In-class exercises (lecture 4)

At the start of every lecture:

1. Pull the latest code from the lecture-code repo
2. Open the Evening_lectures folder
3. Copy this week's folder somewhere else so you can edit the code without causing GitHub conflicts
4. Open the code:
    1. Find the build.gradle file in the folder called LectureX
    2. Double click build.gradle to open the project in IntelliJ.

Exercise 1: ADT specification example

Open up **Java 11's Color docs (https://docs.oracle.com/en/java/javase/11/docs/api/java.desktop/java/awt/Color.html)** (click the link or Google "Java 11 color").

Identify the ADT specification components:

- *Overview & abstract state* - A description of what the ADT is for. Description of key fields without implementation details.
- *Creators* - Methods that create a new object i.e. constructors.
- *Observers* - Methods that return values or information about an object without changing any values.
- *Producers* - Methods that create and return a new object. Mostly for *immutable* ADTs.
- *Mutators* - Methods that change the value of a field or otherwise modify an ADT object. *Mutable ADTs only!*

Exercise 2: Implement `insert` in ListOfStrings

Complete the `insert` method in ListOfStrings. Make sure your implementation meets the specification described in the method's Javadoc. You can use the pre-written tests to determine if your implementation is correct.

Don't worry if you don't finish in the allotted time. The main objective is to understand conceptually how this operation is implemented. A sample solution will be provided.

Exercise 3: Adding new behavior to the list implementations

Add the following methods to the IListOfStrings interface:

```
/**
```

```
* Returns a sub-list of items that contain the given substring.
* @param substring The substring to filter by.
* @return A list of strings.
*/
IListOfStrings filter(String substring);

/**
* Returns a copy of the list with items in reverse.
* @return The list in reverse;
*/
IListOfStrings reverse();
```

Implement these methods first in LinkedListOfStrings, then, if you have time, in ListOfStrings. It's a good idea to be familiar with both array implementations and sequential linked list implementations but, if you're feeling pressed for time, focus on the linked list implementation.