```java

/*
 * Mitch Feigenbaum
 * Period 5
 * November 30, 2021
 * On my honor, I pledge that I have neither given nor received unauthorized assistance on this assignment or test.
 */
import java.util.Scanner;

/**
 * This class contains methods to encrypt and decrypt strings as well as a main
 * method which enables a user to encrypt and decrypt strings through a cli.
 */
public class FeigenbaumU3 {
        /* Defines alphabet */
        private static String alphabetLower = "abcdefghijklmnopqrstuvwxyz";
        private static String alphabetUpper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

        /**
         * Creates a cli interface for encrypting or decrypting strings based on user
         * defined parameters. This method first opens scanners to collect user input.
         * The program then offers documentation of the program to the user with a
         * description briefly describing program input and output. The method then
         * proceeds to collect parameters from the user.
         * <p>
         * The program proceeds the ask the user to enter a message and shift. The shift
         * is immediately moded by 26 and will prompt the user to re-enter their shift
         * if it is 0 or a multiple of 26. Then the user will be prompted to decide
         * whether they wish to encrypt or decrypt the string. The program will then
         * enter a do-while loop in which the user may decide whether to encrypt or
         * decrypt their string and will be returned their tranformed string. If the
         * user did not enter a valid option then the loop will prompt the user again to
         * enter a valid option.
         *
         * @param args added for semantics
         * @see encrypt
         * @see decrypt
         */
        public static void main(String[] args) {
                Scanner scanChar = new Scanner(System.in);
                Scanner scanNum = new Scanner(System.in);
                System.out.println("Program input: A string, a shift, encrypt/decrypt option");
                System.out.println("Program output: Either a decrypted or encrypted string based on user input");
                System.out.print("Enter a string: ");
                String message = scanChar.nextLine();
                System.out.print("Enter a shift: ");
                int key;
                do {
                        key = scanNum.nextInt() % 26;
                        if (key == 0)
                                System.out.print("Please enter a number which is not a multiple of 26: ");
                } while (key == 0);
                String mode = "";
                do {
                        System.out.print("Type 'e' to encrypt the string or 'd' to decrypt the string: ");
                        mode = scanChar.nextLine().toLowerCase();
                        if (mode.equals("e"))
                                System.out.printf("Your encrypted string is: %s%n", encrypt(message, key));
                        else if (mode.equals("d"))
                                System.out.printf("Your decrypted string is: %s%n", decrypt(message, key));
                        else
                                System.out.print("Please enter your choice again: ");
                } while (!mode.equals("e") && !mode.equals("d"));
        }

        /**
         * Returns an encrypted string based on a shift and unencrypted message. The
         * method encrypts the string by looping through the string, finding the index
         * at which the current character of the string is at in the alphabet and then
         * adds a shift to this index to find the letter which corresponds to the proper
         * shift. This letter is then added to a string which is returned to the user
         * once the string has been iterated through.
         * <p>
         * This method also contains support for lowercase and uppercase letters, which
         * allows for the passage of case-sensitive strings. In the event that a
         * non-alphabetical character is iterated upon the character is simply added to
         * the encrypted string. This method can also be used to decrypt strings by
```

```java
	 * passing 26 minus the shift to the key parameter. This is what the decrypt
	 * method does behind the scenes.
	 *
	 * @param message an unencrypted string
	 * @param key     a shift by which to encrypt the string
	 * @return a string encrypted by a shift of key
	 * @see decrypt
	 */
	public static String encrypt(String message, int key) {
		String encryptedString = "";
		char currentChar;
		for (int i = 0; i < message.length(); i++) {
			currentChar = message.charAt(i);
			if (alphabetLower.indexOf(currentChar) != -1)
				encryptedString += alphabetLower
						.charAt((alphabetLower.indexOf(currentChar) + key) % 26);
			else if (alphabetUpper.indexOf(currentChar) != -1)
				encryptedString += alphabetUpper
						.charAt((alphabetUpper.indexOf(currentChar) + key) % 26);
			else
				encryptedString += currentChar;
		}
		return encryptedString;
	}

	/**
	 * Returns a decrypted string based on a shift and encrypted message. The method
	 * uses the encrypt method to decrypt the string by using a shift of 26 minus
	 * the shift used to encrypt the string.
	 *
	 * @param message an encrypted string
	 * @param key     a shift to decrypt the string with
	 * @return a string decrypted by a shift of key
	 * @see encrypt
	 */
	public static String decrypt(String message, int key) {
		return encrypt(message, 26 - key);
	}
}
```