Lab 04 – If-Else, Branching, Enumerations

Introduction

This lab builds upon lab 3, using **if-else** and other branching statements to classify short messages (like tweets) based on keywords in the message. This is a task that might reasonably be done in real-world situations. E.g., certain tweets might contain information (e.g., a request for help, a fire sighted) that must be acted on quickly, and computer programs could be used to identify these based upon the text in the message.

In this lab, you will create a Java enumeration called **MessageCategory** inside of **ClassifyMessage** that lists several message categories (**NEED, OFFER, ALERT, INFO, UNKNOWN**). Your program will parse the text of a user specified message, identify the category of the message, and assign this value to a variable (**category**) declared to be of type **MessageCategory** (only values from the enumeration can be assigned to the variable). You will also identify the latitude and longitude specified in the message and determine whether these are within ranges defined elsewhere in the program.

Exercise – Using Enumerations and If-Else statements

Since the primary purpose of this lab is not String manipulation, we will use a simpler message format than that found in lab 3. You will still use the **Scanner** and **String** classes, however.

You may assume that the messages processed by the program all have the following format:

category latitude longitude payload

where *category* is a single keyword indicating the message category (its type in Lab 3), *latitude* and *longitude* are both floating point numbers, and *payload* is a string of text (potentially containing arbitrary characters) constituting the primary body of the message. For instance, the message

```
offer 40.022 -105.226 free essential supplies 4 evacs pets, 2323 55th
st, boulder
```

conforms to the above format. Additional sample messages are provided at the end of this document. You should use them to test your code.

Instructions

1. Create a new class called **ClassifyMessage** (stored in a file called **ClassifyMessage.java**).

2. Declare the enumeration **MessageCategory** by adding the following line:

```
enum MessageCategory {NEED, OFFER, ALERT, INFO, UNKNOWN}
```

- Where does this line of code belong? In the main method? In **ClassifyMessage**? Think about this

- You should declare the following variables in your program:

  - **catString**:   **String**   // The raw text of the message's category
  - **payload**:   **String**   // The primary content of the message
  - **latitude**:   **double**   // The latitude indicated in the message

- o **longitude:** **double** // The longitude indicated in the message
- o **isInRange:** **boolean** // A "flag" indicating whether the latitude and
  // longitude values are within bounds
- o **category:** **MessageCategory** // The message's category

3. Additionally, you should declare the following **double** variables and initialize them to the shown values. These define geographic boundaries that the program uses. Some messages will originate from within those bounds, other messages will not. **Is it a good idea to make these constants?**

- o **south** **double** 39.882343 // *southernmost latitude*
- o **north** **double** 40.231315 // *northernmost latitude*
- o **west** **double** -105.743511 // *westernmost longitude*
- o **east** **double** -104.907864 // *easternmost longitude*

4. After the variables have been declared, write a statement to prompt the user with the following message:

Please enter a formatted message:

5. Use the **keyboard** object's **next()**, **nextDouble()**, and **nextLine()** methods to read in values for **catString**, **latitude**, **longitude**, and **payload**. You may assume that the message is entered as a single line of text and formatted as described earlier.

6. For **payload** (and **catString**, if needed) you should trim (using the **trim()** method of the **String** class, if needed) any leading and trailing white spaces from the text.

7. Use a multi-branch if-else statement to match the value stored in **catString** to one of the elements of the enumeration **MessageCategory**. The conditions should be the following:

   - o If the value of **catString** is one of **"fire"** or **"smoke"**, then **category** should be assigned the value **MessageCategory.ALERT**.

   - o Otherwise, if the value of **catString** is **"need"**, then **category** should be assigned the value **MessageCategory.NEED**.

   - o Otherwise, if the value of **catString** is **"offer"**, then **category** should be assigned the value **MessageCategory.OFFER**.

   - o Otherwise, if the value of **catString** is one of **"structure"**, **"road"**, **"photo"**, or **"evac"**, then **category** should be assigned the value **MessageCategory.INFO**.

   - o Otherwise, **category** should be assigned the value **MessageCategory.UNKNOWN**.

When comparing the strings, you should use the **equalsIgnoreCase** method. Why use the **equalsIgnoreCase** method? Why not use **==**?

8. Use another if-else statement to determine whether the latitude and longitude specified in the message are within the geographic boundaries indicated by **north**, **south**, **east**, and **west**. Variable **isInRange** should be assigned the value **true** if and only if *both* of the below conditions are met. Otherwise, **isInRange** should be assigned the value **false**.

   I. If **latitude ≥ south** and **latitude ≤ north**.

II.   If `longitude ≥ west` and `longitude ≤ east`.

Additional Requirements

These are things that make the graders lives easier, and ultimately, you will see in the real world as well. Remember the teaching staff does not want to touch your code after they gave you requirements; they want to see the perfect results they asked for! Here is a checklist of things you can **lose points** for:

- (-1 point) If the source file(s)/class(es) are named incorrectly (case matters!)
- (-1 point) If your source file(s) have a package declaration at the top
- (-1 point) If any source file you submit is missing your Statement of Academic Honesty at the top of the source file. All submitted source code files must contain your Statement of Academic Honesty at the top of each file.
- (-1 point) If you have more than one instance of Scanner in your program. Your program should only have one instance of Scanner.
- (-1 point) Inconsistent I/O (input/output) that does not match our instructions or examples exactly (unless otherwise stated in an assignment's instructions). Your program's I/O (order, wording, formatting, etc.) must match our examples and instructions.
- (-2 points) If your submission is late
- (-7 points) If the source file(s) are not submitted before the specified deadline's late period ends (48 hours after the deadline) or if they do not compile.
- If your (-1 point) comments or (-1 point) variables are "lacking"
  - Here, "lacking" means that you or a TA can find **any** lines of code or variables that take more than 10 seconds to understand, and there is no comment, or the variable name does not make sense (variable names like **b**, **bb**, **bbb**, etc. **will almost never be acceptable)**
- (-1 point) Indentation is not consistent throughout your source code
  - Refresh your memory of indentation patterns in chapter 2 in the course textbook
  - Be careful of a combination of tabs and spaces in your files (use one or the other)!

If any of the above do not make sense to you, talk to a TA or your lab instructor.

eLC Submission and Grading

After you have completed and thoroughly tested **`ClassifyMessage.java`**, submit it to *eLC* in order to receive credit for the lab. Always double check that your submission was successful on *eLC*!

The lab will be graded according to the following guidelines.

- Students may earn 3 points for lab attendance and 7 points if their program passes various test cases that we use for grading. **Some test cases may use values taken from the examples in this document and some test cases may not**. You should come up with additional test cases to check that your program is bug free.
- All source code must adhere to good Java programming style standards as discussed in lecture class and in the course textbook readings.
- Points will be deducted for not following any of the aforementioned instructions or requirements.

Sample Input

1. smoke 40.499812 -105.012075 its raining ash

2. offer 40.022 -105.226 free essential supplies 4 evacs pets, 2323 55th st, boulder

3. structure 40.029854 -105.391055 damaged: 224 left fork road (shed) (house okay)

4. wind 40.3523 -105.2045 just switched, and now the smoke is thick around our house

5. fire 40.0515 -105.332 firefighter sees a glow: sw from west coach rd toward sunshine canyon

6. structure 40.050904 -105.373941 damaged: unknown number, by 204 gold run road

7. evac 40.383 -105.113 overflow shltr 4 evacuees at walt clark middle school, loveland

8. nofire 40.367 -105.292 activity: west of the pinewood res dam

9. photo 40.052304 -105.319374 local: wild horse

10. info 40.499812 -105.012075 its raining ash: windsor, co

11. need 40.011471 -105.28638 volunteers sun 8a-10p donation ctr 2 help w/donations for victims & firefighters: 3111 28th st

12. need 40.031131 -105.259259 people to help sort donations Sept 12 8am-9pm 3111 28th st

13. open 40.072208 -105.354437 evacuees can return to homes: lee hill

14. open 40.076431 -105.309757 lee hill dr.

Sample Input and Output

These are sample runs of the program. Input is indicated in red. Your input and output should be consistent with what is shown here except for the line longs of input or output (these long lines word wrap in this document, but in your program, these long lines of input or output should be on a single line).

```
Please enter a formatted message:
smoke 40.499812 -105.012075 its raining ash
Category:    ALERT
Raw Cat:     smoke
Message:     its raining ash
Latitude:    40.499812
Longitude:   -105.012075
In Range:    false
```

```
Please enter a formatted message:
offer 40.022 -105.226 free essential supplies 4 evacs pets, 2323 55th st, boulder
Category:    OFFER
Raw Cat:     offer
Message:     free essential supplies 4 evacs pets, 2323 55th st, boulder
Latitude:    40.022
```

```
Longitude:   -105.226
In Range:    true
```

```
Please enter a formatted message:
structure 40.029854 -105.391055 damaged:  224 left fork road (shed) (house okay)
Category:    INFO
Raw Cat:     structure
Message:     damaged:  224 left fork road (shed) (house okay)
Latitude:    40.029854
Longitude:   -105.391055
In Range:    true
```

```
Please enter a formatted message:
wind 40.3523 -105.2045 just switched, and now the smoke is thick around our house
Category:    UNKNOWN
Raw Cat:     wind
Message:     just switched, and now the smoke is thick around our house
Latitude:    40.3523
Longitude:   -105.2045
In Range:    false
```

```
Please enter a formatted message:
fire 40.0515 -105.332 firefighter sees a glow: sw from west coach rd toward sunshine
canyon
Category:    ALERT
Raw Cat:     fire
Message:     firefighter sees a glow: sw from west coach rd toward sunshine canyon
Latitude:    40.0515
Longitude:   -105.332
In Range:    true
```

```
Please enter a formatted message:
structure 40.050904 -105.373941 damaged: unknown number, by 204 gold run road
Category:    INFO
Raw Cat:     structure
Message:     damaged: unknown number, by 204 gold run road
Latitude:    40.050904
Longitude:   -105.373941
In Range:    true
```

```
Please enter a formatted message:
evac 40.383 -105.113  overflow shltr 4 evacuees at walt clark middle school, loveland
Category:    INFO
Raw Cat:     evac
Message:     overflow shltr 4 evacuees at walt clark middle school, loveland
Latitude:    40.383
Longitude:   -105.113
In Range:    false
```

```
Please enter a formatted message:
nofire 40.367 -105.292 activity: west of the pinewood res dam
Category:    UNKNOWN
Raw Cat:     nofire
Message:     activity: west of the pinewood res dam
Latitude:    40.367
Longitude:   -105.292
In Range:    false
```

```
Please enter a formatted message:
photo 40.052304 -105.319374 local: wild horse
Category:    INFO
Raw Cat:     photo
Message:     local: wild horse
Latitude:    40.052304
Longitude:   -105.319374
In Range:    true
```

```
Please enter a formatted message:
info 40.499812 -105.012075 its raining ash: windsor, co
Category:    UNKNOWN
Raw Cat:     info
Message:     its raining ash: windsor, co
Latitude:    40.499812
Longitude:   -105.012075
In Range:    false
```

```
Please enter a formatted message:
need  40.011471 -105.28638 volunteers sun 8a-10p donation ctr 2 help w/donations for
victims & firefighters: 3111 28th st
Category:    NEED
Raw Cat:     need
Message:     volunteers sun 8a-10p donation ctr 2 help w/donations for victims &
firefighters: 3111 28th st
Latitude:    40.011471
Longitude:   -105.28638
In Range:    true
```

```
Please enter a formatted message:
need  40.031131 -105.259259 people to help sort donations  Sept 12 8am-9pm 3111 28th st
Category:    NEED
Raw Cat:     need
Message:     people to help sort donations  Sept 12 8am-9pm 3111 28th st
Latitude:    40.031131
Longitude:   -105.259259
In Range:    true
```

```
Please enter a formatted message:
```

open 40.072208 -105.354437 evacuees can return to homes: lee hill
```
Category:    UNKNOWN
Raw Cat:     open
Message:     evacuees can return to homes: lee hill
Latitude:    40.072208
Longitude:   -105.354437
In Range:    true
```

```
Please enter a formatted message:
```
open 40.076431 -105.309757 lee hill dr.
```
Category:    UNKNOWN
Raw Cat:     open
Message:     lee hill dr.
Latitude:    40.076431
Longitude:   -105.309757
In Range:    true
```