TA Guide for SED Ex4 - Creation and Dependency

To prepare, watch the lecture video linked below and download the exercise spec from CATE:

https://imperial.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=bfe8c56b-bcf5-4ab6-b31d-af3e00a5b016

During the lab (Tuesday 4-6pm in Level 2 computer labs) you should:

Walk around the room, checking how people are doing. Look over their shoulder. Don't be afraid to talk to them. Do not just stand at the side and wait for people to put their hand up.

If students are unsure where to start, show them BookSearchQueryTest. There are a lot of "null" values passed to the BSQ constructor, which is ugly. They should update the existing tests to create queries using a builder rather than the BookSearchQuery constructor, and hence remove the nulls.

They might also use a factory method to create the builder. That's fine, but not required.

Think about the naming of all the methods. Make sure it reads nicely when you look at the test. For example, it can be nice to use .withAuthorSurname(…) rather than .withName2(…) - creating the builder gives us an opportunity to improve the API, even if we can't change the field names in BSQ.

Then they should change BritishLibraryCatalogue to be a singleton, and extract an interface from it.
A common point of confusion is whether the getInstance() method can / should go in the interface.

Once the have their interface they can inject the dependency either through the constructor of BSQ (allowing the same query to be executed repeatedly on the same catalogue) or as a param to execute() (allowing the same catalogue to be queried more than once, e.g. once per day). That's up to them.

The interface allows them to test the logic in BSQ in isolation by using a mock (or fake) of the Catalogue. They do not need to write tests for the Catalogue - we assume the BLC developer will do that.

Example solution for the exercise:

https://imperial.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=9daba293-d5fc-4c2c-9685-ad5f00c9d4c6

Marking scheme:

+ 10 if they get 3/3 on LabTS.

+ 2 Builder Pattern : .withMethods() with fluent interface, .build() method returning BSQ)
+ 1 for all good names (e.g. renaming name1, name2, a nice way to go is .publishedBefore(…) etc)

+ 2 Correct Singleton Pattern : private static field, private constructor, public static getInstance() - iff the go for lazy instantiation of the field then the getInstance() method must be synchronized.

+ 2 Correct dependency inversion : extract Catalogue interface, inject as described above

+ 2 for testing BSQ in isolation using a mock (or fake) catalogue
+ 1 for testing (once) that the value returned from Calatogue.searchFor() is propagated to be returned by BSQ.execute()