Homework 5: Train Schedule

1. Objective

Between the two largest cities of Russia, St. Petersburg and Moscow make daily trips of N trains. For each train, it is known its departure time from St. Petersburg and the time of arrival in Moscow. You need to find the fastest train and its speed. It is guaranteed that the fastest train is determined uniquely.

In this project you need to use:

1. Objects
2. Arrays
3. File Input
4. User Interface

In this homework, you need to use the user interface client code for working with your supplier classes. See the reference on the page.

1. Input Data Files and storage

The text file INPUT.TXT, your program will read from, follows this format:

1. start with one integer on its own line represents the distance between two cities
2. next line – one integer (the number of trains in the everyday schedule 1 ≤ N ≤ 100)
3. followed by a series of lines that describe every train. This description contains 3 pieces of information:
   * **a train name** - a string with a length of 1 to 50 characters. It can contain letters of the English alphabet, spaces, numbers, and dashes ("-"). Lowercase and uppercase letters in the names of trains vary.
   * **time of departure and arrival** is indicated in a 24-hour format as follows HH MM. The format specifies hours (2 digits) and minutes (2 digits) separated by a space. The journey time for each of the trains is at least one minute and does not exceed 24 hours. Pay attention to this example, if the train departs at 19:50 and arrives at 8:30, it means that it arrives the next day and the time of journey is 12 hours and 40 minutes.

Your program does not have to be responsible for files that do not match this format (in other words, if the end user gives you a filename with bad data and the program crashes, that's ok). You can create any text file you want for testing (use a program like Notepad or any other basic text editor). Here is a file I've created for you to use: trains.txt.

1. Output

Print the name of the fastest train and its speed. Speed output in kilometers per hour and round to the nearest whole by mathematical rules. Follow the output format shown in the examples.

1. Code Specification

Implement the class specifications below. To get full credit, your program's public interface must match these descriptions **exactly**.

**Objects You’ll Create**

Here are UML Class Diagrams for the objects you are to create. Pay attention to the diagram notation indicating whether methods are public (+) or private (-); ask questions if you need clarification. Understanding the model is of *critical* importance here.

|  |  |  |
| --- | --- | --- |
| **Class: Time** |  | **Class: Train** |
| **Properties**  *(you figure out the private data needed)* |  | **Properties**  (*you figure out the private data needed*) |
| **Constructor**  +Time(hour : Integer, minute : Integer)  **Accessors**  + getHour() : Integer  + getMinute() : Integer  **Mutators**  +setHour(Integer)  +setMinute(Integer)  + timeBetween(Time) : Time  + minBetween(Time) : Integer  + zeroTime(Integer) : String  + toString() : String |  | **Constructor**  +Train(name : String, departure : Time, arrival: Time, distance : Integer)  **Accessors**  +getDeparture () : Time  +getArrival () : Time  +getDistance () : Integer  **Mutators**  +setDeparture(Time)  +setArrival(Time)  +setDistance(Integer)  +averageSpeed() : Integer  +travelTime() : Time  +toString() : String |

1.The method **zeroTime**(Integer) uses for formatting the output as follows:

Hour = 8, minute = 5. Output should be 08:05

2. The method **minBetween**(Time) returns the difference between two Time objects in minutes. It will be needed for sorting trains using departure time.

|  |
| --- |
| **Class: Schedule** |
| **Properties**  - trains : Train[]  - distance : Integer |
| **Constructor**  +Schedule (fileName : String)  **Methods**  + fillArray(fileName :String)  + fastestTrain() : Train  + sortDeparture()  + toString() : String |

* + - 1. The method **fillArray(fileName** :**String)** reads data from the file and store in the Train[] array.
      2. The method **sortDeparture()** sorts an array of trains using departure time.

1. Suggestions

* I hope by this point in the quarter you appreciate the benefit of working pieces of your solution one at a time. I recommend building and testing your Time and Train classes first. Next work on reading the data (the Schedule class method). Then tackle one or two methods at a time.
* If it makes sense to decompose any of these methods, do so. However, the helper methods should all be private. Only these methods listed here should be part of the public interface.
* Also, when dealing with the **FileNotFoundException**, every method that calls the Schedule constructor needs the throws clause in its method signature (unless you want to use try/catch blocks in any client code, but that is not a requirement for this assignment).

1. Documentation and Style
2. Make sure to write complete Javadoc comments for each class and each public method.
3. Include sufficient internal documentation.
4. Use appropriate style (variable names, indenting, class constants, etc.) throughout the program.
5. Be wise in your choice of instance variables. Only the data that an object needs to know (and remember) should be an instance variable.

Example of the output

This is an example of the final trains schedule output after sorting by the departure time.

The last part includes the information about the fastest train.

Budget

Departure 05:50

Arrival 18:55

Travel Time 13:05

Average speed 50 km/h

Star

Departure 10:30

Arrival 19:25

Travel Time 08:55

Average speed 73 km/h

Comfort

Departure 11:45

Arrival 20:40

Travel Time 08:55

Average speed 73 km/h

Day Rocket

Departure 11:55

Arrival 16:00

Travel Time 04:05

Average speed 159 km/h

Super Star

Departure 12:20

Arrival 16:48

Travel Time 04:28

Average speed 146 km/h

Best Choice

Departure 20:00

Arrival 07:25

Travel Time 11:25

Average speed 57 km/h

Good Morning

Departure 21:20

Arrival 08:45

Travel Time 11:25

Average speed 57 km/h

Sunrise

Departure 21:30

Arrival 05:40

Travel Time 08:10

Average speed 80 km/h

Red Arrow

Departure 22:10

Arrival 07:45

Travel Time 09:35

Average speed 68 km/h

Night Express

Departure 23:55

Arrival 08:15

Travel Time 08:20

Average speed 78 km/h

The fastest train:

Day Rocket

Departure 11:55

Arrival 16:00

Travel Time 04:05

Average speed 159 km/h