|  |  |
| --- | --- |
| **Test Statistics** | **Test Analyses** |
| **Gherkin Test Log**  7 minutes 57 seconds ago  [REPORT](file:///C:\Users\Velislav\Documents\acc%20QA%20tester%20classes\testing2\lab8\results\gherkin\report.html)  **Test Statistics**   |  | **Total** | **Pass** | **Fail** | **Elapsed** | **Pass / Fail** | | --- | --- | --- | --- | --- | --- | | **Critical Tests** | 1 | 1 | 0 | 00:00:00 |  | | **All Tests** | 1 | 1 | 0 | 00:00:00 |  |  | **Statistics by Tag** | **Total** | **Pass** | **Fail** | **Elapsed** | **Pass / Fail** | | --- | --- | --- | --- | --- | --- | | No Tags |  |  |  |  |  |  | **Statistics by Suite** | **Total** | **Pass** | **Fail** | **Elapsed** | **Pass / Fail** | | --- | --- | --- | --- | --- | --- | | [**Gherkin**](file:///C:\Users\Velislav\Documents\acc%20QA%20tester%20classes\testing2\lab8\results\gherkin\log.html#s1) | 1 | 1 | 0 | 00:00:00 |  |   00:00:00.050 TEST SUITE: **Gherkin**   |  |  | | --- | --- | | **Full Name:** | Gherkin | | **Documentation:** | Example test case using the gherkin syntax.  This test has a workflow similar to the keyword-driven examples. The difference is that the keywords use higher abstraction level and their arguments are embedded into the keyword names.  This kind of *gherkin* syntax has been made popular by [Cucumber](http://cukes.info/). It works well especially when tests act as examples that need to be easily understood also by the business people. | | **Source:** | [/home/itse20458/u2045805/Calculator/gherkin.txt](file:///C:\Users\Velislav\Documents\acc%20QA%20tester%20classes\testing2\lab8\results\gherkin\gherkin.txt) | | **Start / End / Elapsed:** | 20170408 10:48:35.227 / 20170408 10:48:35.277 / 00:00:00.050 | | **Status:** | 1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed | | Describing    TEST CASE: **Addition**   |  |  | | --- | --- | | **Status:** | 1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Driven Test Log**  Generated 20170408 10:49:38 GMT -05:00 7 minutes 31 seconds ago  [REPORT](file:///C:\Users\Velislav\Documents\acc%20QA%20tester%20classes\testing2\lab8\results\data_driven\report.html)  **Test Statistics**   | **Total Statistics** | **Total** | **Pass** | **Fail** | **Elapsed** | **Pass / Fail** | | --- | --- | --- | --- | --- | --- | | **Critical Tests** | 6 | 5 | 1 | 00:00:00 |  | | **All Tests** | 6 | 5 | 1 | 00:00:00 |  |  | **Statistics by Tag** | **Total** | **Pass** | **Fail** | **Elapsed** | **Pass / Fail** | | --- | --- | --- | --- | --- | --- | | No Tags |  |  |  |  |  |  | **Statistics by Suite** | **Total** | **Pass** | **Fail** | **Elapsed** | **Pass / Fail** | | --- | --- | --- | --- | --- | --- | | [**Data Driven**](file:///C:\Users\Velislav\Documents\acc%20QA%20tester%20classes\testing2\lab8\results\data_driven\log.html#s1) | 6 | 5 | 1 | 00:00:00 |  |      |  |  | | --- | --- | | **Full Name:** | Data Driven | | **Documentation:** | Example test cases using the data-driven testing approach.  Tests use `Calculate` keyword created in this file, that in turn uses keywords in `CalculatorLibrary`. An exception is the last test that has a custom *template keyword*.  The data-driven style works well when you need to repeat the same workflow multiple times.  Notice that one of these tests fails on purpose to show how failures look like. | | **Source:** | [/home/itse20458/u2045805/Calculator/data\_driven.txt](file:///C:\Users\Velislav\Documents\acc%20QA%20tester%20classes\testing2\lab8\results\data_driven\data_driven.txt) | | **Start / End / Elapsed:** | 20170408 10:49:38.269 / 20170408 10:49:38.346 / 00:00:00.077 | | **Status:** | 6 critical test, 5 passed, **1 failed** 6 test total, 5 passed, **1 failed** |   00:00:00.006 TEST CASE: **Addition**   |  |  | | --- | --- | | **Full Name:** | Data Driven.Addition | | **Start / End / Elapsed:** | 20170408 10:49:38.311 / 20170408 10:49:38.317 / 00:00:00.006 | | **Status:** | PASS (critical) |   00:00:00.005 TEST CASE: **Subtraction**   |  |  | | --- | --- | | **Full Name:** | Data Driven.Subtraction | | **Start / End / Elapsed:** | 20170408 10:49:38.318 / 20170408 10:49:38.323 / 00:00:00.005 | | **Status:** | PASS (critical) |   00:00:00.005 TEST CASE: **Multiplication**   |  |  | | --- | --- | | **Full Name:** | Data Driven.Multiplication | | **Start / End / Elapsed:** | 20170408 10:49:38.323 / 20170408 10:49:38.328 / 00:00:00.005 | | **Status:** | PASS (critical) |   00:00:00.004 TEST CASE: **Division**   |  |  | | --- | --- | | **Full Name:** | Data Driven.Division | | **Start / End / Elapsed:** | 20170408 10:49:38.329 / 20170408 10:49:38.333 / 00:00:00.004 | | **Status:** | PASS (critical) |   00:00:00.003 **TEST CASE: Failing**   |  |  | | --- | --- | | **Full Name:** | Data Driven.Failing | | **Start / End / Elapsed:** | 20170408 10:49:38.334 / 20170408 10:49:38.337 / 00:00:00.003 | | **Status:** | **FAIL** (critical) | | **Message:** | 2 != 3 |   00:00:00.003 **KEYWORD: Calculate** 1 + 1, 3   |  |  | | --- | --- | | **Start / End / Elapsed:** | 20170408 10:49:38.334 / 20170408 10:49:38.337 / 00:00:00.003 |   00:00:00.001 KEYWORD: **CalculatorLibrary.Push Buttons** C${expression}=   |  |  | | --- | --- | | **Documentation:** | Pushes the specified `buttons`. | | **Start / End / Elapsed:** | 20170408 10:49:38.335 / 20170408 10:49:38.336 / 00:00:00.001 |   00:00:00.001 **KEYWORD: CalculatorLibrary.Result Should Be** ${expected}   |  |  | | --- | --- | | **Documentation:** | Verifies that the current result is `expected`. | | **Start / End / Elapsed:** | 20170408 10:49:38.336 / 20170408 10:49:38.337 / 00:00:00.001 |  |  |  |  |  | | --- | --- | --- | --- | | 10:49:38.336 | **FAIL** | 2 != 3 |  |   00:00:00.008 TEST CASE: **Calculation error** | Status: 6 critical test, 5 passed, 1 failed  6 test total, 5 passed, 1 failed    Describing  TEST CASE: Addition  TEST CASE: Multiplication  TEST CASE: Division  TEST CASE: Failing   |  |  | | --- | --- | | **Status:** | **FAIL** (critical) | | **Message:** | 2 != 3 |   00:00:00.003 **KEYWORD: Calculate** 1 + 1, 3  TEST CASE: Calculation error |

|  |  |
| --- | --- |
| **Test Statistics** | **Test Analyses** |
| **Test Statistics**   | **Total Statistics** | **Total** | **Pass** | **Fail** | **Elapsed** | **Pass / Fail** | | --- | --- | --- | --- | --- | --- | | **Critical Tests** | 5 | 5 | 0 | 00:00:00 |  | | **All Tests** | 5 | 5 | 0 | 00:00:00 |  |  | **Statistics by Tag** | **Total** | **Pass** | **Fail** | **Elapsed** | **Pass / Fail** | | --- | --- | --- | --- | --- | --- | | No Tags |  |  |  |  |  |  | **Statistics by Suite** | **Total** | **Pass** | **Fail** | **Elapsed** | **Pass / Fail** | | --- | --- | --- | --- | --- | --- | | [**Keyword Driven**](file:///C:\Users\Velislav\Documents\acc%20QA%20tester%20classes\testing2\lab8\results\keyword_driven\log.html#s1) | 5 | 5 | 0 | 00:00:00 |  |      |  |  | | --- | --- | | **Full Name:** | Keyword Driven | | **Documentation:** | Example test cases using the keyword-driven testing approach.  All tests contain a workflow constructed from keywords in `CalculatorLibrary`. Creating new tests or editing existing is easy even for people without programming skills.  This kind of style works well for normal test automation. If also business people need to understand tests, using *gherkin* style may work better. | | **Source:** | [/home/itse20458/u2045805/Calculator/keyword\_driven.txt](file:///C:\Users\Velislav\Documents\acc%20QA%20tester%20classes\testing2\lab8\results\keyword_driven\keyword_driven.txt) | | **Start / End / Elapsed:** | 20170408 10:50:13.016 / 20170408 10:50:13.073 / 00:00:00.057 | | **Status:** | 5 critical test, 5 passed, 0 failed 5 test total, 5 passed, 0 failed |   00:00:00.002 TEST CASE: **Push button**   |  |  | | --- | --- | | **Full Name:** | Keyword Driven.Push button | | **Start / End / Elapsed:** | 20170408 10:50:13.058 / 20170408 10:50:13.060 / 00:00:00.002 | | **Status:** | PASS (critical) |   00:00:00.000 KEYWORD: **CalculatorLibrary.Push Button** 1   |  |  | | --- | --- | | **Documentation:** | Pushes the specified `button`. | | **Start / End / Elapsed:** | 20170408 10:50:13.059 / 20170408 10:50:13.059 / 00:00:00.000 |   00:00:00.001 KEYWORD: **CalculatorLibrary.Result Should Be** 1   |  |  | | --- | --- | | **Documentation:** | Verifies that the current result is `expected`. | | **Start / End / Elapsed:** | 20170408 10:50:13.059 / 20170408 10:50:13.060 / 00:00:00.001 |   00:00:00.003 TEST CASE: **Push multiple buttons**   |  |  | | --- | --- | | **Full Name:** | Keyword Driven.Push multiple buttons | | **Start / End / Elapsed:** | 20170408 10:50:13.060 / 20170408 10:50:13.063 / 00:00:00.003 | | **Status:** | PASS (critical) |   00:00:00.003 TEST CASE: **Simple calculation**   |  |  | | --- | --- | | **Full Name:** | Keyword Driven.Simple calculation | | **Start / End / Elapsed:** | 20170408 10:50:13.063 / 20170408 10:50:13.066 / 00:00:00.003 | | **Status:** | PASS (critical) |   00:00:00.001 TEST CASE: **Longer calculation**   |  |  | | --- | --- | | **Full Name:** | Keyword Driven.Longer calculation | | **Start / End / Elapsed:** | 20170408 10:50:13.067 / 20170408 10:50:13.068 / 00:00:00.001 | | **Status:** | PASS (critical) |   00:00:00.002 TEST CASE: **Clear** | |  |  | | --- | --- | |  |  |   Describing  TEST CASE: Push button  TEST CASE: Push multiple buttons  TEST CASE: Simple calculation  TEST CASE: Longer calculation  TEST CASE: Clear  Status: 5 critical test, 5 passed, 0 failed  5 test total, 5 passed, 0 failed |