

Unit Test Examples

Produced
by:

Eamonn de Leastar (edelestar@wit.ie)

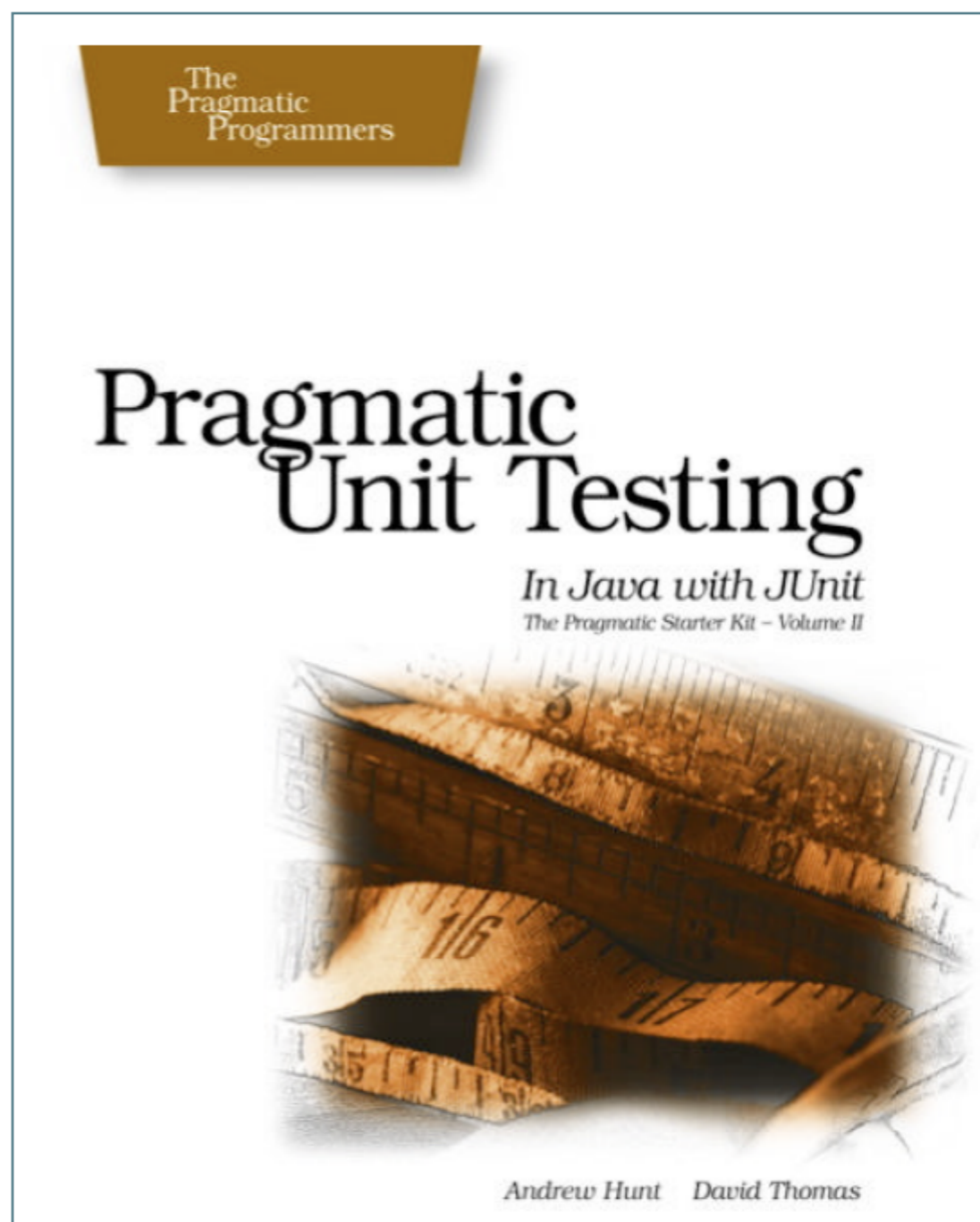


Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>

Some Unit Test Examples

- VideoPlayer



VideoPlayer Specification

- Implementations will provide methods to control a VCR or tape deck.
- There's the notion of a current position that lies somewhere between the beginning of tape (considered to be zero) and the end of tape - considered to be the tape duration in seconds.
- You can ask for the current position and move from there to another given position. Fast-forward moves from current position toward end by some amount. Rewind moves from current position toward the beginning by some amount. When tapes are first loaded, they are positioned at beginning automatically.
- A "Mark" can be established at any location - and this is remembered by the player. Going to a mark should make the location of the mark the current position.

```
public interface VideoPlayer
{
    public void fastForward (int seconds);

    public void rewind (int seconds);

    public int currentPosition ();

    public void markTimePosition (String name);

    public void gotoMark (String name);
}
```

VideoPlayer

- Initial implementation....

the System Under Test
(SUT)

```
public class VideoPlayerImpl implements VideoPlayer
{
    private int duration;
    private int currentPosition;
    private Map<String, Integer> marks;

    public VideoPlayerImpl(int length)
    {
        this.duration = length;
        marks = new HashMap<String, Integer>();
    }

    public int currentTimePosition ()
    {
        return currentPosition;
    }

    public void gotoMark (String name)
    { }

    public void markTimePosition (String name)
    { }

    public void rewind (int seconds)
    { }

    public void fastForward (int seconds)
    { }
}
```

Test Case Specifications

1. Verify that the initial position is 0.
2. Fast forward by some allowed amount (not past end of tape), then rewind by same amount. Should be at initial location.
3. Rewind by some allowed amount amount (not past beginning of tape), then fast forward by same amount. Should be at initial location.
4. Fast forward past end of tape, then rewind by same amount. Should be before the initial location by an appropriate amount to reflect the fact that you can't advance the location past the end of tape.
5. Try the same thing in the other direction (rewind past beginning of tape).
6. Mark various positions and return to them after moving the current position around.
7. Mark a position and return to it without moving in between

Test Fixture

```
public class VideoPlayerTest
{
    private VideoPlayer editor;

    @Before
    public void setUp() throws Exception
    {
        editor = new VideoPlayerImpl(100);
    }

    @After
    public void tearDown() throws Exception
    {
        editor = null;
    }
    ...
}
```

1.

(Test)

- Verify that the initial position is 0

```
@Test
public void testBOT()
{
    assertEquals(0, editor.currentTimePosition());
}
```

```
public class VideoPlayerImpl implements VideoPlayer
{
    private int duration;
    private int currentPosition;
    private Map<String, Integer> marks;

    public VideoPlayerImpl(int length)
    {
        this.duration = length;
        marks = new HashMap<String, Integer>();
    }

    public int currentTimePosition ()
    {
        return currentPosition;
    }
}
```

(SUT)₇

2.

- Fast forward by some allowed amount (not past end of tape), then rewind by same amount. Should be at initial location.

```
@Before
public void setUp() throws Exception
{
    editor = new VideoPlayerImpl(100);
}

@Test
public void testFF()
{
    editor.fastForward(30);
    editor.rewind(30);
    assertEquals(0, editor.currentTimePosition());
}
```

```
public void rewind (int seconds)
{
    currentPosition -= seconds;
}

public void fastForward (int seconds)
{
    currentPosition += seconds;
}
```


3.

- Rewind by some allowed amount (not past beginning of tape), then fast forward by same amount. Should be at initial location.

```
@Before
public void setUp() throws Exception
{
    editor = new VideoPlayerImpl(100);
}

@Test
public void testRewind()
{
    editor.fastForward(30);

    editor.rewind(20);
    editor.fastForward(20);
    assertEquals(30, editor.currentTimePosition());
}
```

```
public void rewind (int seconds)
{
    currentPosition -= seconds;
}

public void fastForward (int seconds)
{
    currentPosition += seconds;
}
```

4.

- Fast forward past end of tape, then rewind by same amount.
- Should be before the initial location by an appropriate amount to reflect the fact that you can't advance the location past the end of tape.

```
public void fastForward (int seconds)
{
    currentPosition += seconds;
}
```

```
@Before
public void setUp() throws Exception
{
    editor = new VideoPlayerImpl(100);
}

@Test
public void testFFBeyondEnd()
{
    editor.fastForward(50);

    editor.fastForward(60);
    editor.rewind(60);
    assertEquals(40, editor.currentTimePosition());
}
```

```
public void fastForward (int seconds)
{
    if ((currentPosition + seconds) < duration)
    {
        currentPosition += seconds;
    }
    else
    {
        currentPosition = 100;
    }
}
```

5

- Try the same thing in the other direction (rewind past beginning of tape).

```
@Before
public void setUp() throws Exception
{
    editor = new VideoPlayerImpl(100);
}

@Test
public void testRewindBeyondStart()
{
    editor.fastForward(50);

    editor.rewind(60);
    editor.fastForward(60);
    assertEquals(60, editor.currentTimePosition());
}
```

```
public void rewind (int seconds)
{
    if ((currentPosition - seconds) >= 0)
    {
        currentPosition -= seconds;
    }
    else
    {
        currentPosition = 0;
    }
}
```

```
public void rewind (int seconds)
{
    currentPosition -= seconds;
}
```

6,7

- Mark various positions and return to them after moving the current position around.
- Mark a position and return to it without moving in between.

```

@Test
public void testMark()
{
    editor.fastForward(30);
    editor.markTimePosition("one");
    editor.fastForward(30);
    editor.markTimePosition("two");

    editor.gotoMark("one");
    assertEquals(30, editor.currentTimePosition());
    editor.gotoMark("two");
    assertEquals(60, editor.currentTimePosition());
}

```

```

public void gotoMark (String name)
{
    if (marks.containsKey(name))
    {
        currentPosition = marks.get(name);
    }
}

public void markTimePosition (String name)
{
    marks.put(name, currentPosition);
}

```