|  |
| --- |
| Київський національний університет  «Київський політехнічний інститут» |
| Факультет інформатики та обчислювальної техніки |
| Кафедра обчислювальної техніки |
| Лабораторна робота №5 |
| з дисципліни «Системне програмування» |
| *Залікова книжка № 4213* |
|  |
| Виконав студент 3-го курсу |
| групи ІО-42 |
| Кочетов Данило |

|  |
| --- |
| 2016 |

Мета: Одержання навичок створення механізмів синтаксичного розбору методом рекурсивного спуска і створення обробника синтаксичних помилок вхідного тексту.

|  |  |  |
| --- | --- | --- |
| № вар. | Вираз, який відтворюється в дереві розбору | Мова відтворення |
| 13 | begin if (a>j) then begin if (p<n) then begin k:=t; end; end; end. | Pascal |

# Лістинг програми

**Lab5.cpp**

#include "token.h"

#include "visgrp.h"

#include "tables.h"

#include "lexan.h"

#include "langio.h"

#include <stdio.h>

#include <string.h>

#include <conio.h>

#include "synan.h"

#include <stdlib.h>

using namespace std;

extern struct recrdKWD \*tablKWD;

struct lxNode nodes[200] = // масив приймач вузлів дерева

{ { \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },{ \_nil,NULL,NULL,0,0,0,0,0,NULL,0 },

};

char file\_name[20];

extern enum ltrType ltClsC[256];

extern enum ltrType ltClsP[256];

extern enum tokType dlCdsC[256];

extern enum tokType dlCdsP[256];

extern enum ltrType ltClsC[256];

extern enum ltrType ltClsP[256];

enum ltrType \*ltCls = ltClsC;

enum tokType \*dlCds = dlCdsC;

int main(int argc, char\* argv[]) {

int nn = -1;//np,

if (argc > 1) {

strcpy(file\_name, argv[1]);

printf("Processing file -- %s\n", file\_name);

}

else {

printf("Please enter file Name: ");

scanf("%s", file\_name);

strcat(file\_name, ".h");

}

opFls(file\_name);

//opFls("v1.h");

LxAnInit('P');

srtBin(tablKWD, 67);

do {//np=nn;

nn = LxAnlzr();

} while (nodes[nn].ndOp != \_EOF);

//Начало синтаксического анализа

synNode\* mainNode = synAnalysis(nodes);

//Конец синтаксического анализа

mainNode->toString(0);

//prLaTxt(nodes, nn);

getch();

getch();

system("pause");

return 0;

}

