Joanna Klukowska
joannakl@cs.nyu.edu

Assignment 3

Due date: Feb. 22, 11:55PM EST.

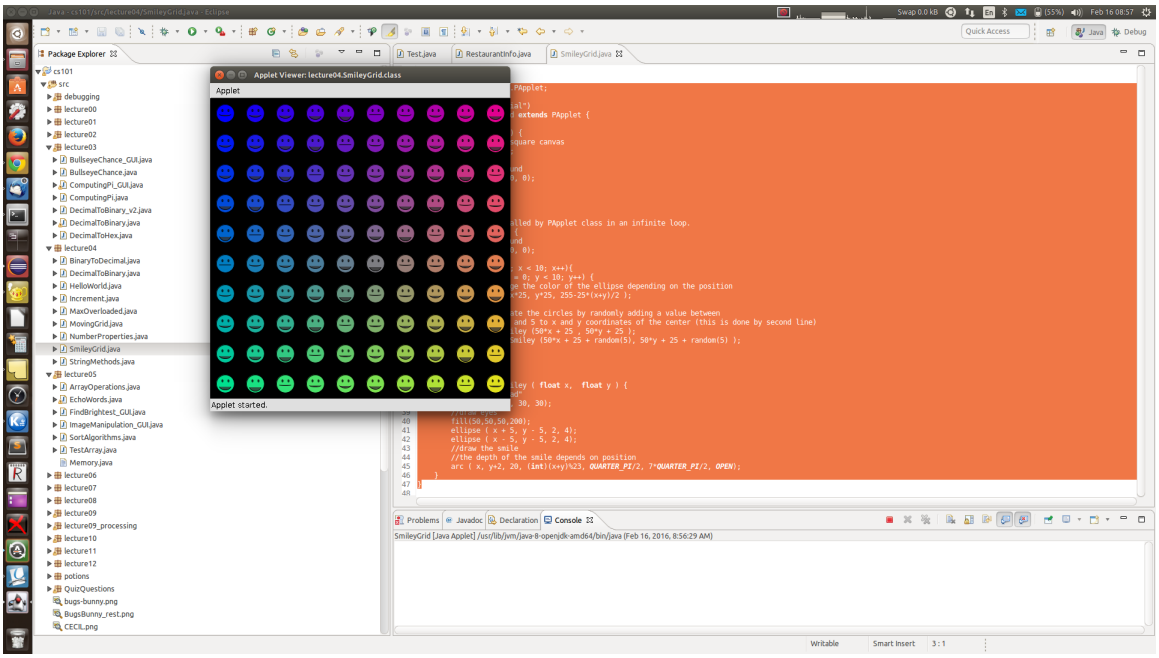
You may discuss any of the assignments with your classmates and tutors (or anyone else) but **all work for all assignments must be entirely your own**. Any sharing or copying of assignments will be considered cheating.

You should not use any features of Java that have not been covered in class. If you have doubt if you are allowed to use certain structures, just ask your instructor.

Problem 1 (20 points): Setup Your Eclipse to Run Processing

Follow the instructions posted on the website at `http://cs.nyu.edu/~joannakl/cs101.06_f15/notes/processingInEclipse.pdf` to setup your Eclipse to run Processing.

Write and run the test program at the end of those instruction and take a screen-shot once the window of the program is running - submit that image for this problem. On my machine it looks something like this:



Feel free to play around with this program: you can change colors, sizes, etc. Just don't break it!

Call your image file: **`Smiles.jpg`** or **`Smiles.png`** (You can use any image format).
You are only submitting the image for this problem, not the source code!

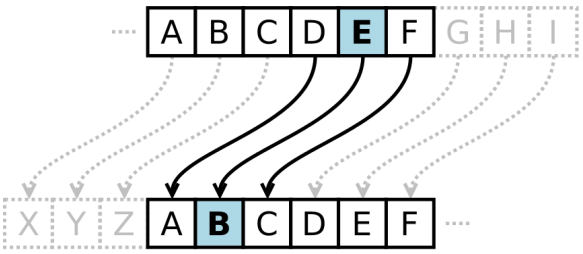
Joanna Klukowska
joannakl@cs.nyu.edu

Problem 2 (80 points): Caesar Cipher

You are going to write a simple encryption/decryption program. The algorithm used is very easy to break, so do not start using your program to "securely store your bank account!

Encryption is a process of encoding a message or information in such a way that only authorized parties can read it. (At least this is the idea until somebody discovers how to break an encryption pattern.)

Caesar cipher is one of the simplest encryption techniques. It was used by Julius Caesar to encrypt his private correspondence. It is sometimes called shift cipher because each letter in the unencrypted message is replaced by a different letter which is a fixed number of places down the alphabet (hence shifting).



"Caesar cipher left shift of three" by Matt_Crypto - http://en.wikipedia.org/wiki/File:Caesar3.png. Licensed under Public Domain via Wikimedia Commons. Accessed June 2015.

The figure above illustrates a left shift of three, so that each occurrence of E in the plaintext becomes B in the ciphertext.

When encrypting a message, a person needs to lookup each plain letter, find its corresponding cipher letter and perform a substitution. The following example uses a right shift of three:

```
Plain:  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
Cipher: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

The message "Hello class" is encrypted by replacing each character with a character three spaces to the right in the alphabet resulting in "Khoor fodvv". The message "Ohz Brun" is decrypted by replacing each character with a character three spaces to the left in the alphabet resulting in "New York". It is easy to perform both tasks once we know the shift value and the alphabet.

Your Assignment

Write a program that performs encryption/decryption based on the Caesar cipher. Here is a detailed outline of the program:

- Display the application banner:

```
======================
 Encrypt/Decrypt Tool
======================
```

- Ask the user if they want to perform encryption or decryption.

```
What would you like to do? (E)ncrypt or (D)ecrypt?
```

  The program should accept both upper and lower case letters. If the user enters a different letter, the program should display an error message `ERROR: invalid choice.` and terminate immediately.

- Ask the user for the key value. The key value is the shift amount. It should be an integer in the range -26-26 (inclusive). A shift of zero indicates that no encryption is performed since each letter in the plaintext is replaced with itself in the ciphertext. A positive number means shift right during encryption and left during decryption. A negative number means shift left during encryption and right during decryption. Your program should verify that the key is in the valid range. If the key is not valid, your program should display an error message and terminate immediately.

- Ask the user for the message. You may assume that the message is contained in one line of text, but may contain spaces and punctuation characters.

- Compute the encrypted/decrypted message and print it to the screen. All alphabetic characters should be replaced by their encrypted/decrypted equivalents. Any other characters (including digits) should be left unmodified. Your program should preserve case of the letters: encrypting 'a' using the above cipher should result in 'd' and encrypting 'W' should result in 'z'.

Joanna Klukowska
joannakl@cs.nyu.edu

NOTE: You will be using different methods of the `Scanner` class to read in the user's input. For the first two inputs you can use `.next()` and `.nextInt()` methods. For the third input you need to you `.nextLine()`. This creates a problem in Java. Your textbook mentions it in section 4.4.5 (basically suggesting that you should never use `.nextLine()` after using any of the other `.next...()` methods). The workaround for the problem is to make a dummy call to the `.nextLine()` method right before you need to really call it to read in the string from the user. You can read more about the reasons for this in chapter 12.

Your program has to use methods: you should have an `encrypt` method and a `decrypt` method. Both methods should take the string to be encrypted/decrypted as a parameter along with an integer variable that holds the key value. Both methods should return a String that is an encrypted/decrypted version of the string in the parameter.

Sample Runs of the Program:

Assume user selects encryption, shift of 5 and the message "This is a nice day!"

```
======================
 Encrypt/Decrypt Tool
======================

What would you like to do? (E)ncrypt or (D)ecrypt?  e

Enter your encryption key: 5

Enter your message: This is a nice day!

Your encrypted message is: Ymnx nx f snhj ifd!
```

Assume user selects decryption, shift of 5 and the message "Ymnx nx f snhj ifd!"

```
====================
Encrypt/Decrypt Tool
====================

What would you like to do? (E)ncrypt or (D)ecrypt?: D

Enter your encryption key:  5

Enter your message: Ymnx nx f snhj ifd!

Your decrypted message is: This is a nice day!
```

Assume user selects decryption, shift of 7 and the message "Ymnx nx f snhj ifd!" (Note that this is not the same decryption key that was used for encruption, so the message is displayed, but does not match the original.)

```
====================
Encrypt/Decrypt Tool
====================

What would you like to do? (E)ncrypt or (D)ecrypt?: d

Enter your encryption key:  7

Enter your message: Ymnx nx f snhj ifd!

Your decrypted message is: Rfgq gq y lgac byw!
```

Assume user selects decryption, shift of 2 and the message "Orug ri wkh Ulqjv"

```
======================
 Encrypt/Decrypt Tool
```

Joanna Klukowska
joannakl@cs.nyu.edu

```
======================

What would you like to do? (E)ncrypt or (D)ecrypt?  D

Enter your encryption key: 3

Enter your message: Orug ri wkh Ulqjv

Your decrypted message is: Lord of the Rings
```

Assume user selects encryption, shift of -10 and the message "Lord of the Rings"

```
======================
 Encrypt/Decrypt Tool
======================

What would you like to do? (E)ncrypt or (D)ecrypt?  E

Enter your encryption key: -10

Enter your message: Lord of the Rings

Your encrypted message is: Beht ev jxu Hydwi
```

Assume user enters invalid choice

```
======================
 Encrypt/Decrypt Tool
======================

What would you like to do? (E)ncrypt or (D)ecrypt?  A

ERROR: invalid choice.
```

Joanna Klukowska
joannakl@cs.nyu.edu

Grading

Does the program compile? If not, you will loose all the points for that problem.

Is the program properly documented? (worth approximately 20% of each problem)

Proper documentation at this point in the course includes:

- preamble with the name of the author, date of creation and brief description of the program (the description should specify what the program does, not that it is a solution to problem 1 of homework 1);
- appropriately chosen variable names, i.e., descriptive names (a good name for the variable that stores the bonus amount in the last problem is `bonus`, not `x`);
- comments inside the code describing steps needed to be taken to accomplish the goal of the program;
- appropriate formatting, indentation and use of white space to make the code readable.

Remember that the code is read by humans and it should be easy to read for people who were not involved in its development.

Is the program well developed? (worth approximately 40% of each problem) Make sure you create variables of appropriate types, use control statements (conditionals and loops) that are appropriate for the task, accomplish your task in a well designed and simple way (not a convoluted algorithm that happens to produce the correct output for some unknown reason). You should also design a friendly and informative user interface.

Is the program correct? (worth approximately 40% of each problem) Make sure that your program produces valid results that follow the specification of the problem every time it is run. At this point you can assume a "well behaved user" who enters the type of data that you request. If the program is not completely correct, you get credit proportional to how well it is developed and how close you got it to the completely correct code.

What and how to submit?

You should submit three source code files combined into a single **zip** file to NYU Classes. Do not submit all the files that Eclipse creates, just the source code files that have `.java` extensions. Name your classes as specified in the problems.

If you wish to use your (one and only) freebie for this project (one week extension, no questions asked), then complete the form at `http://goo.gl/forms/fpUJrF64b5` **before the due date for the assignment**. All freebies are due seven days after the original due date and should be submitted to NYU Classes.

Questions

Post any questions you have regarding this assignment to Piazza under the "homeworks" topic.