



Курсов проект по ООП

*Специалност „Информационни системи“
Летен семестър 2012/2013
23.4.2013 г.*

Система за управление на влакове (Етап 1)

Увод

Вашата задача е да напишете приложение, което да позволява да се проверява разписанието и да се запазват места за пътуване с влак.

Входни файлове

Вашето приложение трябва да работи със следните четири файла:

cities.txt

Това е текстов файл, в който на отделни редове са дадени имената на градове, до които може да се осъществяват превози. Програмата ви трябва да позволява да се работи само и единствено с градовете, които се съдържат в този файл. Така например ако трябва да се въведе нов маршрут, той трябва да е между два от включените във файла градове. Вдясно е дадено съдържанието на един примерен файл `cities.txt`.

Plovdiv
Varna
Sofia
Burgas
Lom

trains.txt

Това е текстов файл, в който на отделни редове са описани влаковете, с които разполагаме в момента. Информацията за всеки влак е на отделен ред и съдържа:

- Идентификатор на влака
- Брой места
- Кратко описание

Отделните части на информацията са разделени помежду си със запетаи. Описанието е последно и завършва в края на реда. В началото и края на всеки ред, а също и около запетаите може да има произволен брой празни символи (whitespace). Например:

t01,	100,	There can be no better train!
train02,	90,	Riding on this one will be quite fun.
train03,	110,	Plenty of space in the wagons...if you are a hobbit
train04,	70,	We have a fourth train?

schedule.txt

Това е текстов файл, в който са описани разписанията, по които се движат влаковете. Всеки запис е на отделен ред и съдържа следната информация:

- Начален град и краен град
- Дата, на която ще пътува влакът
- Час на тръгване.
- Идентификатор на влак, който обслужва линията.
- Цена на билета.

Отделните части на информацията във всеки запис са разделени със запетаи. Между началния и крайния град се поставят символите „->“.

При попълването на файла, потребителят може да остави произволен брой празни символи (whitespace) в началото и края на реда, а също и между запетаите. Например по-долу са дадени два различни варианта, по които може да се представи една и съща информация:

```
Sofia -> Plovdiv, 01.05.2013, 07:00, t01, 15.99
Plovdiv -> Varna, 01.05.2013, 12:00, train02, 15.00
Plovdiv -> Burgas, 01.05.2013, 12:00, train03, 15.00
Plovdiv -> Sofia, 01.05.2013, 12:00, t01, 15.99
Sofia -> Lom, 11.11.2013, 07:00, train04, 12.00
```

```
Sofia -> Plovdiv, 01.05.2013, 07:00, t01, 15.99
Plovdiv -> Varna, 01.05.2013, 12:00, train02, 15.00
Plovdiv -> Burgas, 01.05.2013, 12:00, train03, 15.00
Plovdiv -> Sofia, 01.05.2013, 12:00, t01, 15.99
Sofia -> Lom, 11.11.2013, 07:00, train04, 12.00
```

```
Sofia -> Plovdiv, 01.05.2013, 07:00, t01, 15.99
Plovdiv -> Varna, 01.05.2013, 12:00, train02, 15.00
Plovdiv -> Burgas, 01.05.2013, 12:00, train03, 15.00
Plovdiv -> Sofia, 01.05.2013, 12:00, t01, 15.99
Sofia -> Lom, 11.11.2013, 07:00, train04, 12.00
```

Функционалност

Вашето приложение трябва да позволява на потребителя да извършва следните дейности:

Относно градовете

1. Извеждане на екрана на списък на градовете
2. Добавяне на град
3. Изтриване на град

Относно влаковете

1. Извеждане на екрана на списък на влаковете.
2. Добавяне на запис за влак.
3. Редактиране на вече съществуващ запис за влак.
4. Изтриване на влак.

Относно разписанията

1. Извеждане на екрана на всички разписания.
2. Добавяне на запис за едно пътуване.
3. Редактиране на вече съществуващ запис за пътуване.
4. Изтриване на запис.
5. Търсене на всички възможности за пътуване между два града. Потребителят трябва да може да въведе начален и краен град и системата да изведе всички възможности за пътуване между двата града (час на тръгване, идентификатор на влак и цена на билет).

Продажба на билети

Информацията за продадените билети трябва да се съхранява в двоични файлове. Те трябва да се съхраняват в директорията на вашата програма или в подходяща поддиректория. За всеки запис в schedule.txt трябва да се поддържа отделен двоичен файл. Сами изберете подходящ начин, по който да именувате вашите файлове, така че да можете да направите връзката с разписанията.

Двоичният файл има толкова на брой елемента, колкото са местата във влака. Всеки елемент съответства на едно място, като първият елемент (на индекс 0) съответства на първото място във влака, вторият (с индекс 1) – на второто място и т.н.

Елементите в двоичния файл съответстват на следната структура:

```
struct Ticket {  
    bool IsReserved; // Дали даденото място е заето (true)  
                    // или не (false)  
  
    time_t SoldOn; // Кога е бил закупен билетът  
  
    double Price;   // Цена, на която е бил продаден билетът  
};
```

Приложението ви трябва да поддържа следната функционалност:

1. **Продажба на билет.** Потребителят избира по коя от въведените в разписанието линии иска да пътува и след това системата извежда списък на свободните места във влака. (всяко място има номер от едно до N). Потребителят избира местата, които иска да

заеме (например 5,6 и 7) и след това системата ги маркира като заети в двоичния файл. Ако потребителят избере места, които не са свободни, системата извежда съобщение за грешка и го кара да избере отново. Системата трябва автоматично да попълни полетата за време на закупуване на билета и цената, на която той е бил продаден (т.е. те не се въвеждат на ръка от потребителя, а се взима текущото време в системата и цената на пътуването, която е указана в schedule.txt).

2. **Извеждане на информация за дадено пътуване.** Потребителят избира една от въведените в разписанието линии. След това системата извежда следната информация за нея: кои места са заети, кои места са свободни, колко са приходите от продадените билети, между кои дати са били продадени билетите.

Бележки

1. Решението трябва да бъде обектно-ориентирано.
2. Всички описани по-горе операции трябва да се извършват директно във файловете. Т.е. ако трябва да добавите нов запис или да изтриете съществуващ, това трябва да се направи директно във файла.
3. Приложението ви трябва да осигурява поддържането на референциална цялост на информацията във файловете. В рамките на проекта ще трябва да се подситеgurите, че:
 - a. Ако потребителят промени идентификатора на влак, който участва в един или повече маршрути, приложението ви трябва да актуализира и съответните записи във файла с маршрутите.
 - b. Приложението ви не трябва да позволява на потребителя да изтрива информация от файл X, която се реферира от запис в друг файл Y. Например не трябва да позволявате да се изтрива град, който участва в маршрут или влак, който участва в маршрут. Също така не може да изтривате маршрут, за който има продадени билети.
4. Достатъчно е вашето приложение да търси само в маршрутите, които са изрично описани в routes.txt. Така например според дадените по-горе файлове има възможност да се стигне от София до Варна с два влака (София -> Пловдив и Пловдив -> Варна). Въпреки това, няма нужда вашето приложение да може да открива такива варианти за пътуване.