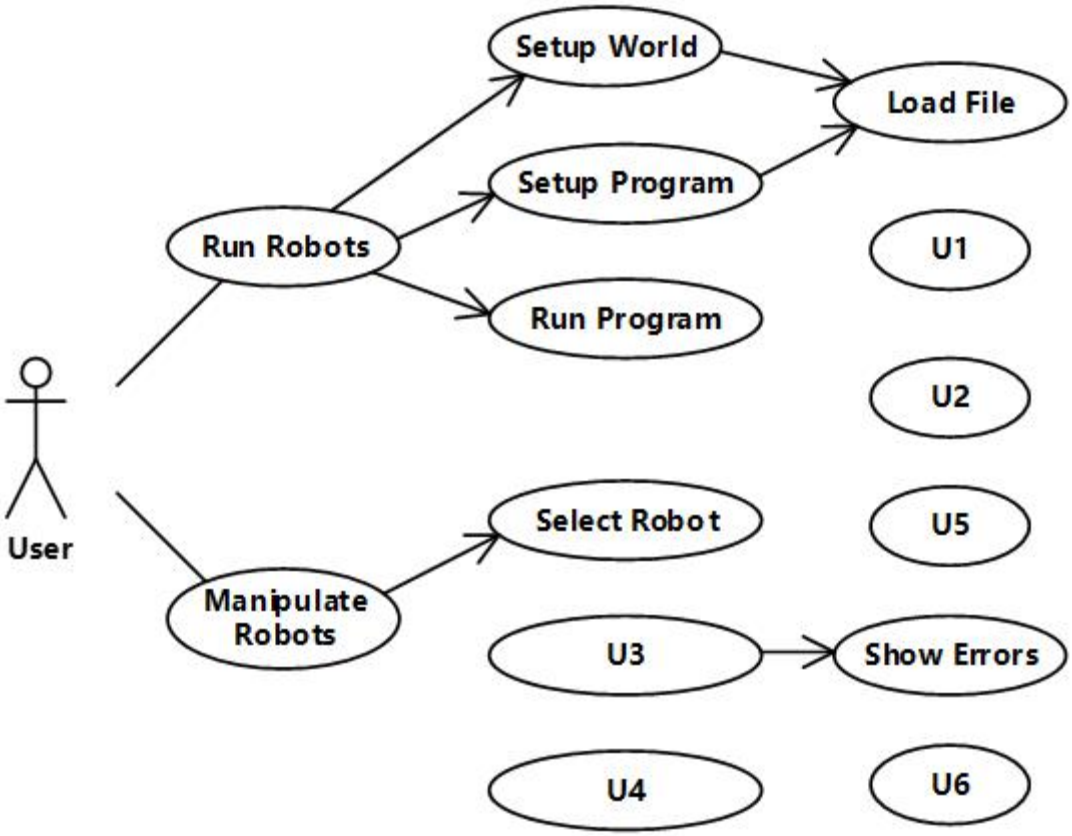
②Left：机器人原地沿逆时针方向旋转 90 度。

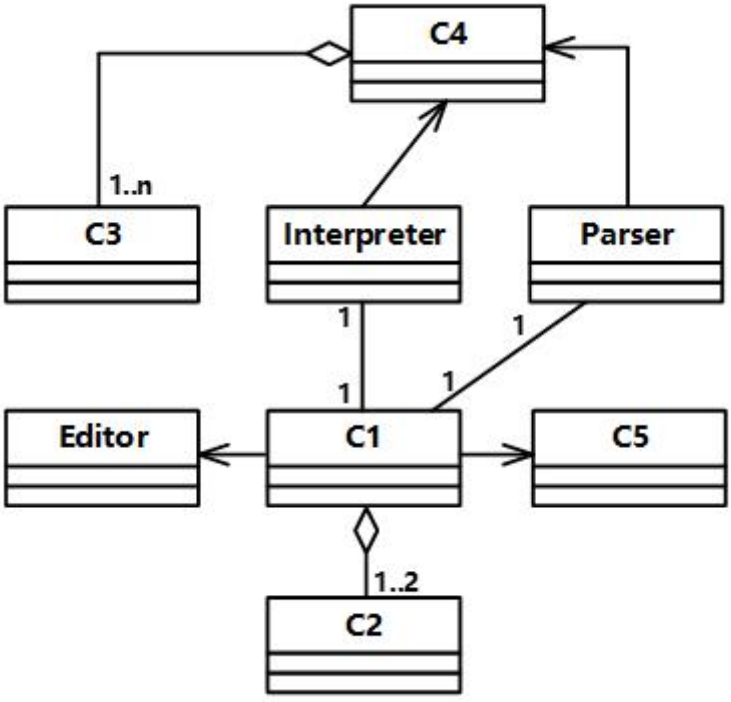
③Read：机器人读取其所在位置的字符，并将这个字符的值赋给 b1；如果这个位置上没有字符，则不改变 b1 的当前值。

④Write：将 b1 中的字符写入机器人当前所在的位置，如果这个位置上已经有字符，该字符的值将会被 b1 的值替代。如果这时 b1 没有值，即在执行 Write 动作之前没有执行过任何 Read 动作，那么需要提示用户相应的错误信息（Show Errors）。

手动控制与单步控制的区别在于，单步控制时执行的是指令中的动作，只有一种控制方式，即执行下个动作；而手动控制时有 4 种动作。

现采用面向对象方法设计并实现该仿真系统，得到如图 3-1 所示的用例图和图 3-2 所示的初始类图。图 3-2 中的类“Interpreter”和“Parser”用于解析描述虚拟世界的文件以及机器人行为文件中的指令集。


Setup World
Load File
Setup Program
Run Robots
U1
Run Program
U2
User
Select Robot
U5
Manipulate Robots
U3
Show Errors
U4
U6


C4
1..n
C3
Interpreter
Parser
1
1
1
1
Editor
C1
C5
1..2
C2

【问题 1】（6 分)

根据说明中的描述，给出图 3-1 中 U1～U6 所对应的用例名。

【问题 2】（4 分）

图 3-1 中用例 U1～U6 分别与哪个（哪些）用例之间有关系，是何种关系？

【问题 3】（5 分）

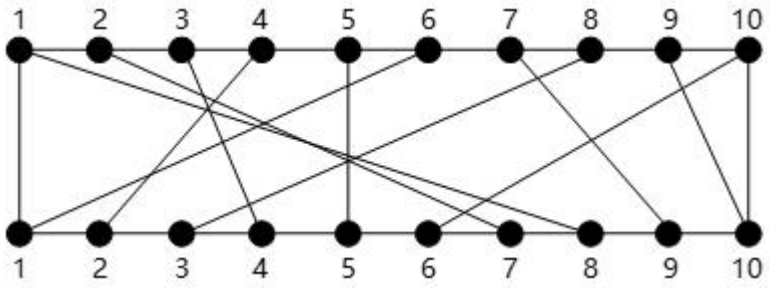
根据说明中的描述，给出图 3-2 中 C1～C5 所对应的类名。

试题四（共 15 分）

阅读下列说明和 C 代码，回答问题 1 至问题 3，将解答写在答题纸的对应栏内。

【说明】

在一块电路板的上下两端分别有 n 个接线柱。根据电路设计，用(i, π(i))表示将上端接线柱 i 与下端接线柱 π(i)相连，称其为该电路板上的第 i 条连线。如图 4-1 所示的 π(i)排列为{8,7,4,2,5,1,9,3,10,6}。对于任何 1≤i<j≤n，第 i 条连线和第 j 条连线相交的充要条件是 π(i)> π(j)。



在制作电路板时，要求将这 n 条连线分布到若干绝缘层上，在同一层上的连线不相交。现在要确定将哪些连线安排在一层上，使得该层上有尽可能多的连线，即确定连线集 Nets={(i, π(i))，1≤i≤n}的最大不相交子集。

【分析问题】

记 N(i,j)={t|(t, π(t)) ∈ Nets,t ≤ i, π(t) ≤ j}。N(i,j)的最大不相交子集为 MNS(i,j)，size(i,j)=|MNS(i,j)|。

经分析，该问题具有最优子结构性质。对规模为 n 的电路布线问题，可以构造如下递归式：

(1) 当 i = 1 时，$size(1,j)=\begin{cases}0 & j<\pi(1)\\ 1 & \text{其它情况}\end{cases}$

(2) 当 i > 1 时，$size(i,j)=\begin{cases}size(i-1,j) & j<\pi(i)\\ \max\{size(i-1,j),size(i-1,\pi(i)-1)+1\} & \text{其它情况}\end{cases}$

【C 代码】

下面是算法的 C 语言实现。

（1）变量说明

size[i][j]：上下端分别有 i 个和 j 个接线柱的电路板的第一层最大不相交连接数

pi[i]： π(i)，下标从 1 开始

（2）C 程序

```
#include "stdlib.h"
```

```
#include <stdio.h>
#define N 10 /*问题规模*/
int m=0;  /*牢记录最大连接集合中的接线柱*/
Void maxNum(int pi[],int size[N+1][N+1],int n) {/*求最大不相交连接数*/
    int i, j;
    for(j=0; j < pi[1]; j++) size[1][j] = 0; /*当 j<π(1)时 */
    for(j=pi[1];j<=n;j++) （1） ; /*当 j>=π(1)时 */
    for(i=2; i < n; i++) {
        for(j=0; j < pi[i]; j++) （2） ; /*当 j<pi[i]时 */
        for(j=pi[i];j<=n；j++) {/*当 j>=c[i]时,考虑两种情况*/
            size[i][j]=size[i-1][j]>=size[i-1][pi[i]-1]+1 ?size[i-1][j]:size[i-1][pi[i]-1]+1;
        }
    }
    /*最大连接数 */
    size[n][n]=size[n-1][n]>=size[n-1][pi[n]-1]+1 ? size[n-1][n]:size[n-1][pi[n]-1]+1;
}

/*构造最大不相交连接集合，net[i]表示最大不相交子集中第 i 条连线的上端接线柱的序号 */
void constructSet（int pi[],int size[N+1][N+1],int n,int net[n]）{
int i,j=n;
m=0；
    for(i=n ; i>1 ; i--) { /*从后往前*/
        if(size[i][j]!=size[i-1][j]){ /*(i,pi[i])是最大不相交子集的一条连线*/
            （3） ; /*将 i 记录到数组 net 中，连接线数自增 1*/
            j= pi[i]-1; /*更新扩展连线柱区间*/
        }
    }
    if(j>=pi[1]) net[m++]=1; /*当 i=1 时*/
}
```

【问题 1】（6 分）

根据以上说明和 C 代码，填充 C 代码中的空（1）～（3）。

【问题 2】（6 分）

据题干说明和以上 C 代码，算法采用了（4）算法设计策略。

【问题 3】（3 分）

若连接排列为{8,7,4,2,5,1,9,3,10,6}，即如图 4-1 所示，则最大不相交连接数为（7），包含的连线为（8）（用(i, π (i))的形式给出）。

从下列的 2 道试题（试题五至试题六）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（共 15 分）

阅读下列说明和 C++代码，将应填入（n）处的字句写在答题纸的对应栏内。

【说明】

某软件系统中，已设计并实现了用于显示地址信息的类 Address（如图 5-1 所示），现要求提供基于 Dutch 语言的地址信息显示接口。为了实现该要求并考虑到以后可能还会出现新的语言的接口，决定采用适配器（Adapter）模式实现该要求，得到如图 5-1 所示的类图。

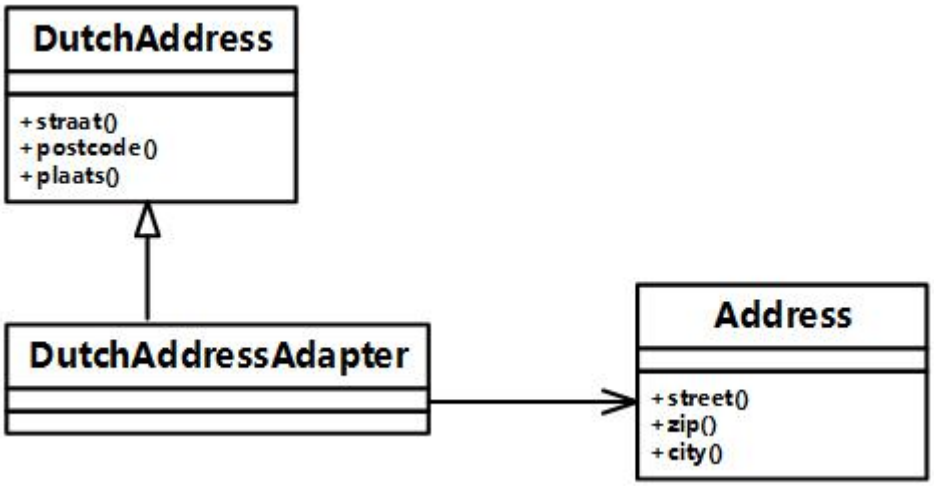


图 5-1 适配器模式类图

【C++代码】

```
#include <iostream>
using namespace std;
class Address{
    public:
        void stree() { /*  实现代码省略  */ }
        void zip() { /*  实现代码省略  */ }
        void city() { /*  实现代码省略  */
    }
    //其他成员省略
```

```
};

class DutchAddress {
    public:
        virtual void straat()=0;
        virtual void postcode()=0;
        virtual void plaats()=0;
        //其他成员省略
};

class DutchAddressAdapter : public DutchAddress {
    private:
        （1） ;
    public:
        DutchAddressAdapter(Address *addr) {
            address = addr;
        }

    void straat() {
        （2） ;
    }

    void postcode(){
        （3） ;
    }
    void plaat(){
        （4） ;
    }
    //其他成员省略
```

```
};

void testDutch(DutchAddress *addr){
        addr->straat();
        addr->postcode();
        addr->plaats();
}

int main(){
        Address*addr = new Address();
        （5） ;
        cout<< "\n The DutchAddress\n"<< endl;
        testDutch(addrAdapter);
        return 0;
}
```

试题六(共 15 分）

阅读下列说明和 Java 代码，将应填入 （n） 处的字句写在答题纸的对应栏内。

【说明】

某软件系统中，已设计并实现了用于显示地址信息的类 Address（如图 6-1 所示），现要求提供基于 Dutch 语言的地址信息显示接口。为了实现该要求并考虑到以后可能还会出现新的语言的接口，决定采用适配器（Adapter）模式实现该要求，得到如图 6-1 所示的类图。

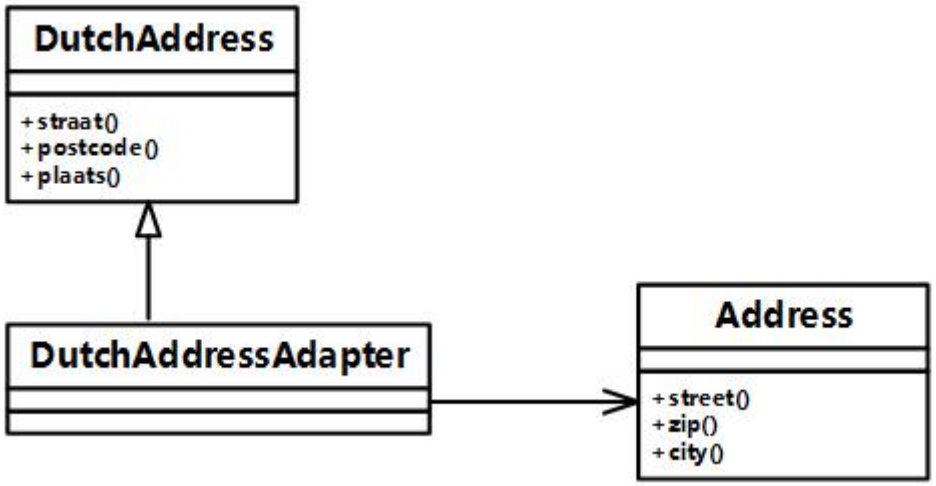


图 6-1 适配器模式类图

【Java 代码】

```
import java.util.*;
Class Address{
    public void street() { //实现代码省略 }
    public void zip() { //实现代码省略 }
    public void city() { //实现代码省略 }
    //其他成员省略
}

class DutchAddress{
    public void straat() { //实现代码省略 }
    public void postcode() { //实现代码省略 }
    public void plaats() { //实现代码省略 }
```

```
    //其他成员省略
}

class DutchAddressAdapter extends DutchAddress {
    private （1） ;
    public DutchAddressAdapter (Address addr){
        address= addr;
    }

    public void straat() {
        （2） ;
    }

    public void postcode() {
        （3） ;
    }

    public void plaats(){
        （4） ;
    }
    //其他成员省略
}

class Test {
    public static void main(String[] args) {
        Address addr= new Address();
        （5） ;
        System.out.println("\n The DutchAddress\n");
        testDutch(addrAdapter);
```

```
    }
    Static void testDutch(DutchAddress addr){
        addr.straat();
        addr.postcode();
        addr.plaats();
    }
}
```