IFJ 2021 compiler

xpojez00, xbudin05, xhribi00, xchupa03

December 5, 2021

1 Introduction

This paper contains solutionists of this project. First page contains formal requirements. Our solutions were originally discussed on separate papers. We have decided to merge all pdfs together with this one. We apologize in advance for this inconvenience.

Tím 084, variant II

Denis Pojezdál xpojez00 - vedúci	30
Rastislav Budinský xbudin05	40
Samuel Hríbik xhribi00	15
Sebastián Chupáč xchupa03	15

Implementované rozšítenia:
BOOLTHEN
CYCLES
FUNEXP

1.1 Point distribution

Lexical analyzer: FSM was created together implemented byt xbudin05. Tested by xpojez00 and xbudin05
Syntax analyzer: Created, implemented and tested by xpojez00 and xbudin05.
Custom LL-grammar generator[1]: Created, implemented and tested by xpojez00.
Semantic analyzer: Created by xpojez00 and xbudin05. Implemented and tested by xbudin05.
Symbol table and Utils[2]: Created, implemented and tested by xpojez00 and xbudin05.
Abstract Syntax Tree: Created by xpojez00 and xbudin05. Implemented and tested by xbudin05.
Built in functions: Created implemented and tested by xhribi00 and xchupa03.
Code generation: Created, implemented and tested by xbudin05.
BOOLTHEN and FUNEXP: Created and implemented by xpojez00 and xbudin05. Tested by xbudin05.
CYCLES: Created, implemented and tested by xbudin05.
Documentation: Written by xbudin05.

[1] Implemented in C++, not a part of submitted but helped during LL-grammar implementation
[2] Stack, list etc.