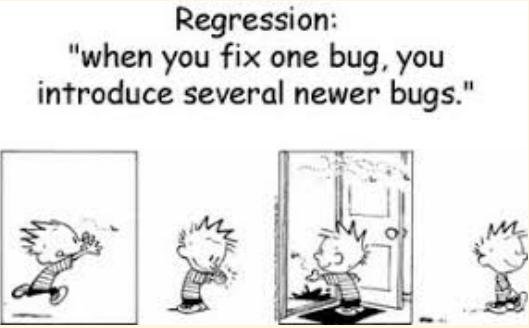


Regression Testing

The importance of regression testing for reliable software

Regression testing

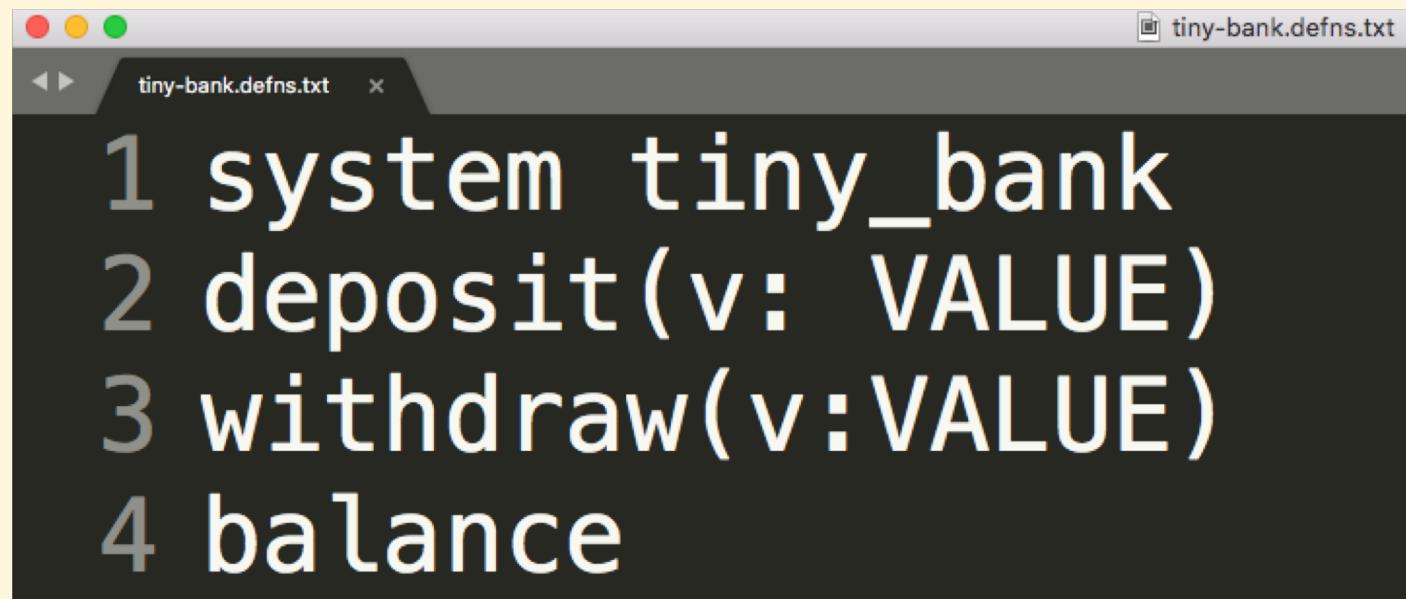
- Testing that ensures that previously developed and tested software still performs the same way after it is changed or interfaced with other software.
- Changes may include software enhancements, patches, configuration changes, etc. During regression testing, new software bugs or regressions may be uncovered.
- The purpose of regression testing is to ensure that changes such as those mentioned above have not introduced new faults
- Re-run previously completed tests



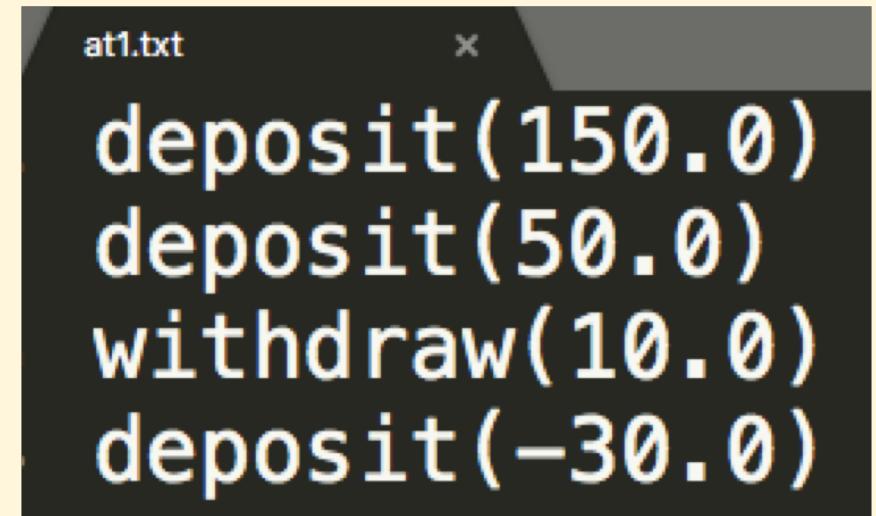
Regression testing: ESpec Unit/Specification Tests

PASSED (33 out of 33)		
Case Type	Passed	Total
Violation	5	5
Boolean	28	28
All Cases	33	33
State	Contract Violation	Test Name
Test1	TEST_STACK_ARRAY	
PASSED	NONE	t0: create empty stack
PASSED	NONE	t1: create stack, pop and push
PASSED	NONE	t2: test array implementation
Test2	TEST_STACK_LIST	
PASSED	NONE	t0: create empty stack
PASSED	NONE	t1: create stack, pop and push
PASSED	NONE	t2: test list implementation
PASSED	NONE	*t3: test {STACK_LIST}.push post-condition of push not satisfied with bad implementation
Test3	TEST_STACK_ARRAY_EVAL	
PASSED	NONE	t0: Evaluate (16.2 +4.1*2.4)
PASSED	NONE	t1: Evaluate (16.2) and also error condition
PASSED	NONE	t2: Evaluate (-922337267) and error condition

Regression Testing: ETF acceptance tests

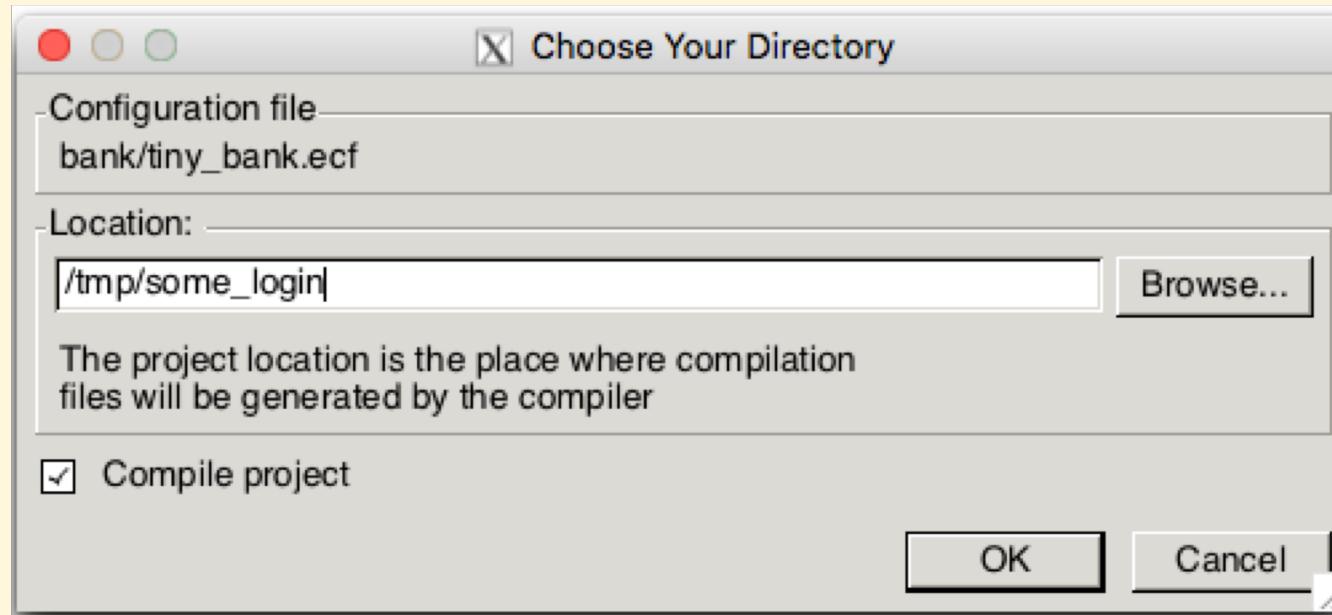


```
tiny-bank.defns.txt
1 system tiny_bank
2 deposit(v: VALUE)
3 withdraw(v:VALUE)
4 balance
```



```
at1.txt
deposit(150.0)
deposit(50.0)
withdraw(10.0)
deposit(-30.0)
```

Regression Testing: ETF acceptance tests



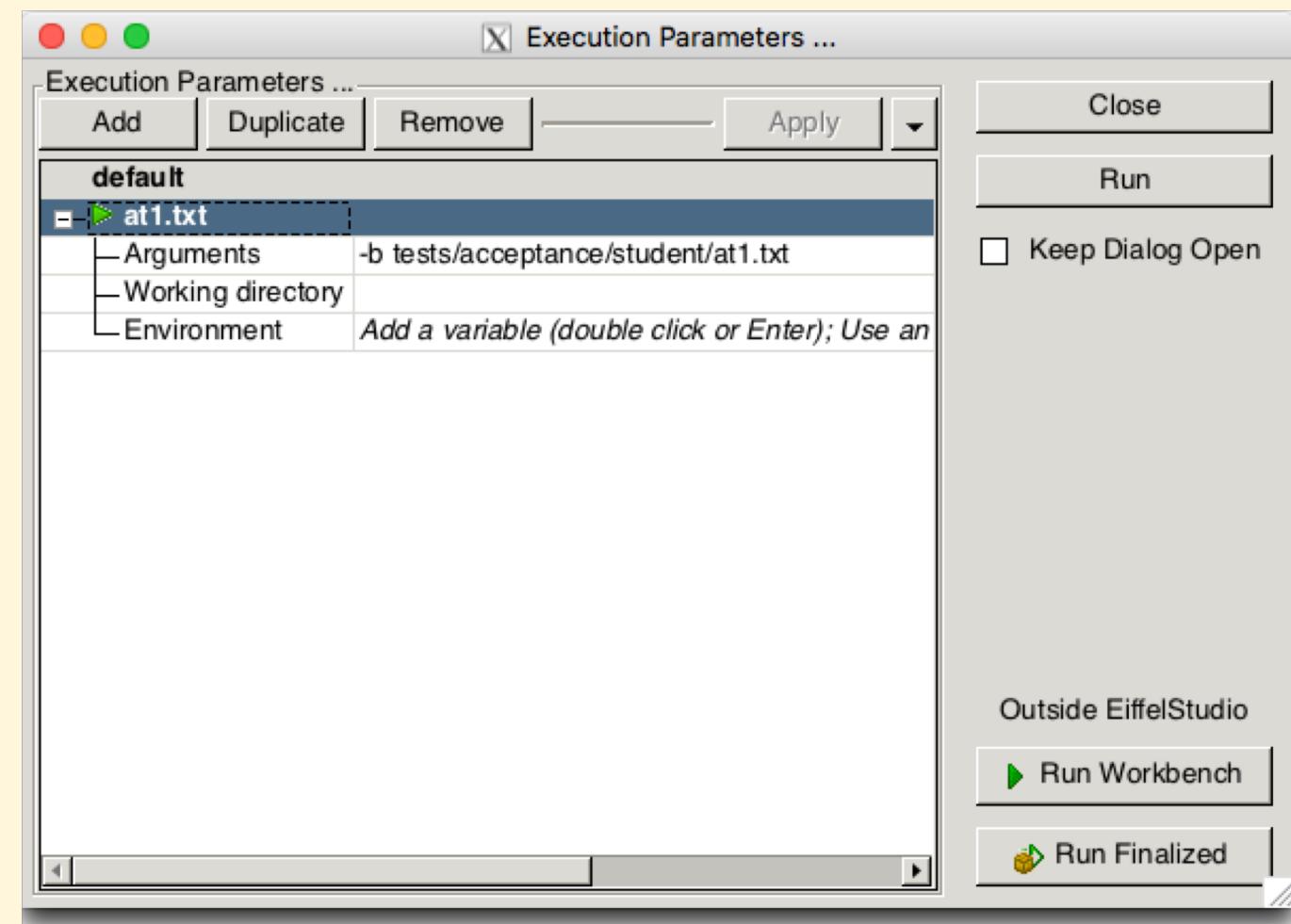
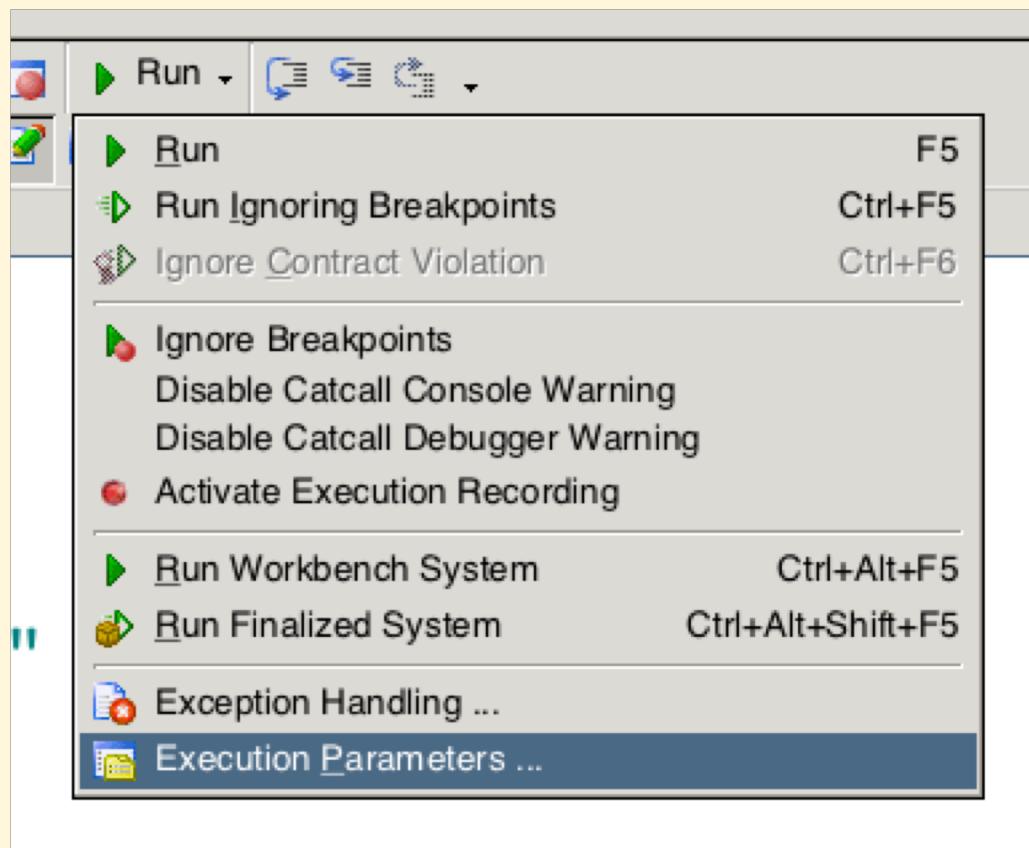
```
feature -- Attributes
    switch: INTEGER
        -- Running mode of ETF application.
    do
        Result := etf_cl_show_history
    end
```

Regression Testing: ETF acceptance tests

```
[jsom3% /tmp/some_account/EIFGENs/tiny_bank/W_code/tiny_bank -b at1.txt
  System State: default model state (0)
->deposit(150)
  System State: default model state (1)
->deposit(50)
  System State: default model state (2)
->withdraw(10)
  System State: default model state (3)
```

```
[jsom3% /tmp/some_account/EIFGENs/tiny_bank/W_code/tiny_bank -b at1.txt
  balance: 0.00
->deposit(150)
  balance: 150.00
->deposit(50)
  balance: 200.00
->withdraw(10)
  balance: 190.00
```

Regression Testing: ETF acceptance tests



Execution parameters: Use the debugger

The screenshot shows a debugger interface with the following details:

- Title Bar:** ETF_DEPOSIT
- Toolbar:** Includes icons for file operations, search, and other common functions.
- Status Bar:** tiny_bank BANK deposit
- Call Stack:** Status = Implicit exception pending
PRECONDITION_VIOLATION raised
- Call Stack Table:**

In Feature	In Class	From Class	@
deposit	BANK	BANK	1
deposit	ETF_DEPOSIT	ETF_DEPOSIT	1+1
fast_call	PROCEDURE	PROCEDURE	0
call	PROCEDURE	PROCEDURE	5
apply	PROCEDURE	PROCEDURE	2
run_input_co...	ETF_ABSTRACT_UI	ETF_ABSTR...	2+2
execute	ETF_SOFTWARE...	ETF_SOFTW...	9+1
exec	ROOT	ETF_CMD_LI...	3
handle_batch...	ROOT	ETF_CMD_LI...	5
make	ROOT	ETF_CMD_LI...	27
make	ROOT	ROOT	16
- Code View:** Shows the code for the deposit feature:

```
deposit (v: VALUE)
  require
    v > v.zero
  do
    balance := balance + v
  end
```

Regression Testing: ETF acceptance tests

```
 1 #!/usr/bin/env python3
 2
 3 # Specify the directory (relative or absolute) where the acceptance test
 4 # files are. If nothing is added, current directory is used.
 5 # WARNING: While absolute paths are supported, the oracle may not play nicely
 6 # with them.
 7 root = "..."
 8
 9 # Specify files or directories (relative to root) of the acceptance tests.
10 # NOTE: Tests can not be named test.expected.txt
11 # NOTE: To run all tests in root directory, use [""] as the test.
12 # WARNING: Does not support partial file names with wildcards ie at*.txt
13 acceptance_tests = [
14     "tests/at1.txt",
15     "tests/at2.txt"
16 ]
17
18 # Should regression testing continue on warnings? If this is set to False,
19 # regression testing will exit.
20 ignore_warnings = True
21
22 # Should the regression testing proceed with a list of expected files, or the
23 # oracle?
24 # When this flag is set True, the value of 'oracle' is ignored.
25 is_expected = True
26
27 # Specify where you put the oracle.
28 oracle = ""
29
30 # Specify the path of the executable built from your project.
31 executable = "/tmp/student/EIFGENs/tiny_bank/W_code/tiny_bank"
32 # The above is an OSX executable
33 # Linux executable for students shall be called: oracle.exe
34
35 # Should the program only print tests that do not pass
36 print_only_errors = True
37
38 # Should the program format the output so it could easily be diffed:
39 # examples include "meld", "diff -y", etc.
40 # If left blank will format output in a generic way
41 format_for_diff = "meld"
42
43 # Set true if you want the comparison to be tolerant on white spaces and empty
44 # lines.
45 # WARNING: While every attempt has been made to make the output return the same
46 # result as the grading script, there are no guarantees. It is strongly advised
47 # that you submit a program that passes with the flag set to False.
48 is_tolerant_on_white_spaces = False
49
```

Python3
regression script

```
▶ ▶ ETF_Test_Parameters.py x
1 #!/usr/bin/env python3
2
3 # Specify the directory (relative or absolute) where the acceptance test
4 # files are. If nothing is added, current directory is used.
5 # WARNING: While absolute paths are supported, the oracle may not play nicely
6 # with them.
7 root = "..."
8
9 # Specify files or directories (relative to root) of the acceptance tests.
10 # NOTE: Tests can not be named test.expected.txt
11 # NOTE: To run all tests in root directory, uses [""] as the test.
12 # WARNING: Does not support partial file names with wildcards ie at*.txt
13 acceptance_tests = [
14     "tests/at1.txt",
15     "tests/at2.txt"
16 ]
17
18 # Should regression testing continue on warnings? If this is set to False,
19 # regression testing will exit.
20 ignore_warnings = True
21
22 # Should the regression testing proceed with a list of expected files, or the
23 # oracle?
24 # When this flag is set True, the the value of 'oracle' is ignored.
25 is_expected = True
26
27 # Specify where you put the oracle.
28 oracle = ""
29
30 # Specify the path of the executable built from your project.
31 executable = "/tmp/student/EIFGENs/tiny_bank/W_code/tiny_bank"
32 # The above is an OSX executable
33 # Linux executable for students shaler be called: oracle.exe
34
35 # Should the program only print tests that do not pass
36 print_only_errors = True
37
38 # Should the program format the output so it could easily be diffed:
39 # examples include "meld", "diff -y", etc.
40 # If left blank will format output in a generic way
41 format_for_diff = "meld"
42
43 # Set true if you want the comparison to be tolerant on white spaces and empty
44 # lines.
45 # WARNING: While every attempt has been made to make the output return the same
46 # result as the grading script, there are no guarantees. It is strongly advised
47 # that you submit a program that passes with the flag set to False.
48 is_tolerant_on_white_spaces = False
49
```

red% ./ETF_Test.py

File ..tests/at1.expected.txt is copied to log/tests/at1.expected.txt.

File ..tests/at2.expected.txt is copied to log/tests/at2.expected.txt.

=====

Running acceptance test from file ..tests/at1.txt.

/Users/student/tiny-bank/tests/at1.txt

Output produced by /tmp/student/EIFGENs/tiny_bank/W_code/tiny_bank wrote to
log/tests/at1.actual.txt.

=====

Running acceptance test from file ..tests/at2.txt.

/Users/student/tiny-bank/tests/at2.txt

Output produced by /tmp/student/EIFGENs/tiny_bank/W_code/tiny_bank wrote to
log/tests/at2.actual.txt.

=====

Test Results: 1/2 passed.

=====

meld log/tests/at1.actual.txt log/tests/at1.expected.txt &

=====

Test Results: 1/2 passed.

=====

Need of Regression Testing

Regression Testing is required when there is a

- Change in requirements and code is modified according to the requirement
- New feature is added to the software
- Defect fixing
- Performance issue fix

