Project 11: Air Ticket App w/Exceptions

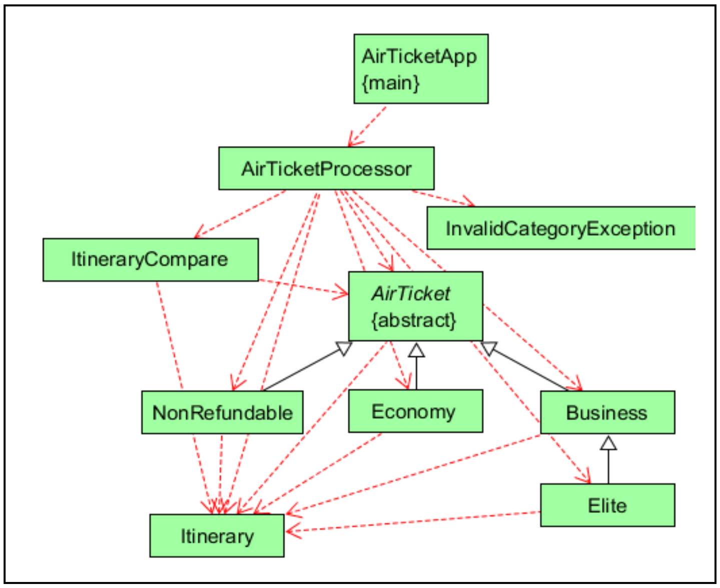
Due: **Wednesday, November 28, 2012 by 11:59 p.m.**

Deliverables

The following project files must be submitted to the "graded" assignment by the due date and time specified above (see the Lab Guidelines for information on submitting project files). You should plan to start on the project not later than Wednesday in lab. To ensure that your classes and methods are named correctly, you should make sure that your file is successfully submitted to the "ungraded" assignment in Web-CAT during lab on Wednesday. **Projects sent via e-mail past the deadline at 11:59 PM will not be accepted without a university-approved excuse.**

Files to submit to Web-CAT:

- Itinerary.java
- AirTicket.java
- NonRefundable.java
- Economy.java
- Business.java
- Elite.java
- ItineraryCompare.java
- AirTicketProcessor.java
- AirTicketApp.java
- InvalidCategoryException.java



Recommendations

You should create new folder for Project 11 and copy your Project 10 source and test files to it. I recommend that you create a JUnit test file for each class and write one or more test methods to ensure the classes and methods meet the specifications. You should create a jGRASP project and add the class and test files as they are created. You may find it helpful to use the new "viewer canvas" feature in jGRASP 2.0.0 as you develop and debug your program.

Specifications

Overview: Project 11 is the <u>third</u> of three projects that will involve the pricing and reporting of air tickets. In Project 9, you developed Java classes that represent an itinerary and categories of air tickets including non-refundable, economy, business, and elite, all of which include an itinerary. In Project 10, you modified the AirTicket class and created three additional classes: (1) AirTicketProcessor that includes a method to read in a ticket data file and methods to generate several reports, (2) IntineraryCompare which implements the Comparator interface for AirTicket; and (3) AirTicketApp which includes the main method for the program. In Project 11, you are to add exception handing and invalid input reporting. You will need to do the following: (1) create a new class named InvalidCategoryException which extends the Exception class, (2) add try-catch statements to catch FileIOException, InvalidCategoryException, and NumberFormatException, and (3) add three methods to the AirTicketProcessor class.

- **Itinerary, AirTicket, NonRefundable, Economy, Business, Elite, ItineraryCompare,**

  Requirements and Design: No changes from the specifications in Project 10

- **InvalidCategoryException.java**

  Requirements and Design: InvalidCategoryException is a user defined exception created by extending the Exception class. This exception is to be thrown and caught in the readAirTicketFile method in the AirTicketProcessor class when a line of input data contains an invalid category. The constructor for InvalidCategoryException takes a single String parameter representing *category* and invokes the super constructor with the following String:

  ```
  "For category: " + "\"" + category + "\""
  ```

  This string will be the toString() value of an InvalidCategoryException when it occurs. For a similar constructor, see InvalidLengthException.java in 11_Exceptions\Examples\Polygons from this week's lecture notes.

- **AirTicketProcessor.java**

  Requirements and Design: In addition to the specifications in Project 10, three new methods, getTickets(), getInvalidInput(), and generateReportForInvalidInput(), should be implemented in the AirTicketProcessor class, and the existing readAirTicketFile method must be modified to recognize invalid input data and catch InvalidCategoryException and NumberFormatException.

  - `getTickets`: Accepts no parameters and returns the value of the field for the ArrayList of AirTickets (i.e., the return type is ArrayList<AirTicket>). Note that this method is primarily to facilitate testing.
  - `getInvalidInput`: Accepts no parameters and returns the value of the field for the ArrayList containing the invalid input read from the file (i.e., the return type is ArrayList<String>). Note that this method is primarily to facilitate testing.
  - `generateReportForInvalidInput:` Has no return value and processes the invalid input ArrayList to produce the "Air Ticket Report for Invalid Input" as shown below near the end in the Example Output.
  - `readAirTicketFile:` Has no return value and accepts the data file name as a String, and throws IOException. This method creates a Scanner object to read in the file and then reads it in line by line. I recommend that you set up a second scanner on the line to scan it. For each *correct* line scanned, you will create the "appropriate" AirTicket object and add it to the ArrayList of AirTicket field. Each line in the file begins with a category for the ticket (N, E, B, and F are valid categories for ticket class indicating **N**onRefundable, **E**economy, **B**usiness, and **F**irst (or Elite) respectively; note that **F** (First) will indicate our Elite class. For each *incorrect* line scanned (i.e., a line of data contains an invalid category or invalid numeric data), your method will need to handle the invalid items properly. If the line of data begins with an invalid category, your program should throw an InvalidCategoryException (see description above). If a line of data has a valid category, but includes invalid numeric data (e.g., the value for *miles* contains an alphabetic character), a NumberFormatException will be thrown automatically by the Java Runtime Environment (JRE). Your readAirTicketFile method should catch and

handle InvalidCategoryException and NumberFormatException as follows. In each catch clause, a String object should be created consisting of

```
       e + " in: " + line
```

where *e* is the exception and *line* is the line with the invalid data. The String object should be added to the ArrayList of invalid input.

- **AirTicketApp.java**

  Requirements and Design: The AirTicketApp class contains the main method for running the program. In addition to the specifications in Project 10, the main method should be modified in two ways.
  (1) A try-catch statement should be added to catch IOException when/if thrown by the readAirTicketFile method in the AirTicketProcessor class. This exception is most likely to occur when an incorrect file name is passed to the readAirTicketFile method. After this exception is caught and printed, your program should loop back and ask the user to enter the file name again. The loop should end only after the readAirTicketFile method completes normally.
  (2) In addition to the three reports in Project 10, it should also print the "Air Ticket Report for Invalid Input" as shown in the "Example Output" beginning on the next page. Note that the lines in the Invalid Input Report are shown "wrapped". However, in your output, these lines should not be wrapped. An example data file (air_ticket_data2.csv) can be downloaded from the Lab web page.

Example Output

```
 ----jGRASP exec: java AirTicketApp

Enter File Name: air_ticket_data2.csv

----------------------------
Air Ticket Report
----------------------------

Flight: DL 1865
ATL-LGA (2012/11/01 1400 - 2012/11/01 1640) 800 miles (1600 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 2.0
Total Fare: $1,000.00 (class Business)
   Includes Food/Beverage: $50.00 Entertainment: $50.00

Flight: DL 1867
ATL-LGA (2012/11/01 1500 - 2012/11/01 1740) 800 miles (1200 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 1.0
Total Fare: $450.00 (class Economy)
   Includes Award Miles Factor: 1.5

Flight: DL 1863
ATL-LGA (2012/11/01 0900 - 2012/11/01 1140) 800 miles (1600 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 2.5
Total Fare: $1,325.00 (class Elite)
   Includes Food/Beverage: $50.00 Entertainment: $50.00
   Includes: Comm Services: $100.00

Flight: DL 1861
ATL-LGA (2012/11/01 0800 - 2012/11/01 1040) 800 miles (800 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 0.45
Total Fare: $182.25 (class NonRefundable)
   Includes DiscountFactor: 0.9

Flight: DL 1866
LGA-ATL (2012/11/01 1400 - 2012/11/01 1640) 800 miles (1600 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 2.0
Total Fare: $1,000.00 (class Business)
   Includes Food/Beverage: $50.00 Entertainment: $50.00

Flight: DL 1868
LGA-ATL (2012/11/01 1500 - 2012/11/01 1740) 800 miles (1200 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 1.0
Total Fare: $450.00 (class Economy)
   Includes Award Miles Factor: 1.5

Flight: DL 1864
LGA-ATL (2012/11/01 0900 - 2012/11/01 1140) 800 miles (1600 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 2.5
Total Fare: $1,325.00 (class Elite)
   Includes Food/Beverage: $50.00 Entertainment: $50.00
   Includes: Comm Services: $100.00

Flight: DL 1862
LGA-ATL (2012/11/01 0800 - 2012/11/01 1040) 800 miles (800 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 0.45
Total Fare: $182.25 (class NonRefundable)
   Includes DiscountFactor: 0.9
```

```
--------------------------------------
Air Ticket Report (by Flight Number)
--------------------------------------

Flight: DL 1861
ATL-LGA (2012/11/01 0800 - 2012/11/01 1040) 800 miles (800 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 0.45
Total Fare: $182.25 (class NonRefundable)
   Includes DiscountFactor: 0.9

Flight: DL 1862
LGA-ATL (2012/11/01 0800 - 2012/11/01 1040) 800 miles (800 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 0.45
Total Fare: $182.25 (class NonRefundable)
   Includes DiscountFactor: 0.9

Flight: DL 1863
ATL-LGA (2012/11/01 0900 - 2012/11/01 1140) 800 miles (1600 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 2.5
Total Fare: $1,325.00 (class Elite)
   Includes Food/Beverage: $50.00 Entertainment: $50.00
   Includes: Comm Services: $100.00

Flight: DL 1864
LGA-ATL (2012/11/01 0900 - 2012/11/01 1140) 800 miles (1600 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 2.5
Total Fare: $1,325.00 (class Elite)
   Includes Food/Beverage: $50.00 Entertainment: $50.00
   Includes: Comm Services: $100.00

Flight: DL 1865
ATL-LGA (2012/11/01 1400 - 2012/11/01 1640) 800 miles (1600 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 2.0
Total Fare: $1,000.00 (class Business)
   Includes Food/Beverage: $50.00 Entertainment: $50.00

Flight: DL 1866
LGA-ATL (2012/11/01 1400 - 2012/11/01 1640) 800 miles (1600 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 2.0
Total Fare: $1,000.00 (class Business)
   Includes Food/Beverage: $50.00 Entertainment: $50.00

Flight: DL 1867
ATL-LGA (2012/11/01 1500 - 2012/11/01 1740) 800 miles (1200 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 1.0
Total Fare: $450.00 (class Economy)
   Includes Award Miles Factor: 1.5

Flight: DL 1868
LGA-ATL (2012/11/01 1500 - 2012/11/01 1740) 800 miles (1200 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 1.0
Total Fare: $450.00 (class Economy)
   Includes Award Miles Factor: 1.5

--------------------------------------
Air Ticket Report (by Itinerary)
--------------------------------------

Flight: DL 1861
ATL-LGA (2012/11/01 0800 - 2012/11/01 1040) 800 miles (800 award miles)
Base Fare: $450.00 Fare Adjustment Factor: 0.45
```

```
    Total Fare: $182.25 (class NonRefundable)
       Includes DiscountFactor: 0.9

    Flight: DL 1863
    ATL-LGA (2012/11/01 0900 - 2012/11/01 1140) 800 miles (1600 award miles)
    Base Fare: $450.00 Fare Adjustment Factor: 2.5
    Total Fare: $1,325.00 (class Elite)
       Includes Food/Beverage: $50.00 Entertainment: $50.00
       Includes: Comm Services: $100.00

    Flight: DL 1865
    ATL-LGA (2012/11/01 1400 - 2012/11/01 1640) 800 miles (1600 award miles)
    Base Fare: $450.00 Fare Adjustment Factor: 2.0
    Total Fare: $1,000.00 (class Business)
       Includes Food/Beverage: $50.00 Entertainment: $50.00

    Flight: DL 1867
    ATL-LGA (2012/11/01 1500 - 2012/11/01 1740) 800 miles (1200 award miles)
    Base Fare: $450.00 Fare Adjustment Factor: 1.0
    Total Fare: $450.00 (class Economy)
       Includes Award Miles Factor: 1.5

    Flight: DL 1862
    LGA-ATL (2012/11/01 0800 - 2012/11/01 1040) 800 miles (800 award miles)
    Base Fare: $450.00 Fare Adjustment Factor: 0.45
    Total Fare: $182.25 (class NonRefundable)
       Includes DiscountFactor: 0.9

    Flight: DL 1864
    LGA-ATL (2012/11/01 0900 - 2012/11/01 1140) 800 miles (1600 award miles)
    Base Fare: $450.00 Fare Adjustment Factor: 2.5
    Total Fare: $1,325.00 (class Elite)
       Includes Food/Beverage: $50.00 Entertainment: $50.00
       Includes: Comm Services: $100.00

    Flight: DL 1866
    LGA-ATL (2012/11/01 1400 - 2012/11/01 1640) 800 miles (1600 award miles)
    Base Fare: $450.00 Fare Adjustment Factor: 2.0
    Total Fare: $1,000.00 (class Business)
       Includes Food/Beverage: $50.00 Entertainment: $50.00

    Flight: DL 1868
    LGA-ATL (2012/11/01 1500 - 2012/11/01 1740) 800 miles (1200 award miles)
    Base Fare: $450.00 Fare Adjustment Factor: 1.0
    Total Fare: $450.00 (class Economy)
       Includes Award Miles Factor: 1.5

    --------------------------------------
    Air Ticket Report for Invalid Input
    --------------------------------------

    InvalidCategoryException: For category: "A" in: A,DL 1865,ATL,LGA,2012/11/01 1400,2012/11/01
1640,800,450,2.0,50.0,50.00

    java.lang.NumberFormatException: For input string: "800.0" in: E,DL 1867,ATL,LGA,2012/11/01
1500,2012/11/01 1740,800.0,450,1.0

    java.lang.NumberFormatException: For input string: "$450" in: F,DL 1863,ATL,LGA,2012/11/01
0900,2012/11/01 1140,800,$450,2.5,50.0,50.00,100.00

    java.lang.NumberFormatException: For input string: "0.90%" in: N,DL 1861,ATL,LGA,2012/11/01
```

```
0800,2012/11/01 1040,800,450,0.45,0.90%

    java.lang.NumberFormatException: For input string: "50.x" in: B,DL 1866,LGA,ATL,2012/11/01
1400,2012/11/01 1640,800,450,2.0,50.x,50.00

    java.lang.NumberFormatException: For input string: "1/0" in: E,DL 1868,LGA,ATL,2012/11/01
1500,2012/11/01 1740,800,450,1/0

    java.lang.NumberFormatException: For input string: "100.oo" in: F,DL 1864,LGA,ATL,2012/11/01
0900,2012/11/01 1140,800,450,2.5,50.0,50.00,100.oo

     ----jGRASP: operation complete.
```

Due to space constraints, the lines in the Invalid Input Report are shown “wrapped” in the example above. However, in your output, these lines should not be wrapped.