

Junit

```
import static org.junit.Assert.*;
import org.junit.Test;
public class SimpleFutureTest {
    @Test
    public void testSetValue() {
        SimpleFuture<Integer> future = new SimpleFuture<Integer>();
        future.setValue(0);
        Integer result = null;
        try {
            result = future.get();
        } catch (Throwable e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        assertEquals((Integer) 0, result);
    }
    @Test(expected = Throwable.class)
    public void testSetException() throws Throwable {
        SimpleFuture<Integer> future = new SimpleFuture<Integer>();
        future.setException(new Throwable("Throwable exception"));
        future.get();
    }
    @Test(expected = IllegalMonitorStateException.class)
    public void testSetNullException() throws Throwable, IllegalMonitorStateException {
        SimpleFuture<Integer> future = new SimpleFuture<Integer>();
        future.setException(null);

        future.get();
    }
    @Test(expected = IllegalMonitorStateException.class)
    public void testSetIllegalMonitorStateException() throws Throwable,
        IllegalMonitorStateException {
        SimpleFuture<Integer> future = new SimpleFuture<Integer>();
        future.setValue(0);
        future.setException(new Throwable("Throwable exception"));
    }
    @Test
    public void testGetValue() throws Throwable {
        SimpleFuture<Integer> future = new SimpleFuture<Integer>();
        future.setValue(0);
        assertEquals((Integer) 0, future.get());
    }
}
```

```

@Test(expected = Throwable.class)
public void testGetException() throws Throwable {
    SimpleFuture<Integer> future = new SimpleFuture<Integer>();
    future.setException(new Throwable("Throwable exception"));
    future.get();
}

@Test(expected = InterruptedException.class)
public void testGetInterruptedException() throws InterruptedException, Throwable {
    Thread parentThread = Thread.currentThread();
    SimpleFuture<Integer> future = new SimpleFuture<Integer>();
    Runnable runnable = () -> {
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        parentThread.interrupt();
    };
    new Thread(runnable).start();
    future.get();
}

@Test
public void testIsDone() {
    fail("Not yet implemented");
}
}

```