

LAB 2 MIF08 : calculator!

October 22, 2017

Arithmetic calculator

Input: a txt file:

```
$ TP02/ariteval$ cat testfiles/test01.txt  
20 + 22;  
a = 4;  
a + 2;  
a * 5;
```

Output: on std output:

```
20+22 = 42  
a now equals 4  
a+2 = 6  
a*5 = 20
```

Arithmetic calculator, how ?

- ▶ An ANTLR Grammar for arithmetical expressions, a slight modification of the one seen in the course:

```
expr:  
  | expr '+' term  
  | expr '-' term  
  | term  
  ;
```

```
term:  
  | term '*' fact  
  | term '/' fact  
  | fact  
  ;
```

```
fact:  
  INT  
  | ID  
  | '(' expr ')'  
  ;
```

- ▶ You can test it (with `grun!`): give expressions and observe parse trees.

Code Infrastructure

```
...code/ariteval$ ls  
Arit2.g4  arit2.py  Makefile  test_ariteval.py  testfiles/
```

- ▶ The grammar will be implemented in `Arit2.g4`.
- ▶ The main file (command line, driver for the lexer/parser):
`arit2.py`.
- ▶ A Makefile.
- ▶ Testfiles, and a test script `test_ariteval.py`.

Arithmetic calculator, test infrastructure

You write your testcases and expected results:

```
# first ex
```

```
20 + 22;
```

```
a = 4;
```

```
a + 2;
```

```
a * 5;
```

```
# EXPECTED
```

```
# 20+22 = 42
```

```
# a now equals 4
```

```
# a+2 = 6
```

```
# a*5 = 20
```

```
# second ex
```

```
a = 4 +b ;
```

```
# EXPECTED
```

```
# b is undefined
```

```
# EXITCODE 1
```

⇒ an helper script (using pytest) will compare the actual and the expected outputs.

Arithmetic calculator, test infrastructure 2/2

```
laure@sorlin:~/Documents/VCS/Teaching/archico/M1_COMPIL_MIF08/2017_2018/TP/TP02/code/ariteval$ make tests
python3 -m pytest -v test_ariteval.py
===== test session starts =====
platform linux -- Python 3.5.2, pytest-3.2.2, py-1.4.34, pluggy-0.4.0 -- /usr/bin/python3
cachedir: .cache
rootdir: /home/laure/Documents/VCS/Teaching/archico/M1_COMPIL_MIF08/2017_2018/TP/TP02/code/ariteval, inif
collected 6 items

test_ariteval.py::TestEVAL::test_expect[./testfiles/test03.txt] PASSED
test_ariteval.py::TestEVAL::test_expect[./testfiles/test05_unary.txt] PASSED
test_ariteval.py::TestEVAL::test_expect[./testfiles/test04_prio.txt] PASSED
test_ariteval.py::TestEVAL::test_expect[./testfiles/test01.txt] PASSED
test_ariteval.py::TestEVAL::test_expect[./testfiles/test02.txt] PASSED
test_ariteval.py::TestEVAL::test_expect[./testfiles/bad01.txt] PASSED

===== 6 passed in 0.62 seconds =====
```