

```

import java.util.Scanner;

public class PlayfairCipher {

    private char[][] keyMatrix;

    public PlayfairCipher (String key) {
        initializeTheKeyMatrix(key);
    }

    private void initializeTheKeyMatrix(String key) { //initializes key matrix
        keyMatrix = new char[5][5];
        String keyWithoutDuplicates = removeTheDuplicateLetters(key + "ABCDEFGHIJKLMNOPQRSTUVWXYZ");

        int col = 0;
        int row = 0;

        for (char c : keyWithoutDuplicates.toCharArray()) {
            if (Character.isLetter(c)) {
                keyMatrix[row][col] = Character.toUpperCase(c);
                col++;
                if (col == 5) {
                    col = 0;
                    row++;
                }
            }
        }
    }

    private String removeTheDuplicateLetters(String input) { //removes duplicate letters in matrix
        StringBuilder result = new StringBuilder();
        for (char c : input.toUpperCase().toCharArray()) {
            if (c != ' ' && result.indexOf(String.valueOf(c)) == -1) {
                result.append(c);
            }
        }
        return result.toString();
    }

    private String prepareTheMessage(String message) { //converts lowercase to uppercase
        return message.replaceAll("[^a-zA-Z]", "").toUpperCase();
    }

    public String encryptMessage(String plaintext) { //encrypts message
        StringBuilder ciphertext = new StringBuilder();
        plaintext = prepareTheMessage(plaintext);

        for (int i = 0; i < plaintext.length(); i += 2) {
            char first = plaintext.charAt(i);
            char second = (i + 1 < plaintext.length()) ? plaintext.charAt(i + 1) : 'X';

            int[] firstPosition = findThePosition(first);
            int[] secondPosition = findThePosition(second);

            int row1 = firstPosition[0];
            int col1 = firstPosition[1];
            int row2 = secondPosition[0];
            int col2 = secondPosition[1];

            if (row1 == row2) {
                col1 = (col1 + 1) % 5;
                col2 = (col2 + 1) % 5;
            } else if (col1 == col2) {
                row1 = (row1 + 1) % 5;
                row2 = (row2 + 1) % 5;
            } else {
                int temp = col1;
                col1 = col2;
                col2 = temp;
            }

            ciphertext.append(keyMatrix[row1][col1]);
            ciphertext.append(keyMatrix[row2][col2]);
        }
    }
}

```

```

    }

    return ciphertext.toString();
}

public String decryptMessage(String theciphertext) { //decrypts message
    StringBuilder plaintext = new StringBuilder();

    for (int i = 0; i < theciphertext.length(); i += 2) {
        char first = theciphertext.charAt(i);
        char second = theciphertext.charAt(i + 1);

        int[] firstPosition = findThePosition(first);
        int[] secondPosition = findThePosition(second);

        int row1 = firstPosition[0];
        int col1 = firstPosition[1];
        int row2 = secondPosition[0];
        int col2 = secondPosition[1];

        if (row1 == row2) {
            col1 = (col1 - 1 + 5) % 5;
            col2 = (col2 - 1 + 5) % 5;
        } else if (col1 == col2) {
            row1 = (row1 - 1 + 5) % 5;
            row2 = (row2 - 1 + 5) % 5;
        } else {
            int temp = col1;
            col1 = col2;
            col2 = temp;
        }

        plaintext.append(keyMatrix[row1][col1]);
        plaintext.append(keyMatrix[row2][col2]);
    }

    return plaintext.toString();
}

private int[] findThePosition(char letter) { //adjusts J to I, then searches for position of letter in keyMatrix
    if (letter == 'J') {
        letter = 'I';
    }
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++) {
            if (keyMatrix[i][j] == letter) {
                return new int[]{i, j};
            }
        }
    }
    return null;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    //gets input from user for the plaintext/keyword
    System.out.print("Enter plaintext: ");
    String originalPlaintext = scanner.nextLine();

    System.out.print("Enter keyword: ");
    String keyword = scanner.nextLine();

    //creates the PlayfairCipher object
    PlayfairCipher playfairCipher = new PlayfairCipher (keyword);

    //processes plaintext entered by user
    String processedPlaintext = playfairCipher.prepareTheMessage(originalPlaintext);

    //displays a menu for encryption/decryption
    System.out.println("Choose desired option:");
    System.out.println("1. Encrypt the text");
    System.out.println("2. Decrypt the text");
}

```

```
int choice = scanner.nextInt();
scanner.nextLine();

if (choice == 1) {
    //displays the encrypted message
    String ciphertext = playfairCipher.encryptMessage(processedPlaintext);
    System.out.println("Encrypted Message: " + ciphertext);
} else if (choice == 2) {
    //displays the decrypted message
    String decryptedText = playfairCipher.decryptMessage(processedPlaintext);
    System.out.println("Decrypted Message: " + decryptedText);
} else {
    System.out.println("Invalid choice");
}

//closes the scanner
scanner.close();
}
}
```