Algorithms & Data Structures (M)

Assessed Exercise 1 (2013–14)

This exercise is about using the Java collection classes (lists and/or sets).

Its weighting is *0.1* in the ADS(M) course assessment.

The deadline is *Friday 7 February 2014* at 16:30.

A simple spell-checker

In this exercise you will write a simple spell-checker program. Your program must read an input document and check it for misspelled words.

A spell-checker uses a collection of words known as its *dictionary*. It assumes that a word is misspelled if the word is not in its dictionary. Your program must read its dictionary from a file named `dict.txt`, which contains one word per line, in no particular order. (*Hint:* For the purposes of testing, you may use a small dictionary.)

Your program must read the input document from a file named `doc.txt`. The document contains words interspersed with spaces and punctuation. You should assume that a word is a sequence of consecutive letters (so, for example, "rat-trap" should be read as two words, "rat" and "trap"). You should also ignore letter case (so, for example, "This" should be treated like "this").

To facilitate testing, the program must output:

(1) the entire contents of `dict.txt`, followed by

(2) the entire contents of `doc.txt`, followed by

(3) all misspelled words found in `doc.txt` (with each word appearing *once* only).

For example, if given the following files:

`dict.txt`

```
a
an
and
by
in
is
not
of
or
that
the
them
to
```

`doc.txt`

```
To be, or not to be - that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune
Or to take arms against a sea of troubles
And by opposing end them.
```

your program should output the contents of `dict.txt`, followed by the contents of `doc.txt`, followed by the following misspelled words:

```
against arms arrows be end fortune mind nobler
opposing outrageous question sea slings suffer take
tis troubles whether
```

You are free to choose the order and format in which the misspelled words are output, but each misspelled word must be output once only. (The words need not be in alphabetical order.)

Using the Java collection classes

This problem can be solved simply, elegantly, and efficiently using the collection classes in the package `java.util`. These include `List` and `Set`, which are introduced in §1 of the course notes. You can find full documentation of the collection classes at docs.oracle.com/javase/7/docs/api.

Submission

By the deadline stated above, you must submit your deliverables through the ADS(M) Moodle page. (Click "Assessed exercise 1 submission".)

The deliverables are all your Java source files, class files, input data files, and output(s). The output(s) are essential to show that you have tested your program thoroughly. Submit all the deliverables as a single `.zip` file.

Important: You must also sign the School's "declaration of originality" form.

Assessment

Your program will be marked against the following criteria:

Effective use of Java collection classes	12 marks
Correctness	12 marks
Efficiency	6 marks

Your total mark will be converted to a grade on the University's 22-point scale. Your mark will be reduced if your code is clumsy or hard to read, or if you do not follow the above submission instructions.