

Laboratory Sheet 5

This Lab Sheet contains material based on Lectures 1 – 10 (up to 22 October 2014), and contains the submission information for Laboratory 5 (week 6, 27 – 31 October 2014).

The deadline for submission is 48 hours after the end of your scheduled laboratory session in week 6 (27 – 31 October 2014).

Aims and objectives

- Reinforcement of object-oriented concepts in Java
- Use of exception handling
- Use of Java Collection library classes and methods

Set up

As usual, download Laboratory5.zip from moodle and unzip this file. You will obtain a folder Laboratory5, containing a subfolder entitled Submission5_1. Remember that for this Laboratory you will have to switch your Eclipse workspace to the Laboratory5 folder.

In subfolder Submission5_1 will be five Java source code files:

- Country.java – for you to complete
- NewsFinder.java – for you to complete
- CheckNews.java – contains main method, no need to modify
- CountryTest.java – JUnit tests for Country class, no need to modify
- NewsFinderTest.java – JUnit tests for NewsFinder class, no need to modify

This exercise involves writing code to search online news pages for mentions of specific countries.

Submission

You are to design and implement a class `Country` representing an international country. Each `Country` instance should have a single `private` field:

- `name` (a `String`)

There should be a single `public` constructor method:

- `Country(String name)`

You should override two inherited `public` instance methods from `Object`:

- `public String toString()` – returns the `Country`'s name as a `String`
- `public boolean equals(Object other)` – returns `true` if this object is compared against another object with the same `name`, `false` otherwise.

Please also define a static method for the `Country` class:

- `public static ArrayList<Country> getCountries()`

This method returns an `ArrayList` of `Country` objects. The list of `Country` objects should be obtained as follows:

1. Create a new empty `ArrayList` of countries
2. Use the Java library method `Locale.getAvailableLocales()`;

3. For each locale object, find its country name `String` via the instance method `Locale.getDisplayCountry(Locale.UK)`
4. Check that this country string is a non-empty string
5. Check that this country string has not been seen before (some are repeated since a single country may have multiple Locales)
6. If both tests succeed, add this country to the `ArrayList` of countries
7. At the end, return the countries list.

Use the JUnit test driver class, `CountryTest`, for this class to check that your code works. (Remember how to add the JUnit library to your project build path, as in previous labs.) The test driver creates a small number of `Countries` and carries out some operations on these objects.

Nest, you should design and implement a `NewsFinder` class. This class will have two `private String` instance fields: `url` and `content`. The `url` field stores the URL from which the news will be fetched. The `content` field stores a cached version of the data fetched from the URL.

Define a single public constructor for `NewsFinder` that takes a single `String` parameter specifying the URL to fetch. In the constructor, you should initialize both the `url` and the `content` fields of the object. The `content` field should be initialized using code similar to:

```
BufferedReader r = new BufferedReader(new InputStreamReader(
    new URL(url).openStream()));
```

and check the `BufferedReader.readLine()` documentation to see how to read lines of input as `Strings`. Consider how to concatenate these `Strings` efficiently into a single `String` object. Remember to use `try/catch` blocks as appropriate to handle exceptions.

Define an instance method `isInNews(Object o)` that belongs to the `NewsFinder` class. This class should check whether the `String` representation of `Object o` is mentioned in the `content String` of cached news. The method returns `true` if `o` is mentioned in the news, `false` otherwise. You could use the `String.indexOf()` method to determine whether `o` appears in `content`.

Use the JUnit test driver class, `NewsFinderTest`, for this class to check that your code works. For these tests to succeed, you will need an internet connection to fetch data from online RSS feeds.

Finally, you should be able to run the program by executing the `main` method in the `CheckNews` class. You should not need to modify this class at all. This main method iterates over countries and prints out whether they appears in the news.

How to submit

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 JOOSE2 moodle site. Click on Laboratory 5 Submission. Click 'Add Submission'. Open Windows Explorer and browse to the folder that contains your Java source code ...\\Laboratory5\\Submission5_1\\ and drag *only* the two Java files `Country.java` and `NewsFinder.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 two `.java` files are uploaded to the system. Then click

submit assignment and fill in the non-plagiarism declaration. Your tutor will inspect your files and return feedback to you via moodle.

Guidelines for Mark Scheme

A: All methods completed, with correct types and behaviour. All JUnit tests pass. Main method executes successfully.

B: Many methods completed, with correct types and behaviour. Some JUnit tests pass.

C: Some methods attempted, possible problems with types and behaviour. Code may not compile.

D: Minimal effort.

Marks will be awarded for clean, elegant code, with appropriate use of control flow and exception handling constructs. Sensible use of library methods is encouraged and will receive credit.