Programming Paradigms — Scala Coursework

To be completed using Scala.

Purposes of this assignment:

To get you started programming *functionally* in Scala.

The problem

You are required to implement the *Playfair Cipher* for enciphering and deciphering messages (see the Wikipedia article for an excellent explanation). We are going to use the following variant:

- The cipher table is filled in left to right, top to bottom.
- Any `'j'` in the input will be replaced by `'i'`.
- If a double letter occurs in a pair, an 'x' will be inserted between them.

  Example: `"te ll in g"` $\longrightarrow$ `"te lx li ng"`, but `"op po se"` does not get an `x` inserted.

  Exception: If a pair is `"xx"`, insert a `'q'` between them.
- If a final letter is needed to complete a pair, use `'z'`.

Details

Here is an example input to be encoded:

```
An anonymous reader sends word of a proof-of-concept Google Chrome browser
extension that steals users' login details. The developer, Andreas Grech,
says that he is trying to raise awareness about security among end users,
and therefore chose Chrome as a test-bed because of its reputation as the
safest browser.
```

With the keyword `"Pennsylvania"`, the resulting code block is:

p	e	n	s	y
l	v	a	i	b
c	d	f	g	h
k	m	o	q	r
t	u	w	x	z

and the example encoded is:

```
fafaw aermw yqnvm vqyns genwm hwoln kqwow ofkpf nexcq wqfvp
dckqu vhzwn ynmyz unsig wazcl wpxnv ipxey mpiqf asmvw lbvpx
dymvd vaken obefm yinhq pdgyb npxfb zcsvp xzbas cxqki bynfn
bonsn yniar wuynd tqbzp vowad sefxe ymnie fzcym ndqkp dfryn
dckqu vinlw nyzlv mvyfl xenmg axpmy etwlx lwain zcnyf onyzl
kqxny m
```

You should name your project `Playfair`, your package `playfair`, and your `"main"` object `Playfair`. The `main` method will interact with the user:

1. It will ask whether to encode, decode, or quit;
2. then (unless quitting) it will ask for the keyword,
3. then it will ask for the name of a file to be encoded/decoded.
4. It will read in the file,
5. encode or decode it, and
6. display the result.

This should be repeated until the user *quits* the program.

Encoding

- The text to be enciphered will be a single `String` of arbitrary text.
- The text will be mixed case, with spaces, newlines, and punctuation marks.
- The return value should be all lowercase; letters should be blocks of 5, with a single space between blocks (the last block may contain fewer than 5 characters).
- There will be ten blocks per line (the last line may have fewer blocks).
- There should be no whitespace at the beginning or the end, and only a single space or a single newline between blocks.
- All the punctuation should be discarded.

Decoding

- The text to be decoded will be in the format produced by the encoder.
- The result will have the same structure as the encoded text (blocks of five letters, whitespace as specified above.
- The number of blocks, and the number of characters per block, may not be the same as in the encoded file.

Your code

Create a class `Coder(keyword: String)` with an `encode(plainText: String): String` method and a `decode(secretText: String): String` method. Use additional classes and objects as you see fit.

`Testing and documentation`

- Every method, except `main`, should be tested with `Scalatest`.
- TDD (Test-Driven Development) is strongly recommended.
- Running the tests should not request input or produce output.
- Your program will also be tested with our unit tests.
- Every method should have `Scaladoc` comments.
- Use the `@author` tag in the comment for your `Playfair` object, so we know who you are.

Submission

You should submit your solution as part of your portfolio.

Grading considerations

In addition to the usual criters, that the program is working correctly, properly formatted, and properly commented, you should ensure that you use the following features of the Scala language:

- `def` — but don't overdo the number of functions; just be sensible.
- literal functions defined with `=>`.
- `map`, `filter`, `foldLeft` (or `/:`), `foldRight` (or `:\`), `yield`, and `Some`.
- `match`.
- `var`, `while`, `Array`, `null`.
- `for`, `List`, and maybe `Some`.

Don't just *throw* in some of the above, without seriously trying to use them, just to get fulfil the requirements of the coursework — you will lose marks!

Why are we doing this? To get you used to using some of the unique constructs in Scala. Remember that it is easy for us to write a program to recognise these keywords and their contextual usage.

Credits

Developed with, and based upon, an original assignment by David Matuszek, at the University of Pennsylvania — hence the encoding!