159.272 Programming Paradigms

Tutorial 4: Test Cases as Specification

Objectives

1. implement an interface
2. use test cases as specification

Deadline

- see stream
- no late submissions allowed

Instructions

Your task is to implement a simple data structure, a list of strings, according to a specification that consists of an interface and a set of tests.

1. download template.zip from stream
2. this file contains an Eclipse project, import this as follows into Eclipse:
   File > Import > General > Existing Projects Into Workspace > Select Archive File
3. after importing, rename the project:
   select project > Refactor > Rename , the new name should be tutorial4-<your student id>
4. then rename the package nz.ac.massey.cs.pp.tutorial4.idyourid, replace "yourid" by your student id:
   select package > Refactor > Rename
5. this package contains a class `MySimpleList` with four methods. You task is to implement these for methods until all the tests in `MySimpleListTests` succeed.
6. You must not:
   1. change anything in `MySimpleListTests` or any class or interface in the nz.ac.massey.cs.pp.tutorial4 package
   2. modify the definition of the sole instance variable in `MySimpleList`, or add other instance variables
7. You can run `MySimpleListTests` as follows:
8. select `MySimpleListTests` > Run As > JUnit Test
9. the actual tests are defined as methods in `SimpleListTests`, the superclass of `MySimpleListTests`
10. Initially, most tests will fail. Your task is to program the methods in `MySimpleList` **until all tests succeed**.

Deliverables

1. export your Eclipse project as follows to a zip file:
   1. in Eclipse, select Project, then select File > Export > General > Archive File
   2. once the project has been exported, check the zip file created (use WinZip, 7Zip or similar) to make sure that the sources code files, .project and .classpath are all included. .project and .classpath might not be visible, you may need to enable "show hidden files" in your zip tool or OS file explorer.
2. upload this zip file to stream

Grading

1. this tutorial is worth 1%
2. the maximum number of points you can get is 2 (this will be scaled to allow for half points)
3. internal students have to submit **and** to physically attend the tutorial to get points – make sure your name is signed off by the lecturer or tutor before you leave the lab

Hints

1. when comparing objects, think careful whether to use `equals` or == (identity)
2. use the strategy used in `java.util.ArrayList` when the internal array is full (see lecture notes for details)