

SMART TESTING

user guide

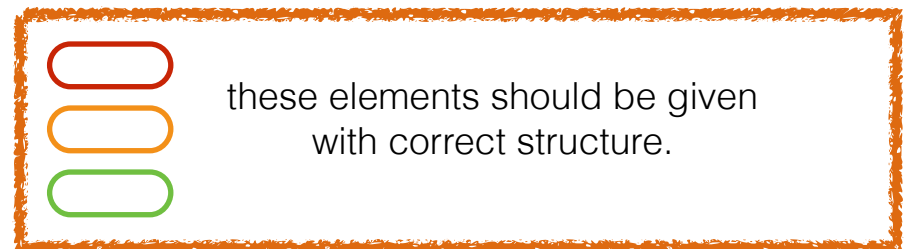
created by:Liu QiaoXi

Welcome to SMART TESTING!

step 0

First, you need a catalog file(.xml) with correct format to load in.

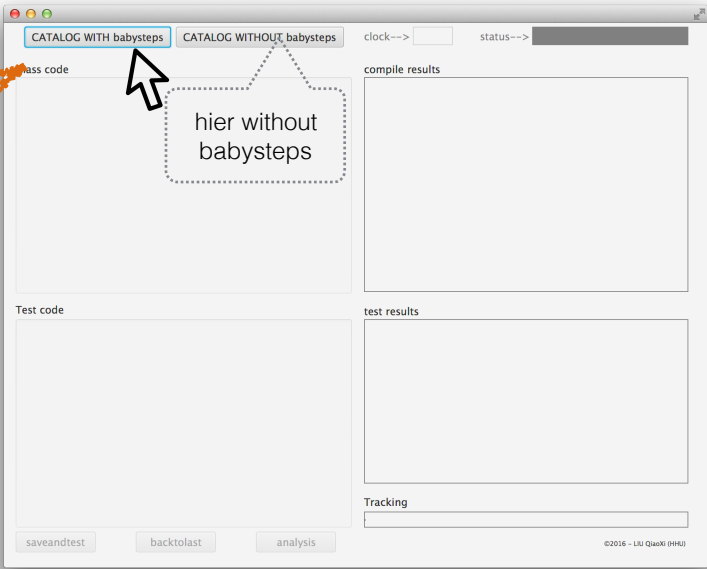
Here is an example:



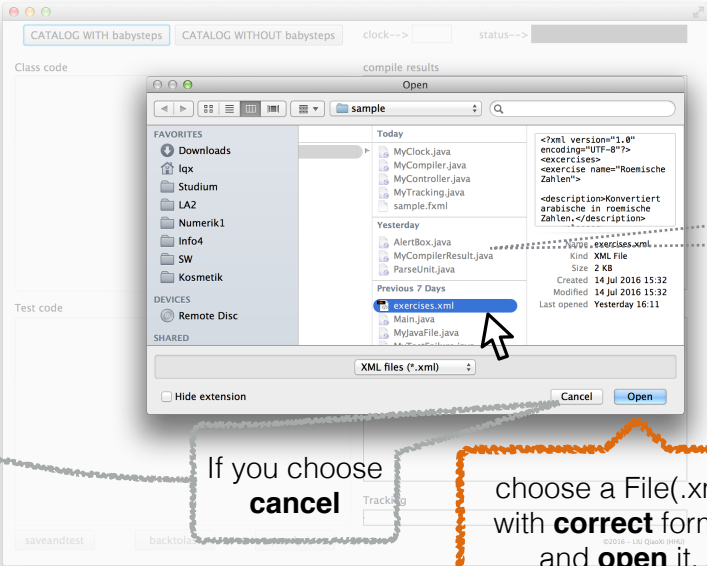
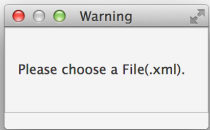
```
<?xml version="1.0" encoding="UTF-8"?>
<exercises>
  <exercise name="Roemische Zahlen">
    <description>Konvertiert arabische in roemische Zahlen.</description>
    <classes>
      <class name="RomanNumbers">
        public class RomanNumbers {
          public static String roman(int n) {
            return "I am groot.";
          }
        }
      </class>
    </classes>
    <tests>
      <test name="RomanNumbersTest">
        import static org.junit.Assert.*;
        import org.junit.*;
        public class RomanNumbersTest {
          @Test
          public void dieZahlEinsErgibtI() {
            RomanNumbers a = new RomanNumbers();
            String berechnet = a.roman(1);
            String erwartet = "I";
            assertEquals(erwartet, berechnet);
          }
        }
      </test>
    </tests>
    <config>
      <babysteps value="False" />
      <timetracking value="True" />
    </config>
  </exercise>
  <exercise name="Roemische Zahlen">
    <description>Konvertiert arabische in roemische Zahlen.</description> <classes>
      <class name="RomanNumberConverter">
        public class RomanNumberConverter {
        }
      </class>
    </classes>
    <tests>
      <test name="RomanNumberConverterTest">
        import static org.junit.Assert.*;
        import org.junit.Test;
        public class RomanNumberConverterTest {
          @Test
          public void testSomething() {
          }
        }
      </test>
    </tests>
    <config>
      <babysteps value="True" time="2:00"/>
      <timetracking value="True" />
    </config>
  </exercise>
</exercises>
```

Step 1

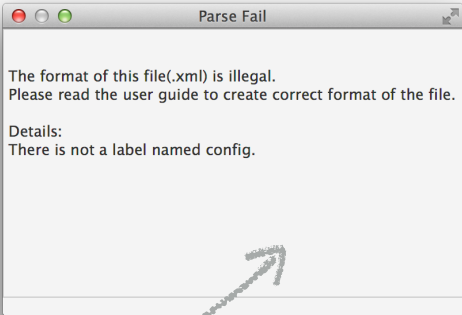
start your practice hier: with the babysteps mode



Step 2



Other extension can't be chosen.



If you choose **cancel**

choose a File(.xml) with **correct** format and **open** it.

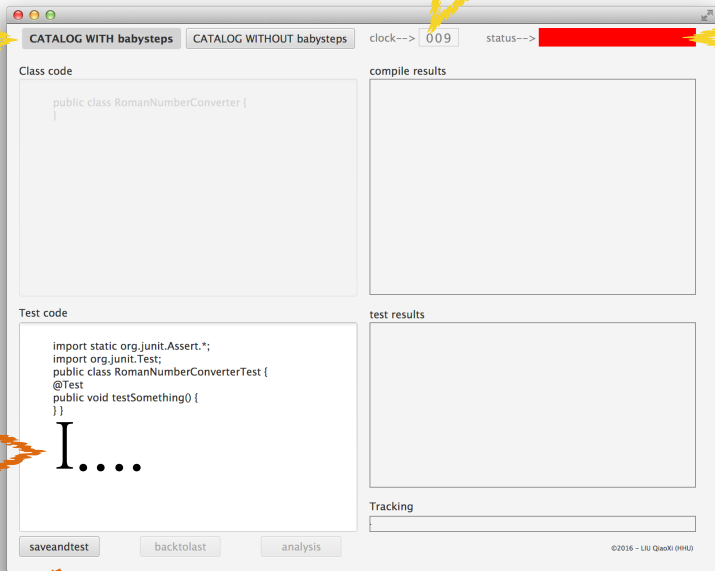
if the format is **wrong** (also shows the details of parse failure)

Step 3

The bold Font with grey background means, you are in babysteps mode

Now, you can modify your test code.

When you have finished, click this "saveandtest" button.



There is a clock when you are in the babysteps mode.

The status bar shows the momentan status.

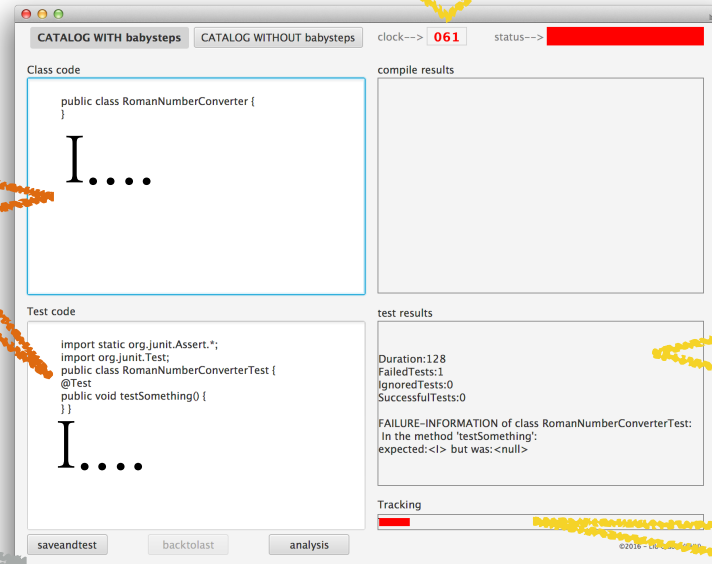
Step 4

When you are in baby steps mode:

- your clock will turn red when the half limit time reached. (The limit time is set up in the file(.xml)).
- when your clock reached limit time, the test/class code area will be cleaned and load the latest files again.

Your test code is successful compiled. So now you can edit your class code as well as test code.

Now you can not click the "back" button. Because it is grey(means disabled).



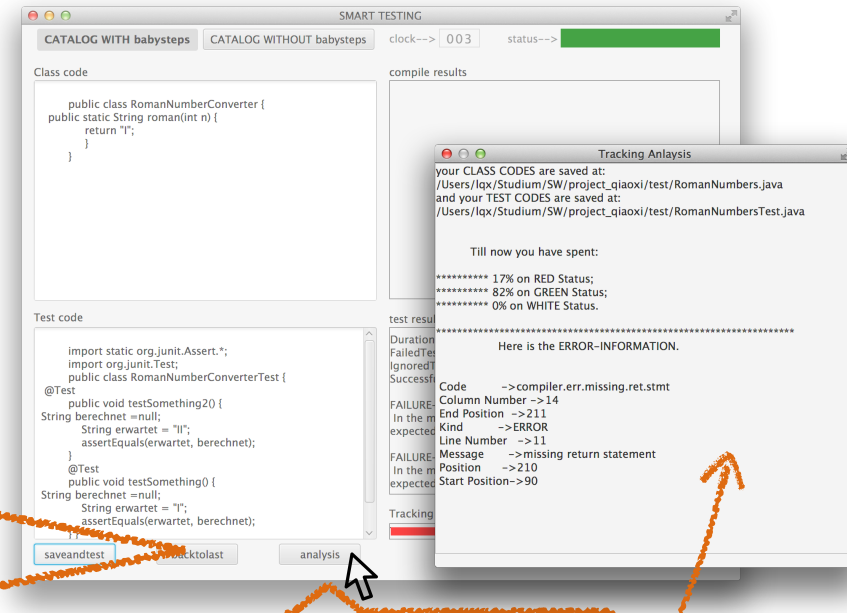
Here shows your test results.

Here is the Tracking bar. It notes your costed time in every status. Each time you click "save", it will change.

Step*

If the "back" button is active, you can click it, and then you will come back to last status, that you have already saved your files. The test code and class code area will be cleaned and then loaded your latest saved files.

Now, you can click the "analysis" button to check your tracking, your compile information(if you have) as well as your files' path.



Step**

During the whole TDDT, you can change your catalog at any time. when you click the catalog button, there is a confirmation dialog.

