Inf2C tutorial SE3: Testing and Sequence Diagrams

Study this tutorial sheet and prepare your answers BEFORE the tutorial. Take a copy of this sheet with you to the tutorial.

1 Testing

1. What are the steps to create a simple test case using JUnit 4?
2. What is Test-Driven Development? Why is it useful?
3. Recall that the first two *Fibonacci numbers* are both 1, and that each subsequent Fibonacci number is the sum of the preceding two. Thus the first few Fibonacci numbers are:

   $$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144$$

   Write a JUnit 4 test case – that is, a test class with appropriate annotations and test methods – for a class Fibonacci. The specification of the class is that it should have a zero-argument constructor, a public method `void printFibonacci(int n)`, which should print the nth Fibonacci number to standard output, and a public[1] method `int calcFibonacci(int n)`, which should calculate the nth Fibonacci number and return it.

   You will need to consider:

   - which method(s) do we need to/can you test?
   - what input(s) shall you use?
   - do you need to comment your tests, or can you arrange that the test code is so clear that it can be regarded as self-documenting?

   You are only expected to take a "common sense" approach to choosing inputs at this stage, but this is something the Software Testing course considers in more depth.

4. Write the class Fibonacci, and run your tests. Fix any bugs!

[1] This is a change from the first version of this sheet

2 Sequence diagrams

1. For an easy warmup, draw a sequence diagram showing:

   - an actor t of class Tester and an object f of class Fibonacci
   - t invoking f's `printFibonacci` with argument 5
   - f returning the result.

   Include activation bars.

2. Extend your diagram to show that `printFibonacci` works by invoking `calcFibonacci`

3. In the light of your implementation, consider whether you could or should show more detail about the messages that pass in the scenario considered above. How do you decide how much detail to show? What choices make sense?

4. What do you think of the names used in this tutorial sheet?

3 Exam question from August 08/09

1. In UML, what is a sequence diagram? Give two circumstances under which a developer might consider it worthwhile to draw one. *(4 marks)*

2. Study the code fragments below. Draw a UML sequence diagram to show what happens when an actor of type `T` sends message `foo` with argument `true` to an object of class `A`. Show arguments to messages, activity bars and return arrows.

```
class A {
  // attributes myB of class B, myC of class C...
  // appropriate constructors...
  public void foo (bool b) {
    if (b) { myB.bar(this); }
    myC.mung(b)
  }
}

class B {
  public void bar (A a) {
    a.foo(false);
  }
}

class C {
  public void mung (bool b) {
    //do something which need not be modelled in detail
  }
}
```

(10 marks)

3. What is Extreme Programming? List the three of its practices which most directly relate to code, explaining each in a sentence. *(8 marks)*

4. Briefly explain why people practising Extreme Programming seldom use a UML tool to develop detailed sequence diagrams. *(3 marks)*