

题型一。阅读下列说明和 UML 图,回答问题 1 至问题 3, 将应填入__(n)__处的解答写在答卷相应序号栏内。

【说明】

某网上购物平台的主要功能如下:

(1)创建订单。顾客 (Customer) 在线创建订单 (Order), 主要操作是向订单中添加项目、从订单中删除项目。订单中应列出所订购的商品 (Product) 及其数量 (quantities)。

(2)提交订单。订单通过网络来提交。在提交订单时, 顾客需要提供其姓名 (name)、收货地址 (address)、以及付款方式 (form of payment) (预付卡、信用卡或者现金)。为了制定送货计划以及安排送货车辆, 系统必须确定订单量 (volume)。除此之外, 还必须记录每种商品的名称 (Name)、造价 (cost price)、售价 (sale price) 以及单件商品的包装体积 (cubic volume)。

(3)处理订单。订单处理人员接收来自系统的订单; 根据订单内容, 安排配货, 制定送货计划。在送货计划中不仅要指明发货日期 (delivery date), 还要记录每个订单的限时发送要求 (Delivery Time Window)。

(4)派单。订单处理人员将已配好货的订单转交给派送人员。

(5)送货 / 收货。派送人员将货物送到顾客指定的收货地址。当顾客收货时, 需要在运货单 (delivery slip) 上签收。签收后的运货单最终需交还给订单处理人员。

(6)收货确认。当订单处理人员收到签收过的运货单后, 会和顾客进行一次再确认。

现采用面向对象方法开发上述系统, 得到如图 1-1 所示的用例图和图 1-2 所示的类图。

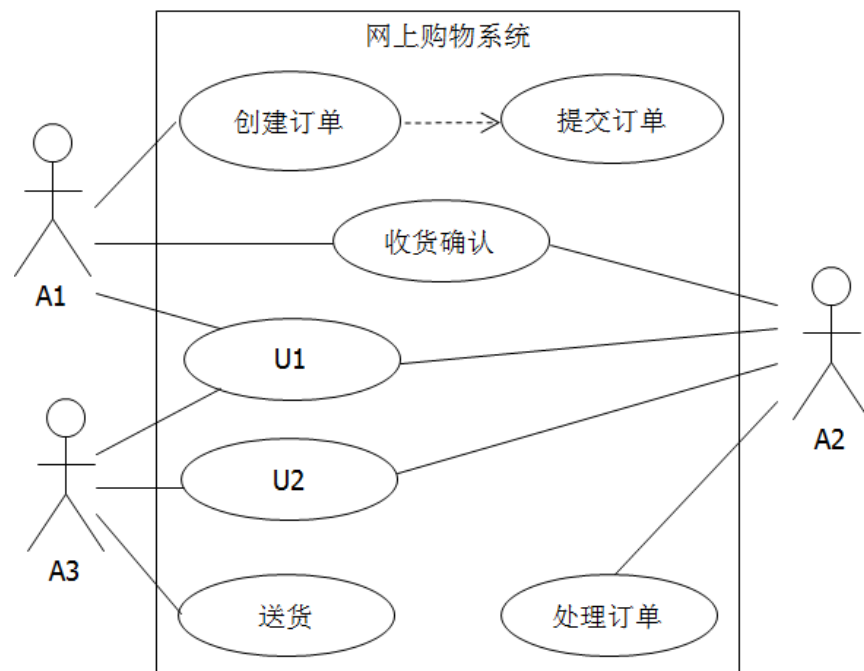


图 1-1

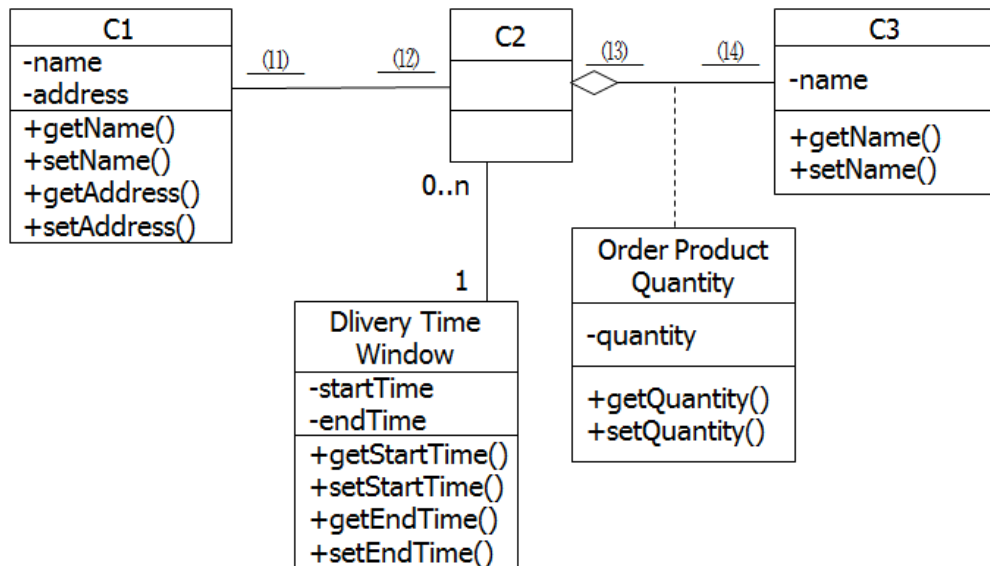


图 1-2

【问题 1】

根据说明中的描述，给出图 1-1 中 A1~A3 所对应的参与者名称和 U1~U2 处所对应的用例名称。

A1: __ (1) __。A2: __ (2) __。A3: __ (3) __。U1: __ (4) __。U2: __ (5) __。

【问题 2】

根据说明中的描述，给出图 1-2 中 C1~C3 所对应的类名，并将类 C2 和 C3 的属性补充完整（属性命名使用说明中给出的英文词汇）。

C1 的名称: __ (6) __。C2 的名称: __ (7) __。C3 的名称: __ (8) __。

补充类 C2 的属性: __ (9) __。补充类 C3 的属性: __ (10) __。

【问题 3】

根据说明中的描述给出图 1-2 中 (11)~(14) 处所对应的多重性（类名使用说明中给出的英文词汇）。

题型二。阅读下列说明和 UML 图, 回答问题 1 至问题 2, , 将应填入 __ (n) __ 处的解答按空格编号顺序写在答题纸的对应栏内。

【说明】

某城市拟开发一个基于 Web 城市黄页，公开发布该城市重要的组织或机构（以下统称为客户）的基本信息，方便城市生活。该系统的主要功能描述如下：

1. 搜索信息：任何使用 Intenert 的网络用户都可以搜索发布在城市黄页中的信息，例如客户的名称、地址、联系电话等。

2. 认证：客户若想在城市黄页上发布信息，需通过系统的认证。认证成功后，该客户成为系统授权用户。

3. 更新信息：授权用户登录系统后，可以更改自己在城市黄页中的相关信息，例如变更联系电话等。

4. 删除客户：对于拒绝继续在城市黄页上发布信息的客户，由系统管理员删除该客户的相关信息。

系统采用面向对象方法进行开发，在开发过程中认定出如表 2-1 所示的类。系统的用例图和类图分别如图 2-1 和图 2-2 所示。

表 2-1 类列表

| 类名 | 说明 |
|----|----|
|----|----|

| | |
|------------------|--------------------|
| InternetClient | 网络用户 |
| CustomerList | 客户集，维护城市网页上的所有客户信息 |
| Customer | 客户信息，记录单个客户的信息 |
| RegisteredClient | 授权用户 |
| Administrator | 系统管理员 |

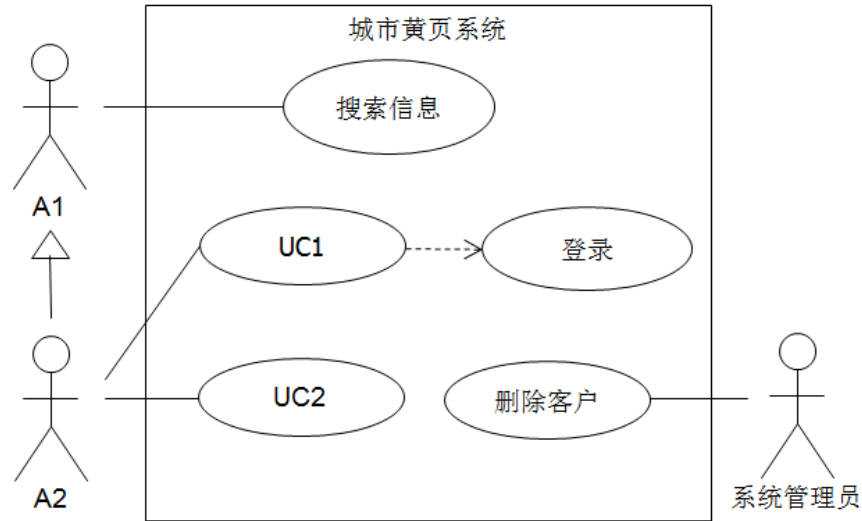


图 2-1 系统用例图

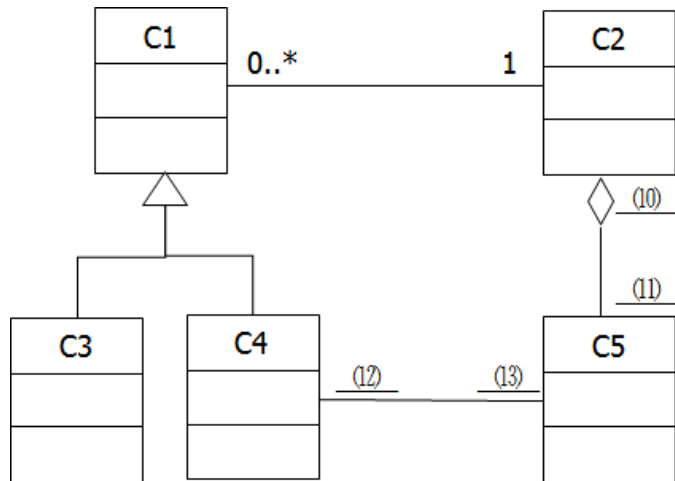


图 2-2 系统类图

【问题 1】

根据说明中的描述，给出图 2-1 中 A1 和 A2 处所对应的参与者名称，UC1 和 UC2 所对应的用例名称。

A1： __ (1) __。 A2： __ (2) __。

UC1： __ (3) __。 UC2： __ (4) __。

【问题 2】

根据说明中的描述，给出图 2-2 中 C1~C5 所对应的类名（表 2-1 中给出的类名）和 (10)~(13)处所对应的多重性。

C1： __ (5) __。 C2： __ (6) __。 C3： __ (7) __。 C4： __ (8) __。 C5： __ (9) __。

题型三。阅读下列说明和 UML 图,回答问题 1 至问题 5,将解答填入答题纸的对应栏内。

【说明】

某银行计划开发一个自动存提款机模拟系统 (ATM System)。系统通过读卡器 (CardReader) 读取 ATM 卡; 系统与客户 (Customer) 的交互由客户控制台 (CustomerConsole) 实现; 银行操作员 (Operator) 可控制系统的启动 (System Startup) 和停止 (System Shutdown); 系统通过网络和银行系统 (Bank) 实现通信。

当读卡器判断用户已将 ATM 卡插入后, 创建会话 (Session)。会话开始后, 读卡器进行读卡, 并要求客户输入个人验证码 (PIN)。系统将卡号和个人验证码信息送到银行系统进行验证。验证通过后, 客户可从菜单选择如下事务 (Transaction):

1. 从 ATM 卡账户取款 (Withdraw);
2. 向 ATM 卡账户存款 (Deposit);
3. 进行转账 (Transfer);
4. 查询 (Inquire) ATM 卡账户信息。

一次会话可以包含多个事务, 每个事务处理也会将卡号和个人验证码信息送到银行系统进行验证。若个人验证码错误, 则转个人验证码错误处理 (Invalid PIN Process)。每个事务完成后, 客户可选择继续上述事务或退卡。选择退卡时, 系统弹出 ATM 卡, 会话结束。

系统采用面向对象方法开发, 使用 UML 进行建模。系统的顶层用例图如图 3-1 所示, 一次会话的序列图 (不考虑验证) 如图 3-2 所示。消息名称参见表 3-1。

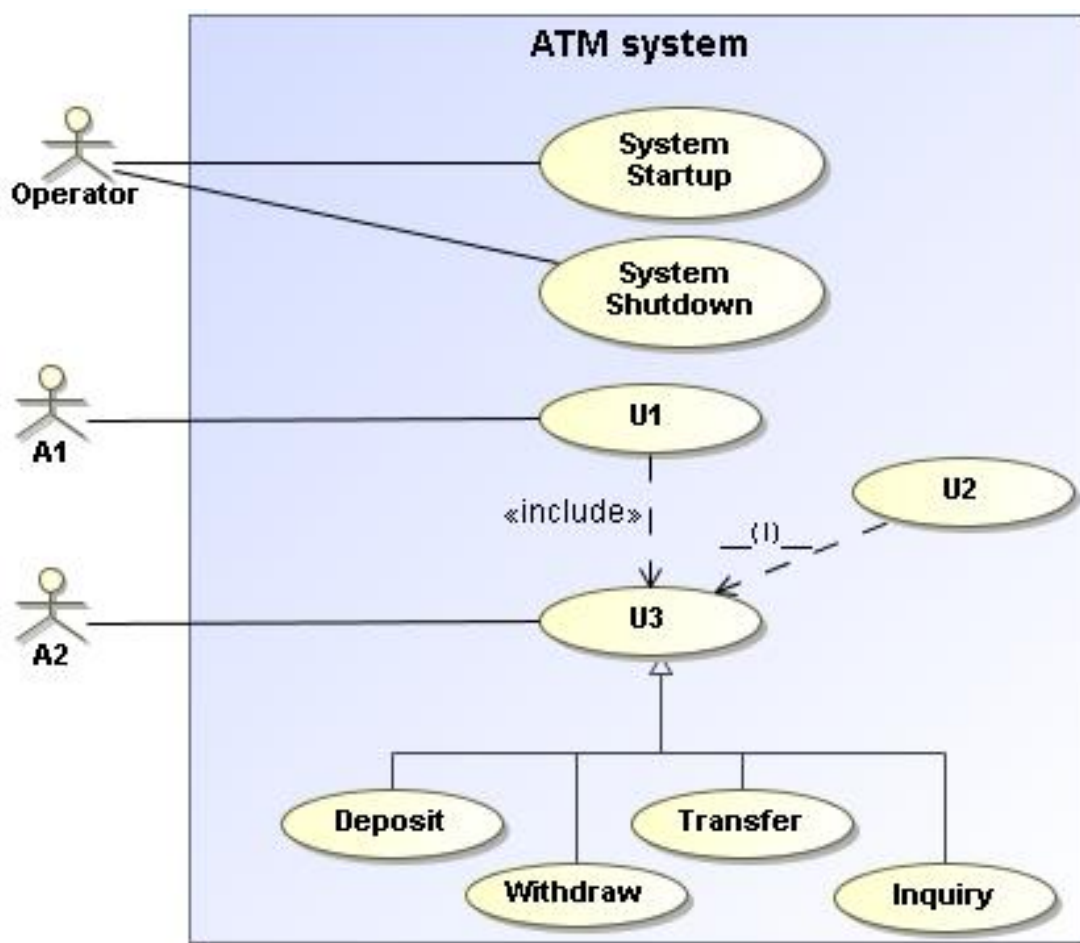


图 3-1

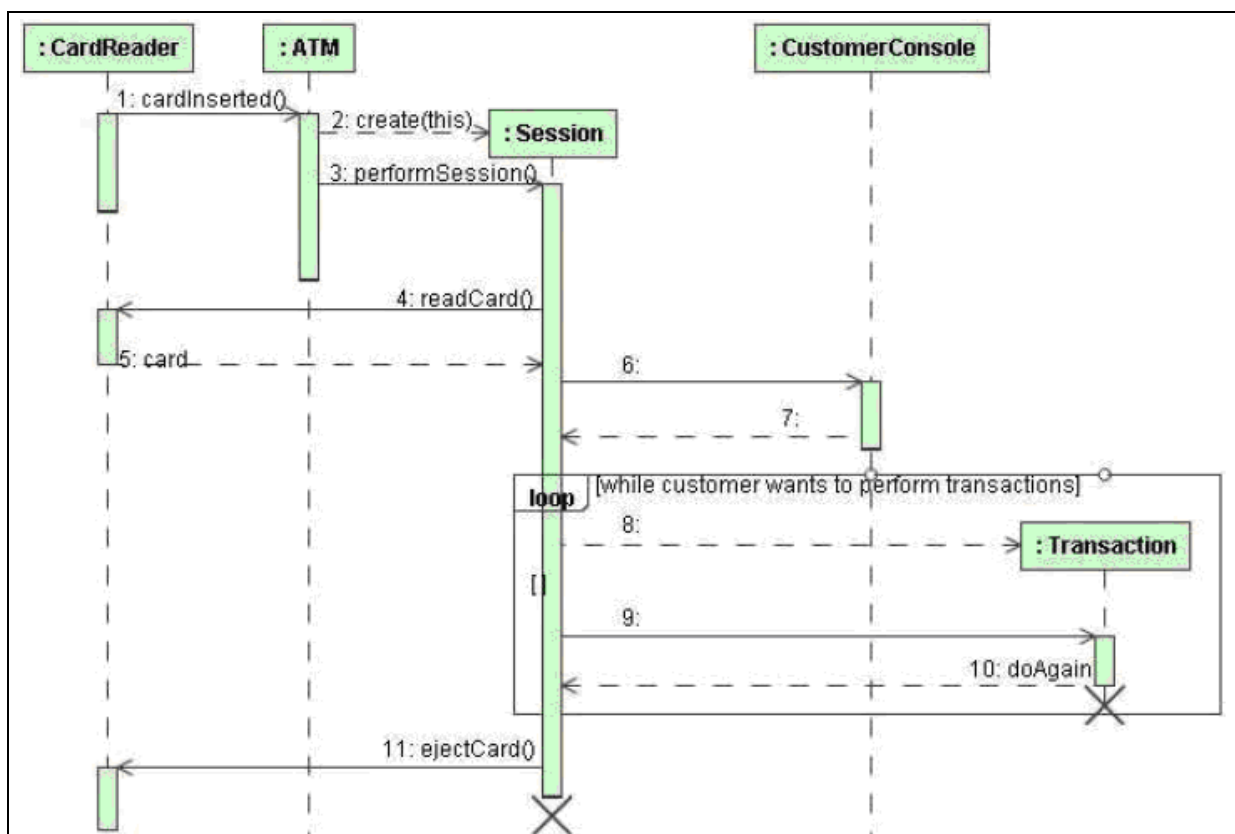


图 3-2

表 3-1 可能的消息名称列表

| 名称 | 说明 | 名称 | 说明 |
|-----------------------------|-----------|----------------------|--------------|
| cardInsert() | ATM 卡已插入 | performTransaction() | 执行事务 |
| performSession() | 执行会话 | readCard() | 读卡 |
| readPin() | 读取个人验证码 | PIN | 个人验证码信息 |
| creat(atm, this, card, pin) | 为当前会话创建事务 | creat(this) | 为当前 ATM 创建会话 |
| card | ATM 卡信息 | doAgain | 执行下一个事务 |
| ejectCard() | 弹出 ATM 卡 | | |

【问题 1】

根据【说明】中的描述，给出图 3-1 中 A1 和 A2 所对应的参与者（使用英文名称）。

【问题 2】

根据【说明】中的描述，给出图 3-1 中 U1 至 U3 所对应的用例（使用英文名称）。

【问题 3】

根据【说明】中的描述，给出图 3-1 中空（1）所对应的关系（使用英文名称）。

【问题 4】

根据【说明】中的描述，使用表 3-1 中的英文名称，给出图 3-2 中 6~9 对应的消息。

【问题 5】

解释图 3-1 中用例 U3 和用例 Withdraw、Deposit、Transfer、Inquire 等四个用例之间的关系及其内涵。

题型四。阅读下列说明、类图、通信图和 JAVA 代码，将应填入__(n)__处的解答写在答卷相应序号栏内。

【说明】

图 4-1 所示的类图展示某销售处理系统中与计算订单总额相关的类。图 4-2 所示的通信图展示了参与计算订单总额行为的对象之间的交互。

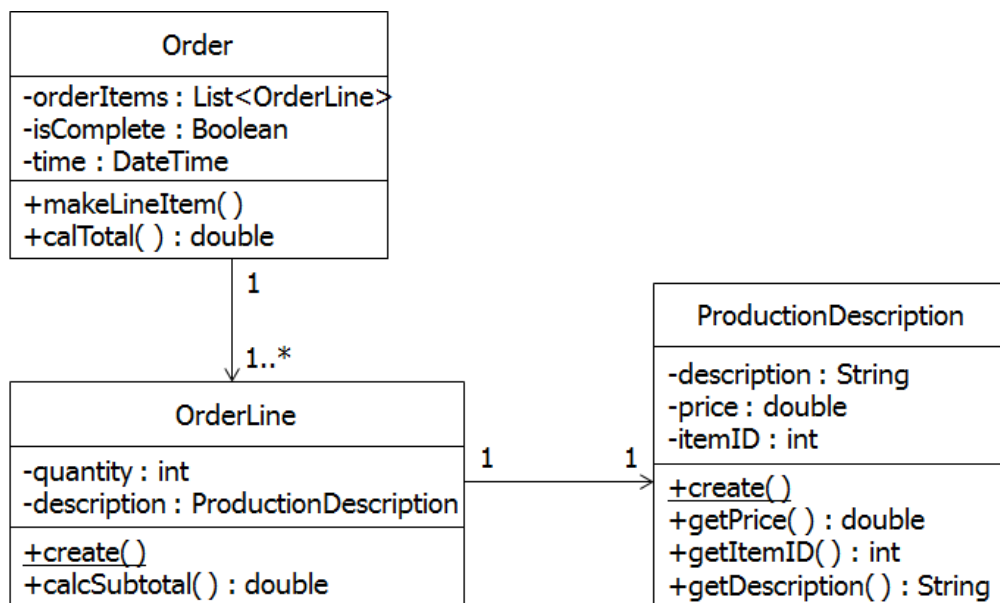


图 4-1

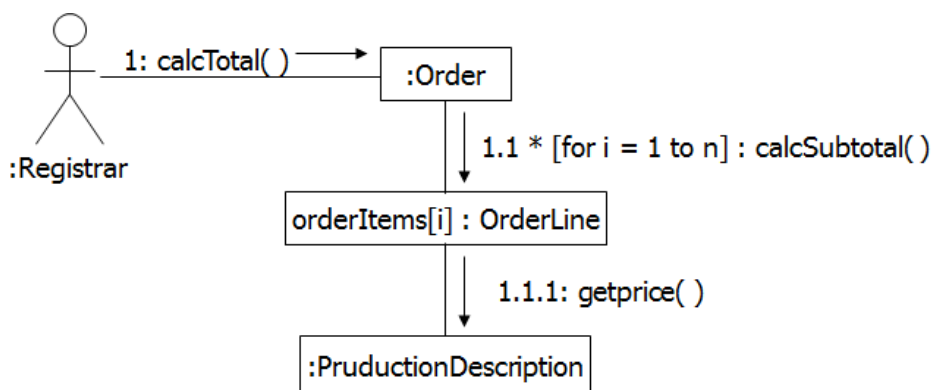


图 4-2

【JAVA 代码】

下面的代码展示了用 JAVA 实现的为计算订单总额进行交互的设计类，所生成的代码是从上面的设计类图和交互图产生的。

```

public class Order
{
    private List<OrderLine> orderItems =
        new ArrayList()<OrderLine>;

    public double calcTotal()
    {
        double total = 0;
        double subtotal = 0;

        for (   (1)   )
    
```

```

        {
            subtotal = (2) ;

            total = (3) ;

        }
        return total;
    }
}

public class OrderLine
{
    private int    quantity;
    private    ProductDescription    description;

    public SalesLineItem (ProductDescription desc, int quantity )
    {
        this.description = desc;
        this.quantity = quantity;
    }

    public double calcSubtotal()
    {
        return (4) ;
    }
}

public class ProductDescription
{
    private double price;
    private ItemID id;
    private String description;

    public ProductDescription
        ( ItemID id, Money price, String description )
    {
        this.id = id;
        this.price = price;
        this.description = description;
    }

    public double getPrice() { return (5) ; }
}

```