CS562: Advanced Topic in Software Engineering Due: Sep 16, 2014

HW2: Regression Test Harness

Huanglin Xiong (Lincoln) *hxiong@wpi.edu*

ABSTRACT

This regression test harness is written in Python 2.7. In the initialization status, the harness should contains:

```
harness-classify-test
harness-init
harness-report-all
harness-report-last
harness-report-test
harness-restart
harness-result-diff
harness-run-all
harness-run-test
harness-test-show
lib\Harness.py
lib\History.py
lib\__init__.py
```

all 13 files.

INSTALL

Extract the zip file into the place you desire to run the harness for. Open the terminal or any other command line console you like and "*cd*" to the harness directory. Run the following command to initialize the harness environment.

```
$ chmod +x harness-init
$ ./harness-init
```

After running the command, you should see some new folders created under harness directory.

```
history\current
results\failed\
results\passed\
results\unclassified\
temp\
test\failed\
test\inactive\
test\new\
test\passed\
test\unclassified\
```

The harness environment is built successfully if all above showing directories is created. Every time, new test files should be copied into "test/inactive" directory. The command will take care of it automatically afterwards.

HOW TO

There are totally 10 command currently available in this harness including the "*harness-init*". All of them are just command line commands allowing taking some optional arguments. Basically, if you find any difficulty understanding how to use the command, just add a "*-h*" or "*–help*" argument after the command you want to run. The build-in help docs for that command will show.

The correct place to put new test files into harness is "test/inactive", no matter in what status of harness. Each time you run command has so-called "run test" or "init/reinit" function, the harness will move test files from "test/inactive" to "test/new" for running.

- **harness-init**
  Initialize the harness. You could run this command at any time you want to swipe out all the history, results and refresh the whole harness environment. NOTE: After running this command, only the test files will remain in the harness!

- **harness-restart**
  Restart/reinitialize the harness. After running this command, the "current" file in history will become "history_*index*" file following with the running index(an increasing 4 digits number). All test files will put into "test/new" directory.

- **harness-run-all**
  Run all the tests in "test/new" directory and put the runned test files into "test/unclassified" also the test running results(the standard output) into "results/unclassified". In fact, in the implementation of this command, all files under "test/inactive" will move to "test/new" at first. Then doing the running. The optional argument "*-V*" or "*–verbose*" will show a verbose output of the result that test ran.

- **harness-run-test**
  Run specific test(s). This command can take an optional argument "*-R*" to allow regular expression of running tests. Or you could just run test(s) by typing their full file name. The optional argument "*-V*" or "*–verbose*" will show a verbose output of the result that test ran.
  Example:

  ```
  $ ./harness-run-test -R test_unit_*
  ```

  This will run all test files name matching the regular expression "test_unit_*".

  ```
  $ ./harness-run-test -V test_unit_isTriangle.test test_unit_helloworld.test
  ```

  This will run the two test files named with "test_unit_isTriangle.test" and "test_unit_helloworld.test" with the verbose output, printing out the result of running the two tests.

- **harness-classify-test**
  Classify test file(s) from "test/unclassified" to "test/passed" or "test/failed". This command can take 3 arguments. The argument "*-p*" means "pass" or "*-f*" means "fail" is required for classification. This command also supports regular expression of test files names if you take "*-R*" this optional argument. Additionally, if you wish to make all tests to pass/fail, you could just use "*-ALL*" argument. Example:

  ```
  $ harness-classify-test -p -ALL
  ```

  Make all test files passed.

  ```
  $ harness-classify-test -f test_unit_isTriangle
  ```

  Make test file named "test_unit_isTriangle" to failed.

  ```
  $ harness-classify-test -p -R test_unit_
  ```

  Make all test files naming match regular expression "test_unit_" to passed.

- **harness-report-last**
  Report on the last execution of the test harness. In fact, this report the current status of harness. Will return a print out report of how many tests have been run, how many passed, how many failed, how many unclassified. Support an optional argument "*-V*" or "*–verbose*" for a verbose report. This verbose report will contains test files name for showing passed/failed/unclassified information.

- **harness-report-all**
  Report on all test runs that have been run since the initialization, or reinitialization, of the harness. Show how many tests run, passed and failed for current status.

- **harness-report-test**
  Report on the history of a specific test or set of tests. Support regular expression of test file name with an optional argument "*-R*". This command will basically report every single line in the history file if is related to the given test files. Additionally, if you take the optional argument "*-s*" or "*–status*", the report will show you how many times this test has been run, passed, failed.
  Example:

```
$ ./harness-report-test -R test_unit_* -s
```

Get a report of all tests name matching regular expression "test_unit_*", and how many times each tests have been run, passed, failed.

- **harness-result-diff**
  Shows the difference between a previous result and the current result. By default, this command will show the difference between last time and current. If take with the optional argument "*-i*" or "*–index*", following with the history index number, the command will show the difference between the given history index with current status of test results. This command also support regular expression with optional argument "*-R*".
  Example:

  ```
  $ ./harness-result-diff -R test_unit_*
  ```

  Report the difference between current result and last time result for all tests name matching regular expresion "test_unit_*".

  ```
  $ ./harness-result-diff test_unit_isTriangle -i 2
  ```

  Report the difference between current result and the historical index 2 result for test "test_unit_isTriangle".

- **harness-test-show**
  Print out a or a set of specific test(s) for viewing. This command support an optional argument "*-R*" for taking regular expression for test file names.

This is how all the commands work for my test regression harness. **REMEMBER**, if forget how to use a single command, just take the ***"-h" or"–help"*** argument with the command. This will show you the help documentation for that command, about how to run the command, what arguments you could take, and some examples. Any further information please look into the code. The code should be well self documented.

SUGGESTED WAY TO TEST

1. Run *harness-init* at the first time testing(installing) the harness. Then put the test files into "test/inactive" or "test/new" directory.
2. Run *harness-run-all* or *harness-run-test* to run all/some tests. Use argument "*-V*" to turn on the verbose output to see the running results in console immediately.
3. Run *harness-classify-test* to classify the runned tests. The tests results were stored in "results/unclassified".
4. Run *harness-report-last* or *harness-report-all* to check the report of status of harness.
5. Run *harness-test-show* to show any tests wish to see or check.
6. Run *harness-restart* to restart a new turn of harness.
7. Run *harness-run-all* or *harness-run-test* for the same purpose as *step 2*.
8. Run *harness-result-diff* to see the difference between current result and last time(historical) result of any tests wish to check.
9. *[Optional]* Run *harness-classify-test* for the same purpose as *step 3*.
10. Run *harness-report-test* to check the report of runned status of any tests.

You could do the above steps recursively for testing. **NOTE:** if run *harness-init*, **ALL** history, results will be swiped out! If you just want to start a new turn of regression test, please run *harness-restart*.