

牛客网-华为机试练习题 99

题目描述

1 总体说明

考生需要模拟实现一个简单的自动售货系统，实现投币、购买商品、退币、查询库存商品及存钱盒信息的功能。

系统初始化时自动售货机中商品为6种商品,商品的单价参见1.1规格说明，存钱盒内放置1元、2元、5元、10元钱币，商品数量和钱币张数通过初始化命令设置，参见2.1 系统初始化。

1.1规格说明

\1. 商品:每种商品包含商品名称、单价、数量三种属性，其中商品名不重复。考生不能修改商品名称和单价，初始化命令设置商品数量。这些信息在考试框架中进行定义，考生在实现功能代码时可直接使用。

商品 名称	单价	数量
A1	2	X
A2	3	X
A3	4	X
A4	5	X
A5	8	X
A6	6	X

\2. 存钱盒信息：钱币面额、张数两种属性。初始化命令设置各种面额钱币张数。这些信息在考试框架中进行定义，考生在实现功能代码时可直接使用。

钱币面额	张数
10元	X
5元	X
2元	X
1元	X

\3. 退币原则：

1) 根据系统存钱盒内钱币的信息，按钱币总张数最少的原则进行退币。

2) 如果因零钱不足导致不能退币，则尽最大可能退币，以减少用户损失。

例如：假设存钱盒内只有4张2元，无其它面额钱币。如果需要退币7元，系统因零钱不足无法退币，则继续尝试退币6元，最终系统成功退币3张2元,用户损失1元钱币。

\4. 投币操作说明：每次投币成功，投入的钱币面额累加到投币余额；同时，本次投入的钱币放入存钱盒中，存钱盒相应面额钱币增加。

\5. 投币余额：指当前自动售货机中用户剩余的可购买商品的钱币总额；例如：投入2元面额的钱币，投币余额增加2元；购买一件价格2元的商品，投币余额减少2元；

\6. 投币余额约束：投币余额不能超过10元。

- \\7. 退币操作说明：退币操作需要遵守 **退币原则**；退币成功后，投币余额清零，同时扣除存钱盒相应的金额。
- \\8. 购买商品操作说明：一次仅允许购买一件商品；购买商品成功后，自动售货机中对应商品数量减1，投币余额扣除本次购买商品的价格。

2 操作说明

命令字与第一个参数间使用一个空格分隔，多条命令采用分号隔开。考试系统会对输入命令格式进行处理，考生不需要关注输入**命令格式**的合法性，只需要实现命令处理函数。

2.1 系统初始化

命令格式：

r A1 数量 -A2 数量 -A3 数量 -A4 数量 -A5 数量 -A6 数量 1 元张数 -2 元张数 -5 元张数 -10 元张数

参数名称	参数说明	类型	取值范围
A1数量	商品A1数量	整数	[0,10]
A2数量	商品A2数量	整数	[0,10]
A3数量	商品A3数量	整数	[0,10]
A4数量	商品A4数量	整数	[0,10]
A5数量	商品A5数量	整数	[0,10]
A6数量	商品A6数量	整数	[0,10]
1元张数	面额1元钱币张数	整数	[0,10]
2元张数	面额2元钱币张数	整数	[0,10]
5元张数	面额5元钱币张数	整数	[0,10]
10元张数	面额10元钱币张数	整数	[0,10]

商品和各种面额钱币取值范围只是作为初始化命令的限制，其它场景下不限制取值范围；考试框架已经实现取值范围的检查，考生不需要关注。

功能说明：设置自动售货机中商品数量和存钱盒各种面额的钱币张数；

约束说明：系统在任意阶段均可执行r初始化系统；考生不需要关注参数的合法性，不需要关注增加或缺少参数的场景；

输出说明：输出操作成功提示（执行完r命令后系统会自动输出操作结果，考生不需要再次调用输出函数），例：

命令	输出	含义
r 6-5-4-3-2-1 4-3-2-1;	S001:Initialization is successful	初始化成功

2.2 投币

命令格式：p 钱币面额

功能说明：

- (1) 如果投入非1元、2元、5元、10元的钱币面额（钱币面额不考虑负数、字符等非正整数的情况），输出“E002:Denomination error”；
- (2) 如果存钱盒中1元和2元面额钱币总额小于本次投入的钱币面额，输出“E003:Change is not enough, pay fail”，但投入1元和2元面额钱币不受此限制。

- (3) 如果投币余额大于10元，输出“E004:Pay the balance is beyond the scope biggest”;
- (4) 如果自动售货机中商品全部销售完毕，投币失败。输出“E005:All the goods sold out”;
- (5) 如果投币成功，输出“S002:Pay success,balance=X”;

约束说明：

- (1) 系统在任意阶段都可以投币；
- (2) 一次投币只能投一张钱币；
- (3) 同等条件下，错误码的优先级：E002 > E003 > E004 > E005；

输出说明：如果投币成功，输出“S002:Pay success,balance=X”。

例：

命令	输出
p 10;	S002:Pay success,balance=10

2.3 购买商品

命令格式：b 商品名称

功能说明：

- (1) 如果购买的商品不在商品列表中，输出“E006:Goods does not exist”;
- (2) 如果所购买的商品的数量为0，输出“E007:The goods sold out”;
- (3) 如果投币余额小于待购买商品价格，输出“E008:Lack of balance”;
- (4) 如果购买成功，输出“S003:Buy success,balance=X”;

约束说明：

- (1) 一次购买操作仅能购买一件商品，可以多次购买；
- (2) 同等条件下，错误码的优先级：E006 > E007 > E008；

输出说明：

如果购买成功，输出“S003:Buy success,balance=X”。

例：

命令	输出
b A1;	S003:Buy success,balance=8

2.4 退币

命令格式：c

功能说明：

- (1) 如果投币余额等于0的情况下，输出“E009:Work failure”；
- (2) 如果投币余额大于0的情况下，按照 **退币原则** 进行“找零”，输出退币信息；

约束说明：

- (1) 系统在任意阶段都可以退币；
- (2) 退币方式必须按照 **退币原则** 进行退币；

输出说明：如果退币成功，按照 **退币原则** 输出退币信息。

例，退5元钱币：

命令 输出

c; 1 yuan coin number=02 yuan coin number=05 yuan coin number=110 yuan coin number=0

2.5 查询

命令格式：q 查询类别

功能说明：

(1) 查询自动售货机中商品信息，包含商品名称、单价、数量。 **根据商品数量从大到小进行排序；商品数量相同时，按照商品名称的先后顺序进行排序。**

例如：A1的商品名称先于A2的商品名称，A2的商品名称先于A3的商品名称。

- (2) 查询存钱盒信息，包含各种面额钱币的张数；
- (3) 查询类别如下表所示：

查询类别	查询内容
0	查询商品信息
1	查询存钱盒信息

如果“查询类别”参数错误，输出“E010:Parameter error”。“查询类别”参数错误时，不进行下面的处理；

输出说明：

“查询类别”为0时，输出自动售货机中所有商品信息（商品名称单价数量）例：

命令	输出
q 0;	A 1 2 6A 2 3 5A 3 4 4A 4 5 3A 5 8 2A 6 6 0

“查询类别”为1时，输出存钱盒信息（各种面额钱币的张数），格式固定。例：

命令	输出
q 1;	1 yuan coin number=42 yuan coin number=35 yuan coin number=210 yuan coin number=1

输入描述:

依照说明中的命令码格式输入命令。

输出描述:

输出执行结果

示例1

输入

```
r 1-1-1-1-1-1 10-5-2-1;p 1;q 1;
```

输出

```
S001:Initialization is successful
S002:Pay success,balance=1
1 yuan coin number=11
2 yuan coin number=5
5 yuan coin number=2
10 yuan coin number=1
```

解决代码:

```
import java.util.Scanner;

/**
 * 自动售货系统
 * @author Administrator
 *
 */
public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while(sc.hasNext()){
            String s = sc.nextLine();
            doString(s);
        }
        sc.close();
    }
    public static void doString(String input){
        String[] arr = input.split(",");
        String initial = arr[0];
        Goods goods = OpReset(initial);
        for(int i=1;i<arr.length;i++){
            String temp = arr[i];
            String[] temp2 = temp.split(" ");
            if(temp2[0].equals("p")){
                goods = OpPay(temp,goods);
            }else if(temp2[0].equals("b")){
                goods = OpBuy(temp,goods);
            }else if(temp2[0].equals("c")){
                goods = OpChange(temp,goods);
            }else if(temp2[0].equals("q")||temp2[0].matches("q\\d")){
                OpQuery(temp,goods);
            }
        }
    }
}
```

```

    }

}

public static Goods OpReset(String initial){//初始化
    String[] arr = initial.split(" ");
    String[] Asome = arr[1].split("-");
    String[] Moneysome = arr[2].split("-");
    int[] A = new int[Asome.length];
    int[] B = new int[Moneysome.length];
    for(int i=0;i<Asome.length;i++){
        A[i]=Integer.valueOf(Asome[i]);
    }
    for(int i=0;i<Moneysome.length;i++){
        B[i]=Integer.valueOf(Moneysome[i]);
    }
    Goods goods = new Goods(A,B,0);
    System.out.println("S001:Initialization is successful");
    return goods;
}

public static Goods OpPay(String payString, Goods goods){//投币
    String[] arr = payString.split(" ");
    int payNum = Integer.valueOf(arr[1]);
    if(payNum!=1&&payNum!=2&&payNum!=5&&payNum!=10){
        System.out.println("E002:Denomination error");
        return goods;
    }
    else if((payNum==5||payNum==10)&&(goods.num_coin[0]*1+goods.num_coin[1]*2<payNum)){
        System.out.println("E003:Change is not enough, pay fail");
        return goods;
    }
    else if(payNum>10){
        System.out.println("E004:Pay the balance is beyond the scope biggest");
        return goods;
    }
    else
if(goods.num_goods[0]==0&&goods.num_goods[1]==0&&goods.num_goods[2]==0&&goods.num_goods[3]==0
&&goods.num_goods[4]==0&&goods.num_goods[5]==0){
        System.out.println("E005:All the goods sold out");
        return goods;
    }
    else{
        switch(payNum){
            case 1:goods.num_coin[0]++;break;
            case 2:goods.num_coin[1]++;break;
            case 5:goods.num_coin[2]++;break;
            case 10:goods.num_coin[3]++;break;
        }
        int tmp = payNum + goods.toubi_yu_e;
        goods.toubi_yu_e += payNum;
        System.out.println("S002:Pay success,balance="+tmp);
        return goods;
    }
}

}

public static Goods OpBuy(String buyString, Goods goods){//购买商品
    String[] goodsString = {"A1", "A2", "A3", "A4", "A5", "A6"};
    String[] arr = buyString.split(" ");
    String buy = arr[1];
    int price = 0;

```

```

int index = 10;
for(int i=0;i<goodsString.length;i++){
    if(buy.equals(goodsString[i])){
        switch(i){
            case 0:price = 2;index=i;break;
            case 1:price = 3;index=i;break;
            case 2:price = 4;index=i;break;
            case 3:price = 5;index=i;break;
            case 4:price = 8;index=i;break;
            case 5:price = 6;index=i;break;
        }
    }
}
if(index==10){
    System.out.println("E006:Goods does not exist");
    return goods;
}
else if(goods.toubi_yu_e<price){
    System.out.println("E008:Lack of balance");
    return goods;
}
else if(goods.num_goods[index]==0){
    System.out.println("E007:The goods sold out");
    return goods;
}
else{
    goods.toubi_yu_e=goods.toubi_yu_e-price;
    System.out.println("S003:Buy success,balance="+goods.toubi_yu_e);
    return goods;
}
}

public static Goods opChange(String changeString,Goods goods){//退市
    if(goods.toubi_yu_e==0){
        System.out.print("E009:Work failure");
        return goods;
    }else{
        int tuibi = goods.toubi_yu_e;
        int num_shi = tuibi/10;

        if(goods.num_coin[3]-num_shi<0){
            num_shi = goods.num_coin[3];
        }
        int num_wu = (tuibi-10*num_shi)/5;
        if(goods.num_coin[2]-num_wu<0){
            num_wu = goods.num_coin[2];
        }
        int num_er = (tuibi-10*num_shi-5*num_wu)/2;
        if(goods.num_coin[1]-num_er<0){
            num_er = goods.num_coin[1];
        }
        int num_yi = (tuibi-10*num_shi-5*num_wu-2*num_er)/1;
        if(goods.num_coin[0]-num_yi<0){
            num_yi = goods.num_coin[0];
        }
        goods.num_coin[3] = goods.num_coin[3]-num_shi;
        goods.num_coin[0] = goods.num_coin[0]-num_yi;
        goods.num_coin[1] = goods.num_coin[1]-num_er;
        goods.num_coin[2] = goods.num_coin[2]-num_wu;
    }
}

```

```

        goods.toubi_yu_e=0;
        System.out.println("1 yuan coin number="+num_yi);
        System.out.println("2 yuan coin number="+num_er);
        System.out.println("5 yuan coin number="+num_wu);
        System.out.println("10 yuan coin number="+num_shi);
        return goods;
    }
}

public static void OpQuery(String queryString, Goods goods){
    String[] arr = queryString.split(" ");
    String query = "";
    if(arr.length==2){
        query = arr[1];
    }
    else{
        System.out.print("E010:Parameter error");
    }
    if(query.equals("0")){
        System.out.println("A1 "+"2 "+goods.num_goods[0]);
        System.out.println("A2 "+"3 "+goods.num_goods[1]);
        System.out.println("A3 "+"4 "+goods.num_goods[2]);
        System.out.println("A4 "+"5 "+goods.num_goods[3]);
        System.out.println("A5 "+"6 "+goods.num_goods[4]);
        System.out.println("A6 "+"7 "+goods.num_goods[5]);
    }else if(query.equals("1")){
        System.out.println("1 yuan coin number="+goods.num_coin[0]);
        System.out.println("2 yuan coin number="+goods.num_coin[1]);
        System.out.println("5 yuan coin number="+goods.num_coin[2]);
        System.out.println("10 yuan coin number="+goods.num_coin[3]);
    }
}

}

class Goods{
    int[] num_goods = new int[6];
    int[] num_coin = new int[4];
    int toubi_yu_e;
    public Goods(int[] num_goods, int[] num_coin, int toubi_yu_e){
        this.num_goods = num_goods;
        this.num_coin = num_coin;
        this.toubi_yu_e = toubi_yu_e;
    }
}

```