ECOTE regulation

Laboratory - 20 points :

- Introductory project - 5
- Program - 12
- Final documentation – 3

The Introductory project has to be accepted by the supervisor before the program development. During laboratories a given program should be designed, implemented and tested. **It is obligatory to show the execution of the program and to present the technical documentation.**

1

Anna Derezińska

Anna.Derezinska@pw.edu.pl

USOS group 101 Wednesday8.15-10, lab 011

USOS group 104 Wednesday14.15-16, lab 09

MS Teams: *ECOTE2023L_LAB(101&104)* channel

- assignments to collect documentation and code

Do not submit any projects via mail or chat!

2

Introductory project contains:

- Functional specification
- Input/Output description
- Project specification (architecture, data structures – graphical representation, main algorithm), UML may be used to specify project.
- Test cases

ECOTEproject_pattern.doc (*MS Teams*)

pdf -> Teams Assignment

3

Laboratory timetable

12.04 – Introduction, discussion of subjects

19.04 – Presentation of **first version** of introductory **project**

26.04 – **Deadline** for introductory project (final version)

10.5, 17.5 and 24.05 – Presentation of **preliminary version of program**

31.05– **Deadline** for program presentation

07.06 - Deadline for final documentation, last laboratory

14.06 – Results of laboratory

4

Groups 101 and 104:

Basic ECOTE tasks have been assigned (files in Teams).

There is also a set of free tasks.

A student can ask for changing from a basic task to a free one (during the first week).

The **detailed scope** of the task has to be decided individually. It strongly depends on the language selected, grammar, the tools used etc.

Remark: *A solution cannot be mimicked by a ready Internet program, a Python library, etc.*

5

Lab task – (*if appropriate*)

- Prepare a simple scripting language to automate certain domain specific activities.
- Could be used for (or only inspired by) a diploma thesis
- Implement a translator of the language to ...,

  or an interpreter

Or write another ECOTE-like program that might be useful for the diploma thesis

6

ANTLR

ANother Tool for Language Recognition

https://www.antlr.org/

parser generator

Tool that translates a grammar to a parser/lexer in Java (or other target language) and the runtime library

The parser can build and walk parse trees

7

Antlr - sample

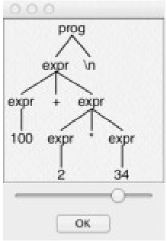
```
grammar Expr;
prog:    (expr NEWLINE)* ;

expr:    expr ('*'|'/') expr
         | expr ('+'|'-')  expr
         | INT
         | '(' expr ')’
         ;

NEWLINE : [\r\n]+ ;

INT :    [0-9]+ ;
```

```
$ antlr4 Expr.g4
$ javac Expr*.java
$ grun Expr prog -gui
100+2*34
^D
```



8

Generators of lexical analyzers

Lex, Flex – programs for generation of a lexical analyzers , i.e. tools that split the source file into tokens

Lex - Reads input with rules of the lexical analysis (regular expressions) and generates source code of lexical analyzer (in C)

Analysis is performed on DFA

Flex – Fast Lexical analyzer generator (free open source),

variant of Lex, (output in C,C++)

https://github.com/westes/flex

9