

Part 2

Interpret

Interpret is a script that reads XML input and executes all instructions.

It has three possible arguments:

- help prints help
- source=<path> source code with XML input
- input=<path> STDIN input for XML instructions

Outputs are then printed on STDOUT.

Correct IPPcode20 XML representation starts with “<?xml version="1.0" encoding="UTF-8"?>” and continues with Program, that has only Instructions in it. First all instructions that contains a label are processed and saved then all other instructions are processed. The instruction order must be increasing by one with every instruction.

Interpret inst fully finished.

Tests

Script test.php is used in combination with scripts parse.php and interpret.py to test their correctness.

Test can be run with different arguments:

- help prints help
- directory=<path> path to the directory, where all tests are (if missing, looks for test in this directory)
- recursive if this parameter is used, tests are executed in subdirectories also
- parse-script=<path> path to the parse script (if missing, looks for parse.php in this directory)
- int-script=<path> path to the interpret script (if missing, looks for interpret.py in this directory)
- parse-only if this parameter is used, only parser is tested (can't be combined with --int-only or --int-script=<path>)
- int-only if this parameter is used, only interpreter is tested (can't be combined with --parse-only or --parse-script=<path>)
- jexamxml=path path to the directory, with a A7Soft JExamXML tool for comparing XML files

After all arguments are checked, and all important values stored, specified directories are searched for tests. If a *.src file exist, script looks for files with the same name but with suffixes .rc, .in and .out. If they don't exist already, new files are created (with a default content).

Depending on the arguments, parser or interpreter is called with parameters from test files. Results are then compared and saved for later use. After all tests are finished, all results are converted into well-arranged html text that is then printed on STDOUT.

Html table is created with 8 attributes for every test:

1. ID – for every test
2. Test name – name of the test
3. parse-only – checks if an argument --parse-only was used
4. int-only – checks if an argument --int-only was used
5. Exp. value – expected return value of parser/interpreter
6. Real return value – value received after executing tests
7. Identical output – if a comparison of outputs is needed, prints a check or cross
8. Passed – checks if the test run successfully

Every directory is checked separately and a summary is displayed. Full summary is displayed on the end of the table with a number of passed tests.