European Funds
Knowledge Education Development
Wrocław University of Science and Technology
European Union
European Social Fund

ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej

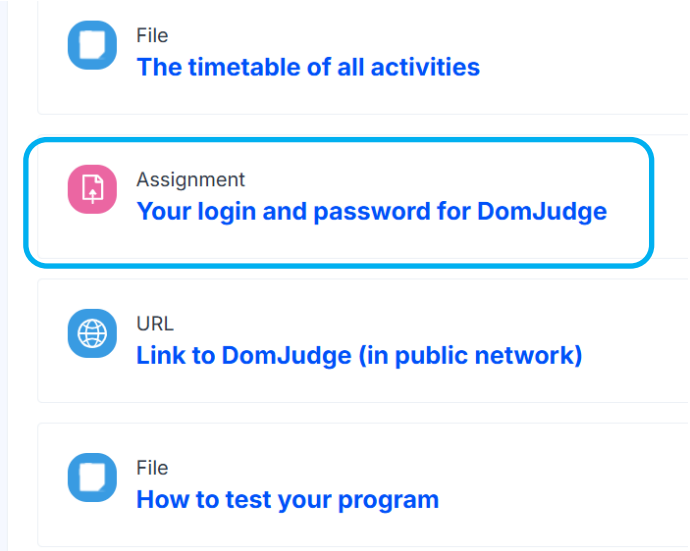
Laboratory – List 01

Introduction.

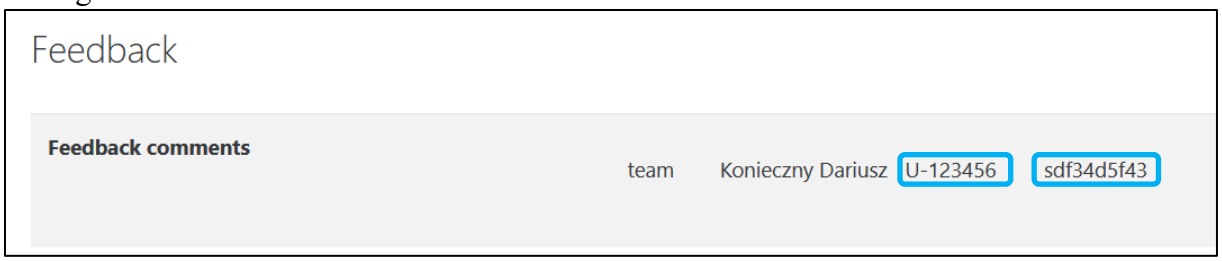
During laboratories we will use automatic checking system. To prepare for that you have to do following steps.

1. Pseudo-Task "Your login and password for DomJudge".

Choose a pseudo-Task "Your login and password for DomJudge" I "Common Information" section:



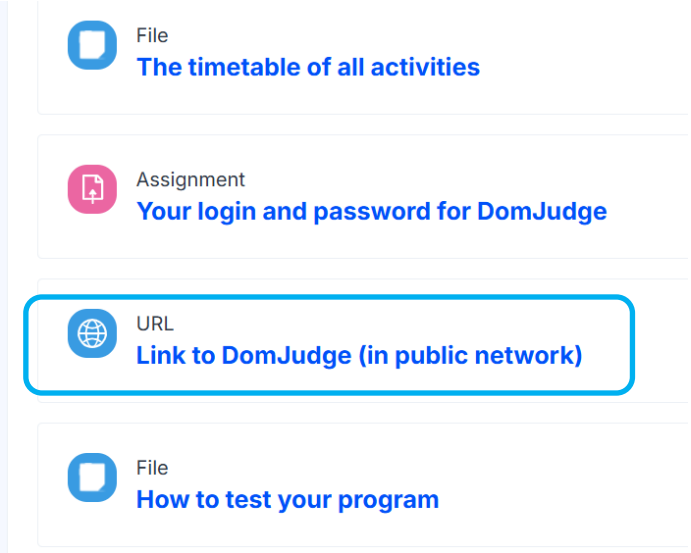
Then go down to see " Feedback comments " which can be like follows:



The string with prefix "U-" is the name of login, so in the example it is "U-123456". The second is the password for this account – "sdf34d5f43".

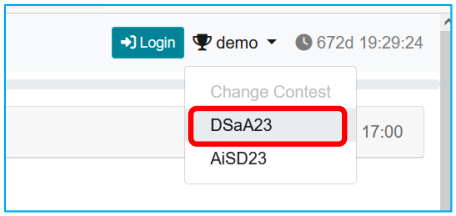
2. Using judgement system

Now you can login into judgement system. Go to main page of the course and choose "Link ro DomJudge (in public network)":



This will open new webpage with the URL http://domj.wit.pwr.edu.pl:25756/domjudge/public.

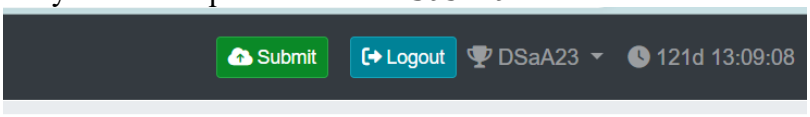
Remember to choose correct contest “DSaA25” (screenshot is from previous semester):



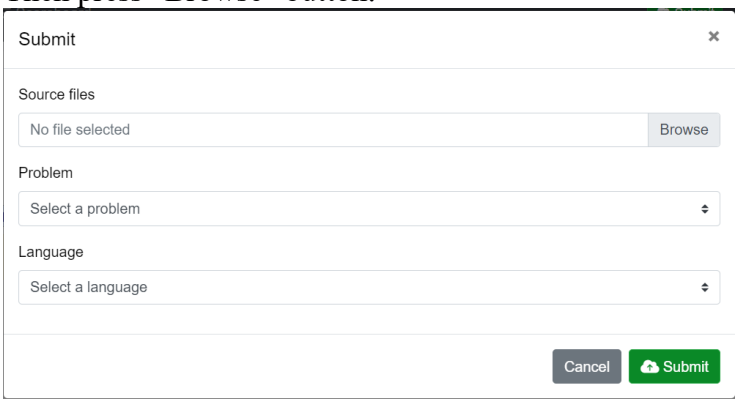
And then press the “Login” button:

For logging use the date from the pseudo-task from point 1:

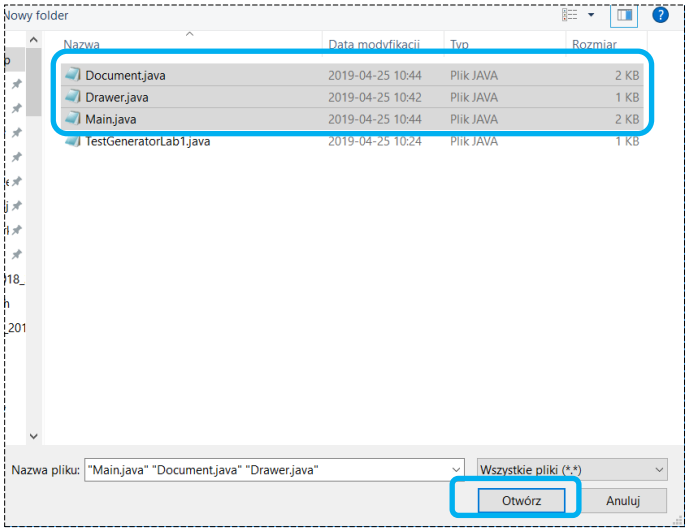
Now you can send you solution of a task list. For example if you want to send solution for task 1.1 you have to press a button “Submit”:



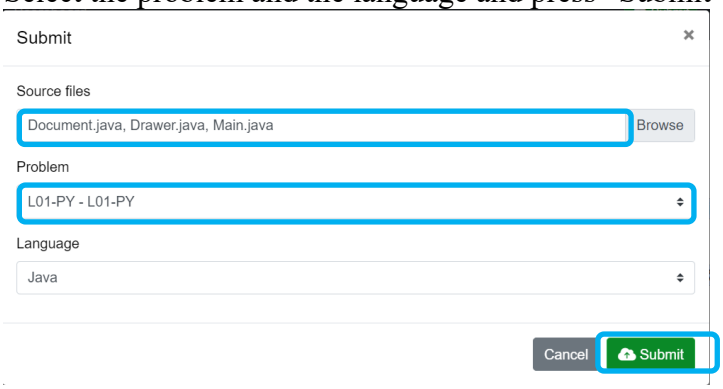
Then press “Browse” button:



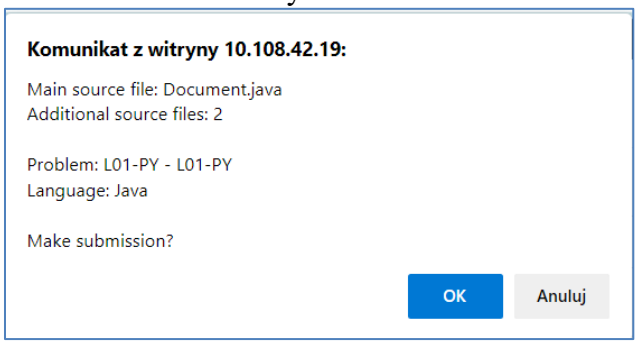
To select all needed files:



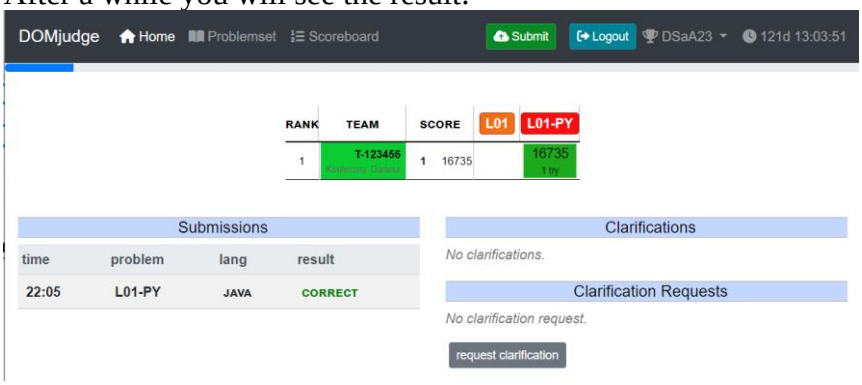
Select the problem and the language and press “Submit”:



Confirm submission by “OK” button and wait for a result:



After a while you will see the result:



Of course the result can be also negative, like:

- Compile-error
- Wrong-answer
- Run-error
- Time-limit

If you click selected submission you will see a little more information about the error. Full information about problems is available only from the teacher's account.

For every task list there will be prepared a template files which will be prepared for automatic testing using the judgement system. So you have to copy these files and fulfill it with your solution of the problem. This approach also allows for easy self-testing of the solution by entering commands of a certain simple language. Information about the language commands of a given task list can be found after the list of tasks. There will be also an example test (using the language) and correct answer for the test.

Task list

1. Write a procedure:

a) `drawPyramid(int n)` which takes as an input one integer value *n* and then output on console a pyramid as on figure below (for example for *n*=4):

```
...X...
..XXX..
.XXXXX.
XXXXXXX
```

b) `drawChristmasTree(int n)` which takes as an input one integer value *n* and then output on console a Christmas tree in which last part height equals *n*. The tree consists of pyramids of heights from 1 to *n*. The shape have to be as presented below (for *n*=4):

```
...X...
...x...
..xxx..
...X...
..XXX..
.XXXXX.
...x...
..xxx..
.xxxxx.
xxxxxxx
```

2. Write a procedure `loadDocument(String name)` which will load and analyze lines after lines searching for link in every line. Link is a word (the words are separated by whitespace, i.e. the line "Ala m4%a kota&." consists of 3 words "Ala", "m4%a" and "kota&."). The link format is as follows: 5 characters "`link=`" after which the is a correct identifier. The correct identifier starts from letter (small or capital) follows by letters or digits or underline '_'. The procedure has to print for every line all correct identifiers in a separated line. Before printing, the identifiers have to be changed to small letters. The document ends with line with the text "`eod`", which means end of document.

For 100 points present solutions for this list till Week 2.
For 80 points present solutions for this list till Week 3.
For 50 points present solutions for this list till Week 4.
After Week 4 the list is closed.

The solution will be automated tested with tests from console of presented below format.

Program start with one line with a string "START".

If an input line starts from '#' sign or a line is empty, the line have to be ignored.
Else the input line have to be copied to output line with exclamation mark before first character. Then the proper operation have to be done.

If a line has a format:
`py` *`n`*
your program has to call `drawPyramid(n)`. You can assume that 1<=*n*<=20.

If a line has a format:
`ct` *`n`*
your program has to call `drawChristmasTree(n)`. You can assume that 1<=*n*<=20.

If a line has a format:
`ld` *`docName`*
your program has to call `loadDocument(String docName)`.

If a line has a format:
`ha`

your program has to end the execution, writing as the last line "`END OF EXECUTION`". Every test ends with this line.

For example for a test file (a part in green frame is a "document"):

```
py 3
ct 3
ld qwert
nnothing is here
link=abc link=qWe link=asd
link= broken li nk=wrong link =not
link=ok123_23sd what is here link=12wRong asdad link=_what12
dfasfdsdfsd
and now start LINK=$2323 LiNk=Ok
eod
ha
```

the output has to be:

```
START
!py 3
..X..
.XXX.
XXXXX
!ct 3
..X..
..X..
.XXX.
..X..
.XXX.
XXXXX
!ld qwert
abc
qwe
asd
ok123_23sd
ok
!ha
END OF EXECUTION
```