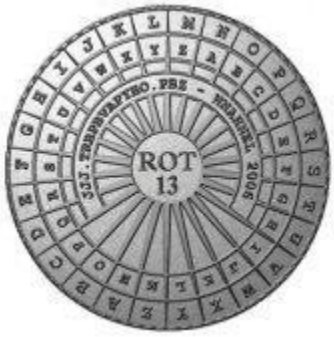
CST 183 Java Programming
Programming Assignment 8 (30 pts)

Purpose To utilize Java strings and Java string handling routines.

Requirements



For this assignment you will create an application that uses geocaching to encrypt secret messages sent to other countries.

Step 1: Understand the requirements.

Write an application to perform text encryption.

You are building a system to send text messages to your overseas embassies. Design you interface with **one text field** and **two text areas**.

The text field will include a two-character code that will define the embassy that is to receive the message. The following list includes all possible countries:

FR - France
GB - Great Briton
CA - Canada
JA - Japan
RU - Russia
GE - Germany
AU - Australia
MX - Mexico

Include error checking to validate that the destination code is among those listed. If the country code is not in your list, provide an error message dialog to the user. Consider parallel arrays of strings for easier management of these strings.

A simple encryption key exists for a very popular activity called geocaching. This key is defined by the following scheme:

A|B|C|D|E|F|G|H| I | J|K|L|M

N|O|P|Q|R|S|T|U|V|W|X|Y|Z

(letter above equals below, and vice versa)

As a test case, if GB is entered in the destination code and the following is entered in the message text area:

THESE ARE THE TIMES THAT TRY MENS SOULS
(quoting Thomas Paine from the American Revolution)

the method of your class should encryt to the following message (send to the second text area):

TO: Great Britain
GURFR NER GUR GVZRF GUNG GEL ZRAF FBHYF

Design the main application driver routine to simply manage the graphical user interface including buttons, text field, and text areas.

Be sure this is an ***object-oriented*** **solution**. The main application driver should send parameters to an **object** of a **class** that should be designed to manage the message, the destination code, and encryption action. The class will store the message and provide the methods for encryption, and destination code validation, and destination code interpretation. Add "set/get" methods as needed. Try to concentrate all user interactions in the driver and all message management actions in this class.

Step 2: Develop Logic Plan

Create a UML diagram depicting your class. Also draw a basic flowchart or write pseudocode that details your logic plan for solving this problem and helps you plan the program.

Step 3: Code It

Use jGRASP or your IDE to write the Java program that addresses the assignment specifics.

Type in your program code

1. **Compile** your program and correct all the syntax errors
2. **Run** your program and evaluate the output for correctness

Deliverables Submit the Java source code file(s) (the class definition(s) and driver .java files you completed) to the **Program 11 Dropbox** within the Delta eLearning System.

Create a hardcopy for turn-in and grading containing the following:

a) Title Page
b) UML diagram(s).
c) Flow chart or pseudocode for program
d) Source code for all class files and driver. (copy and paste from your IDE)
e) Output (copy and paste from your IDE and/or do a screen capture to get the output window).