

# CTS-4

## 概览

考察内容：继承、方法重写、文件I/O等

本次CTS-4你需要实现的全部新功能（命令）如下表所示，详细要求请见下文。

虽然功能看起来不少，但是大部分是在原有基础上修改，新增功能逻辑比较简单。加油，CTS就要结束咧！

命令	参数1	参数2	参数3	参数4	参数5	功能描述
【修改】addUser	姓名	性别	卡号	学生优惠额度	无	添加用户【添加学生购票标志】
【修改】buyTicket	列车车次	出发站	目的站	席位代号	张数	用户购买车票【添加核酸检测判断】

命令	参数1	参数2	参数3	参数4	参数5	功能描述
【新增】 rechargeBalance	充值金额	无	无	无	无	充值钱包余额
【新增】 checkBalance	无	无	无	无	无	查询余额
【新增】 importCert	CSV文件名	无	无	无	无	导入核酸检测报告
【新增】 cancelOrder	列车车次	出发站	目的站	席位代号	张数	取消订单
【新增】 payOrder	无	无	无	无	无	支付全部未付款订单
【修改】listOrder	无	无	无	无	无	查询已购买的车票 listOrder

## 以下功能在原来基础上修改

### • 学生票功能

每位学生有多次以0.5折购买学生票的机会，并在下单时自动抵扣。

**学生优惠仅在结算时生效，订票、查询订单、取消订单时不涉及学生优惠！**

请实现以下功能、命令

- 创建学生类，继承自用户类，并做出如下修改：
  - 添加属性：剩余的优惠次数
- 查询用户身份为学生时，重写学生类的toString方法，按照以下格式输出

```
Name: sb_DYY
Sex: 0
Aadhaar: 091000720112
Discount: 1
```

- 修改方法：addUser
- 参数4存在时创建Student类对象，并将优惠额度数设为参数4对应的值（保证参数4为合法的正整数）；若不存在，则和CTS-1中要求一致。

选项	参数1	参数2	参数3	参数4【新增】	功能描述
addUser	姓名	性别	卡号	学生优惠次数	新增用户对象并储存相关信息，对于非法输入，终端输出相应报错。对于合法输入，调用该对象的 <code>toString()</code> 方法。

- 修改方法：listOrder
  - listOrder命令查看的价格是全票价格（与CTS-3相同），始终不显示折扣后的价格
  - 学生票优惠仅在payOrder结算时进行扣减，在buyTicket之后payOrder之前不进行扣减
  - 输出格式

【新增】`paid:[结算状态(T/F)]` (支付功能详细描述见下方)，T(true)表示已经结算，F(false)表示未结算。

[[列车车次]: [出发站]->[目的站]] seat:[席别代号] num:[张数] price:[总价(保留两位小数)] paid:[结算状态(T/F)]

## • 购票功能

- 修改CTS-3中的购票命令

命令	参数1	参数2	参数3	参数4	参数5	功能描述
buyTicket	列车车次	出发站	目的站	席位代号	张数	用户购买车票

- 输出
  - 购买成功输出 `Thanks for your order`
- 异常处理
  - 命令不存在，输出 `Command does not exist`
  - 参数数量不正确，输出 `Arguments illegal`
  - 当前无用户登录，输出 `Please login first`
  - 列车车次不存在，输出 `Train does not exist`
  - 出发站或目的站不存在（地球上不存在或该车次不经过该车站），输出 `Station does not exist`
  - 席位代号与车次类型不符（席位代号非法或该车次没有这种席位），输出 `Seat does not match`
  - 【新增】若当前用户购买车票坐席为[1A, 2A],且核酸检测证明结果为阳性或不存在，则输出 `Cert illegal`
  - 张数不为正整数，输出 `Ticket number illegal`
  - 张数大于该车次该席位的剩余票数，输出 `Ticket does not enough`

## 添加以下新功能

### • 电子钱包功能

- 在现有功能基础上添加电子钱包功能，请根据实际自行选择合适的实现方式
- 精度不做特殊要求，先取小数点后两位，再进行加减即可
- 新增属性：钱包余额
- 新增命令
  - 充值余额

命令	参数1	功能描述
rechargeBalance	充值金额	充值余额

- 输出
  - 充值成功输出 Recharge Success
- 异常处理
  - 命令不存在，输出 Command does not exist
  - 参数数量不正确，输出 Arguments illegal
  - 当前无用户登录，输出 Please login first
  - 充值金额保证为数字类型，若为负数，输出 Arguments illegal
- 查询余额

命令	功能描述
checkBalance	查询余额

- 输出
  - 保留小数点后两位

```
[+]checkBalance
UserName:SB_DYY
Balance:19198.10
```

- 异常处理
  - 命令不存在，输出 Command does not exist
  - 参数数量不正确，输出 Arguments illegal
  - 当前无用户登录，输出 Please login first

### • 核酸检测功能

- 购买车票坐席为[1A, 2A]时需要保证系统中最新的核酸检测证明为阴性，若证明为阳性或不存在则无法购票
- 实现导入证明的功能，核酸检测证明从系统外部导入，保证Aadhaar及检测结果格式正确
- 若多次导入结果，进行更新操作。总阳性人数和总阴性人数中的【总人数】表示【当前有证明的全部人员数量】
- 多次导入的不同文件中，若有相同的Aadhaar卡号，则为此人保留最后一次导入的文件中的核酸检测结果

- 创建核酸证明类 `cert`, 添加如下属性和方法
  - 属性: 自行设计实现相关数据的存储。
  - 方法: 从CSV数据表批量导入证明 (超级管理员模式)

导入乘客核酸检测证明的命令格式如下:

命令	参数1	功能描述
<code>importCert</code>	CSV文件名 (带.csv扩展名)	机构导入检测结果

CSV文件格式如下 (表格无标题), 第一列为Aadhaar卡号 (保证格式正确), 第二列为最新的检测结果 (P为阳, N为阴), 文件下载地址和文件读取demo详见[其他部分](#)

002301220991	P
100000720000	N

- 输出格式:
  - 导入成功输出 `Import Success, Positive:[总阳性人数] Negative:[总阴性人数]`
- 异常处理:
  - 命令不存在, 输出 `Command does not exist`
  - (命令中) 参数数量不正确, 输出 `Arguments illegal`
- 例:

```
[+] importCert cert.csv
[-] Import Success, Positive:1 Negative:1
```

## • 退票功能

用户购买车票后可进行退票, 实现以下命令:

命令	参数1	参数2	参数3	参数4	参数5	功能描述
<code>cancelOrder</code>	列车车次	出发站	目的站	席位代号	张数	用户退票

- 说明
  - 退票规则
    - 相同出发站、目的站、席位的车票, 优先从最近一次的订单中退票, 最后一个订单张数不足再去寻找其他符合条件的订单进行退票。
    - 退票数小于订单张数, 仅修改该订单张数, 订票的顺序不发生变化; 当一笔订单中的车票数量为0时, 应当删除这个订单
    - 已确认支付的订单不予退票
    - 退票后应当立即释放席位 (不考虑线程安全问题)。
  - 保证命令中列车车次、出发站、目的站、席位代号、张数均合法, 不需额外判断
  - 如:

```
// 退票顺序
[+]buyTicket G1001 Shahe Gao limen SB 50
Thanks for your order
[+]buyTicket G1001 Shahe Hamazhen SB 20
```

```
Thanks for your order
[+]buyTicket G1001 Shahe Gaolimen SB 20
Thanks for your order
[+]listOrder
[G1001: Shahe->Gaolimen] seat:SB num:20 price:12528.00 paid:F
[G1001: Shahe->Hamazhen] seat:SB num:20 price:1872.00 paid:F
[G1001: Shahe->Gaolimen] seat:SB num:50 price:31320.00 paid:F
[+]cancelOrder G1001 Shahe Gaolimen SB 11
Cancel success
[+]listOrder
[G1001: Shahe->Gaolimen] seat:SB num:9 price:5637.60 paid:F
[G1001: Shahe->Hamazhen] seat:SB num:20 price:1872.00 paid:F
[G1001: Shahe->Gaolimen] seat:SB num:50 price:31320.00 paid:F
[+]cancelOrder G1001 Shahe Gaolimen SB 4
Cancel success
[+]listOrder
[G1001: Shahe->Hamazhen] seat:SB num:20 price:1872.00 paid:F
[G1001: Shahe->Gaolimen] seat:SB num:19 price:11901.60 paid:F
```

```
// 已支付订单不予退票
[+]listOrder
[G1001: Shahe->Gaolimen] seat:SB num:20 price:12528.00 paid:F
[G1001: Shahe->Gaolimen] seat:SB num:50 price:31320.00 paid:T
[+]cancelOrder G1001 Shahe Gaolimen SB 21
No enough orders
[+]listOrder
[G1001: Shahe->Gaolimen] seat:SB num:20 price:12528.00 paid:F
[G1001: Shahe->Gaolimen] seat:SB num:50 price:31320.00 paid:T
```

- 输出格式

若退票成功，输出

```
Cancel success
```

- 异常处理

- 命令不存在，输出 `Command does not exist`
- 参数数量不正确，输出 `Arguments illegal`
- 当前无用户登录，输出 `Please login first`
- 未找到匹配的购票信息，输出 `No such Record`
- 退票张数大于已购张数，输出 `No enough orders`

- 结算功能

- 新增结算**全部**订单命令，从电子钱包中扣除余额并修改订单结算状态。一旦扣除后，无法再进行取消订单操作。
- 学生票优惠从最近一笔订单开始抵扣，直到资格耗尽为止，耗尽后其余车票按正常价格计算。购票成功后扣除相应的优惠次数

例如：正常票价500，分两次购买，每次购买3张，共有4张优惠资格，则总应付价格为  $(500*4) * 0.05 + 500*2 = 1100$

- 订单是结算的最小单位，即只能按照整笔订单进行结算，订单的结算状态只有T和F。

- 命令格式如下：

命令	功能描述
payOrder	结算订单

- 输出格式
  - 若结算成功，则输出 `Payment success`
- 异常处理
  - 命令不存在，输出 `Command does not exist`
  - 参数数量不正确，输出 `Arguments illegal`
  - 当前无用户登录，输出 `Please login first`
  - 当前无订单，输出 `No order`
  - 电子钱包余额不足以结算全部订单，输出 `Balance does not enough`
- 说明：在订单的添加、取消和支付部分，我们采用了【栈】的设计思想。可以依此来设计相关的数据结构（仅作参考，不考察具体实现方式）

## 其他

cert.csv文件下载地址:

<https://bhpan.buaa.edu.cn:443/link/AED2B7D2568D3AAE55D84895D487EA95> 有效期限：2022-06-02 23:59

评测时，请将CSV文件放到patpat可执行文件相同路径下。例如：

```
CTS\example
| cert.csv
| cert2.csv
| cert3.csv
| patpat-windows-amd64.exe
|
├─4-学号-姓名
|   ├out
|   └src
|       Test.java
|
└─test
    cts2.yaml
    judge.yaml
```

提交代码时，请保持以上目录结构。

**不要使用excel等软件修改CSV文件**，可能会导致格式混乱。

请保持良好的代码风格以及合理的类封装，下一次实验中可能会添加序列化相关内容。

**请注意：PatPat仅支持单个Scanner对象，使用Scanner读入文件可能导致测评结果错误，请采用其他方式读入！**

举个栗子(使用BufferedReader)

```
public static HashMap<String, Boolean> read(String fileName) throws
IOException {
    HashMap<String, Boolean> cert = new HashMap<>();
    BufferedReader br = new BufferedReader(new FileReader(fileName));
    String line;
    while ((line = br.readLine()) != null) {
        String[] lines = line.split(",");
        cert.put(lines[0], lines[1].equals("P"));
    }
    br.close();
    return cert;
}
```