Implementation documentation for the 2. task for IPP 2018/2019
Name and surname: Sabína Gregušová
Login: xgregu02

1 Introduction

My task was to create two script. One in php for testing purposes that generates html with all the information about conducted tests. The second one was interpret for language IPPcode19 that accepts input in XML format.

2 Interpreting

3 Testing

I have implemented script for testing in file `test.php` very early, so that I could use it for testing first part of this project. The script systematically checks all the possible input parameters and updates their existence in `arguments` array. The information about their existence is stored as boolean. In case of arguments that include path, the path is immediately parsed and stored for later use. I check for forbidden use of parameters and make sure all necessary directories and files exists.

I created my template output in HTML and CSS separately and after all arguments are checked, I generate first part of HTML output, which primarily consist of some basic CSS and headers. Then, I use use my object `Tests` which first reads the directory and looks for files with `src` extension. Once it encounters file with `src` extension, it checks for all the other necessary files with `in`, `out` and `rc` and generates those that are missing as empty files or files with 0. If it the `--recursive` argument is in effect, it also checks for directories and recursively calls function `run_tests` in the same way as described above.

Each test is run in function `run_test` which has 3 possible branches: for testing only parser, only interpret or both. Each test checks the expected return code with the returned code and if they match, the test is considered succesful. Additionally, if the returned code is 0, the reference solution is compared to the real solution and the last status is determined by this comparison. I used simple diff tool for this comparison, except when I compared two XML files, for which I used xmldiff that is available on school server MERLIN.

Each test the generates a piece of HTML code that includes information about its ordinal number, full path, name of file and its status. I decided to distinguish the status for better visibility, red color for failed test and green color for passed test. Each temporary file is properly deleted after its use at the end, I generate brief statistics with percentage of passed tests and total number of tests.