Unit Testing

zyBook Chap 5.11

Testing

- "The ***dynamic*** verification of the behavior of a program on a finite set of test cases, ***suitably selected*** from the usually infinite executions domain, against the ***expected behavior***"

[ISO/IEC TR 19759:2005. Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOK)]

- A process of **verifying the behavior** of our programs and **revealing software faults** (i.e., logic errors)

Unit Testing

- The most basic level of software testing
- Testing the functionality of **individual methods**
  - Independent paths within the source code
  - Logical decisions as both true and false
  - Loops at their boundaries
  - Internal data structures
  - …

Testing Strategies

- **Test Requirements**
- **Test Equivalence Classes**
- **Test Boundary Values**
- Test All Paths
- Test Exceptions

Testing Strategies – Test Requirements

- Testing the main functionality of the method

For example, to test the method

```
public static boolean isPalindrome (int userNum)
```

in PA05-A: Numeric Palindromes.

We need to verify if this method can return `true` when the parameter `userNum` is a palindrome, and return `false` otherwise.

Testing Strategies – Test Equivalence Classes

- Testing representative values from equivalence classes
  - We break up the possible inputs for each parameter into equivalence classes and test representative values for each parameter
  - Preferably the “middle” input values
- Ensure each equivalence class is tested

Testing Strategies – Test Equivalence Classes

As stated in the Program Description, the `userNum` should be within the range of 1 – 999, inclusive.

Hence, we divide the range of possible inputs into

- < 1
- 1 - 999
- > 999

Can the possible user input be further divided into more equivalent classes?

Testing Strategies – Test Equivalence Classes

As stated in the Program Description, the `userNum` should be within the range of 1 – 999, inclusive.

Hence, we divide the range of possible inputs into

- < 1
- 1 – 999
  - 1 – 9
  - 10 – 99
  - 100 – 999
- > 999

Testing Scenarios/Values:
Test our program with at least one value (in this case, a palindrome and a non-palindrome) from each equivalence class.

Testing Strategies – Test Boundary Values

- Once representative values of a method are tested, boundary values (if any) between the equivalence classes should be tested

- < 1
- 1 – 999
  - 1 – 9
  - 10 – 99
  - 100 – 999
- > 999

Boundary Values (in this case):
-1, **0**, 1, 9, 10, 99, 100, 999, 1000

How to improve the PA05-A test cases?

Current PA05-A test cases verify the program behavior given:

- -15
- 151
- 511
- 999
- 1000
- 456