Programming Exercise: Implementing the Caesar Cipher

Assignment 1: Word Play

You will write a program to transform words from a file into another form, such as replacing vowels with an asterix.

Specifically, you should do the following.
- Create a new class called WordPlay.
- Write a method **isVowel** that has one parameter named **ch**. This method returns true if **ch** is a vowel (one of a, e, i, o, or u or the uppercase versions) and false otherwise. You should write a tester method to see if this method works correctly. For example, `isVowel('F')` should return false, and `isVowel('a')` should return true.
- Write a method **replaceVowels** that has two parameters, a String named **phrase** and a character named **ch**. This method should return a String that is the string phrase with all the vowels (uppercase or lowercase) replaced by **ch**. For example, the call `replaceVowels("Hello World", '*')` returns the string "H*ll* W*rld". Be sure to call the method **isVowel** that you wrote and also test this method.
- Write a method **emphasize** with two parameters, a String named **phrase** and a character named **ch**. This method should return a String that is the string phrase but with the character **ch** (upper- or lowercase) replaced by
    - '*' if it is in an odd number location in the string (e.g. the first character has an odd number location but an even index, it is at index 0)
    - '+' if it is in an even number location in the string (e.g. the second character has an even number location but an odd index, it is at index 1)

    For example, the call `emphasize("dna ctgaaactga", 'a')` would return the string "dn* ctg+*+ctg+", and the call `emphasize("Mary Bella Abracadabra", 'a')` would return the string "M+ry Bell+ +br*c*d*br+". Be sure to test this method.

Assignment 2: Caesar Cipher

You will start with the Caesar Cipher algorithm you learned about in the videos, and you will make some enhancements to it, so that it works with all letters (both uppercase and lowercase) and to make it a little bit harder to decrypt. Write these methods in a CaesarCipher class you can use in the next lesson.

Specifically, you should do the following:

- Create a new class called **CaesarCipher**.
- Write the method **encrypt** that has two parameters, a String named **input** and an int named **key**. This method returns a string that has been encrypted using the Caesar Cipher Algorithm explained in the videos. Assume that all the alphabetic characters are uppercase letters. For example, the call

      encrypt("FIRST LEGION ATTACK EAST FLANK!", 23)

  should return the string

      "CFOPQ IBDFLK XQQXZH BXPQ CIXKH!"

- Write the void method **testCaesar** that has no parameters. This method should read a file and encrypt the complete file using the Caesar Cipher algorithm, printing the encrypted message. You may want to include the lines:

      FileResource fr = new FileResource();
      String message = fr.asString();
      String encrypted = encrypt(message, key);
      System.out.println("key is " + key + "\n" + encrypted);

- Modify the **encrypt** method to be able to handle both uppercase and lowercase letters. For example, encrypt("First Legion", 23) should return "Cfopq Ibdflk" and encrypt("First Legion", 17) should return "Wzijk Cvxzfe". Be sure to test the **encrypt** method.
- Write the method **encryptTwoKeys** that has three parameters, a String named **input**, and two integers named **key1** and **key2**. This method returns a String that has been encrypted using the following algorithm. Parameter **key1** is used to encrypt every other character with the Caesar Cipher algorithm, starting with the first character, and **key2** is used to encrypt every other character, starting with the second character. For example, the call encryptTwoKeys("First Legion", 23, 17) should return "Czojq Ivdzle". Note the 'F' is encrypted with key 23, the first 'i' with 17, the 'r' with 23, and the 's' with 17, etc. Be sure to test this method.