Airline Company Email Automation

Assignment Goals

- To master I/O
- To work with regular expressions
- To practice one class/ one responsibility
- To propose design of Emergency Room simulator and implement it using Priority Queue

Background:

OnTime airlines is a company that values its customers. Every time when there is a delay in departure or arrival a company sends apologies to the passengers of that flight. **You goal is to to automate this process.**

The airline company has all the information of its members in a CSV file. **CSV file** is a plain text file that contains data, such that each piece of data is **enclosed in double quotes and separated by a comma**. The first line of the file contains the headers for each column. For example:

```
"first_name" , "last_name" , "phone"
"James"      , "Butt"      , "504-845-1427"
"Josephine"  , "R, Darakjy", "810-374-9840"
"Art"        , "Venere"    , "856-264-4130"
```

The listing above has 3 columns named `first_name`, `last_name`, and `phone`. The first line contains the headers, while all other lines have the information for a specific passenger. Note that comma serves NOT only as a delimiter, there may be pieces of information with comma, for example `"R, Darakjy"` is one piece of information and not two.

Your task:

Attached to this assignment is a sample of Flight CSV file, which contains first and last names, address, city, country, state, zip, phone, email address and rewards membership. Given such CSV file and a template e-mail (see example below), your task is to **create a program that runs on the command line and generates files that will contain the email messages to all the passengers found in the supplied CSV files.** Templates are stored as text files with **special placeholders** in the text. Placeholders are placed between [[and]]. The placeholders may refer to:

1. CSV file's header names
2. Parameters supplied as an input (see explanation below)
3. Some general information that can be inferred from a system (for example, date of the e-mail).

Please see an example template of an e-mail message on the next page.

```
[[Date]]


To:[[email]]
Subject: Please accept our apologies for the [[event]] of your flight
Dear [[first_name]] [[last_name]],


We are very sorry for the [[event]] of your flight. As a valued [[rewards
status] member of our club your time is important to us and we will strive to
improve our service and in the future, and make it on time!


Sincerely,
OnTime airline customer service
```

Given the above email template, we need to substitute information in placeholders to create an e-mail:

- Some of the information is part of lines in the CSV file. For example:

```
"first_name"    , "last_name"   , "email"               , "rewards"
"Art"           , "Venere"      , "a.venere@gmail.com" , "gold”
```

- Some of the information can be inferred from a system (http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html)
- Some of the information will be provided as a parameter/ input to the program, as described below:

Your program needs to accept certain arguments at the command line.

`--email`	generate email messages
`--email-template <file>`	accepts a filename that holds the email template
`--output-dir <path>`	accepts the name of a folder, all output is placed in this folder
`--csv-file <path>`	accepts the name of the csv file to process
`--event <details>`	specifies if the delay refers to departure/arrival

Some options take arguments, for example `--email-template` takes one argument and it is the name of a file, `--output-dir` takes one argument and it is the name of a folder, `--event` takes one string argument “departure”/ “arrival”. Some options take no arguments and indicate an action, i.e., `--email` indicates that we are to generate emails on this execution of the program.

The command line option `--output-dir`, `--csv-file` and `--event` are required. If `--email` is given then `--email-template` must also be provided. Calling your program and passing any other combination of options should generate an error, for example:

`--email --letter-template letter-template.txt --output-dir letters` is illegal.

When an illegal combination of inputs is provided by user**, the program should exit with a helpful error message and a short explanation of how to use the program along with examples.**

For example, if the above input is provided the program should generate:

```
Error: --email provided but no --email-template was given, no event specified
and no csv file provided

Usage:

--email                        generate email messages

--email-template <file>        accepts a filename that holds the email template.
                               Required if --email is used

--output-dir <path>            accepts the name of a folder, all output is placed
                               in this folder

--csv-file <path>              accepts the name of the csv file to process

--event <details>              specifies  if  the  delay  refers  to
                               departure/arrival

For example:

--email  --email-template  email-template.txt  --output-dir  emails  --csv-file
Flight363.csv –-event arrival
```

NOTE the order that the command line options are given does not matter, i.e. the following examples are valid

```
--email  --email-template  email-template.txt  --output-dir  emails  --csv-file
Flight363.csv –-event arrival
```

```
--csv-file  Flight363.csv  --email-template  email-template.txt  --output-dir
emails --email –-event arrival
```

```
–-event arrival --output-dir emails --email --csv-file Flight363.csv --email-
template email-template.txt
```

<u>Summary of the assignment:</u>

- Design and implement the email generator program for the airline company.
- Use the supplied example files to help you develop and test your code. Your code should be flexible enough to work with any CSV file and any template with placeholders specified in [[ ]].
- You should also develop your own templates.
- You should make your program flexible, today you are sending e-mails, tomorrow you might need to send regular letters.
- Make sure that your program works correctly regardless of how your operating system represents paths and files.