

Java Programming (JP2)

Laboratory Sheet 9

This is the last JP2 lab session. This Lab Sheet contains the submission information for Laboratory 9 (week 10, 25 – 29 November 2013). **Next week you will use your lab timeslot to do your lab exam.**

You are expected to begin work on laboratory sheets before your scheduled session. This will be necessary if you are to make best use of the presence of your tutor during the lab, and to make a satisfactory attempt at the submission exercise(s).

The deadline for submission of the lab exercise is 24 hours after the end of your scheduled laboratory session in week 10 (25 –29 November 2013).

Of course you may submit work that is incorrect or incomplete. In order to stretch the stronger members of the class, some of the laboratory exercises are quite challenging, and you should not be too discouraged if you cannot complete all of them.

Aims and objectives

- to reinforce experience of file I/O in Java
- to practise String and character manipulation
- to reinforce object-oriented concepts

Set up

When you download Laboratory9.zip from moodle, please unzip this file. You will obtain a folder Laboratory9, containing a subfolder entitled Submission9_1. Remember that for this Laboratory you will have to switch your Eclipse workspace to the Laboratory9 folder.

In the folder Submission9_1 will be the following files:

- FindPalindromes.java which defines an empty `main` method
- Palindrome.java which is an outline class for you to complete
- NotPalindromicException.java which is an outline class for you to complete
- TestPalindrome.java which provides JUnit tests for you to check your Palindrome implementation
- words.txt which is a plain text file containing a list of words

In Eclipse, you should create a new project entitled Submission9_1; the given files will automatically become part of this project.

Submission material

Preparatory work for this programming exercise, prior to your scheduled lab session, is expected and essential to enable you to submit a satisfactory attempt.

Submission exercise 9

First you need to complete the `Palindrome` class to supply:

- a static method `isPalindromic()` that checks whether a supplied `String` parameter is a palindrome
- a public constructor that takes a single `String` parameter, this constructor should either create a new `Palindrome` object if the parameter is a palindrome, or it should throw a `NotPalindromicException` otherwise

- a `toString()` method that overrides the default inherited `toString()` method, to produce the `String` representation of the palindrome word
- You can check your `Palindrome` implementation using the provided JUnit test cases in the `TestPalindrome` class

You need to complete the `NotPalindromicException` class, so that it has an appropriate place in the inheritance hierarchy and contains useful information about the word that triggered the Exception.

Finally you need to complete the `main` method of the `FindPalindromes` class:

- `args[0]` of `main` should be a filename for input
- the code should read lines from this file (assume one word per line) and check each line to see whether it is a `Palindrome`
- all detected `Palindrome` objects should be stored in an `ArrayList`
- After the file has been read completely, the `ArrayList` of detected palindromes should be printed out to the console. My output looks like this:

```
[A, a, aa, aba, acca, adda, affa, aga, aha, ajaja, aka, ala,
alala, alula, ama, amma, ana, anana, anna, apa, ara, arara, atta,
ava, awa, B, b, bib, bob, boob, bub, C, c, civic, D, d, dad, deed,
deedeed, degged, did, dod, dud, E, e, eke, elle, eme, ere, eve,
ewe, eye, F, f, G, g, gag, gig, gog, H, h, hah, hallah, huh, I, i,
ihi, imi, immi, J, j, K, k, kakkak, kayak, keek, kelek, L, l,
lemel, level, M, m, maam, madam, mem, mesem, mim, minim, mum,
murdum, N, n, nan, non, noon, nun, O, o, oho, otto, P, p, pap,
peep, pep, pip, poop, pop, pup, Q, q, R, r, radar, redder, refer,
repaper, retter, rever, reviver, rotator, rotor, S, s, siris, sis,
sooloos, T, t, tat, tebbet, teet, tenet, terret, tit, toot, tot,
tst, tut, tyt, U, u, ulu, ululu, umu, utu, V, v, W, w, waw, wow,
X, x, Y, y, yaray, yoy, Z, z]
```

Hints

What are the tutors looking for?

- `try-with-resources` statement, or `try/finally` to close input streams.
- `catch` blocks for specific `Exception` subclasses.
- sensible use of Java standard library methods.
- elegant code with appropriate control-flow constructs.
- *minimal redundancy* in code and data structures – e.g. the `Palindrome` object only needs to store half the `String` that encodes the palindrome – since the second half is the same as the first half reversed.

Submission

You should submit your work before the deadline no matter whether the programs are fully working or not.

When you are ready to submit, go to the JP2 moodle site. Click on Laboratory 9 Submission. Click 'Add Submission'. Open Windows Explorer and browse to the folder that contains your Java source code `...\Laboratory9\Submission9_1\` and drag *only* the **three** Java files `Palindrome.java`, `NotPalindromicException.java`, and `FindPalindromes.java` into the drag-n-drop area on the moodle submission page. **Your markers only want to read your java files, not your class files.** Then click the blue save changes button. Check the .java files are uploaded to the system. Then click submit assignment and fill in the non-plagiarism declaration. Your tutor will inspect your file and return feedback to you via moodle.

Gaining the credits for JP2

Recall that the credit criteria for JP2 include obtaining at least 7 ticks for lab assignments. To obtain a tick you must attend the lab and submit the assignment.