|  |
| --- |
| Київський національний університет  «Київський політехнічний інститут» |
| Факультет інформатики та обчислювальної техніки |
| Кафедра обчислювальної техніки |
| Лабораторна робота №2 |
| з дисципліни «Системне програмування» |
| *Залікова книжка № 4213* |
|  |
| Виконав студент 3-го курсу |
| групи ІО-42 |
| Кочетов Данило |

|  |
| --- |
| 2016 |

Мета: Вивчення методів створення вузлів дерево­по­діб­них та ієрархічних графів, організації доступу до інформації, ре­кон­струкції вхідного тексту та скорочення графів через вилу­чення повторних вузлів.

|  |  |  |  |
| --- | --- | --- | --- |
| № вар. | Вираз, який відтворюється в графі внутрішнього подання | Настроювання графа автомата з послідовними станами | Мова відтво­рення |
| 13 | for(b=0;n;n--)b+=a[n]; | Стани 0..6; 3->3(*dlm*), 5->2(*ltr*) | С |

# Лістинг програми

## lab2.cpp

#include "stdafx.h"

#include <stdlib.h>

#include "visgrp.h"

#include "token.h"

#include "automat.h"

char \*imgs[100] = { "b", "0", "n", "a" };

struct lxNode token[100]=

{

{\_for, &token[4], &token[2], 0, 0, 0, 0, 0, NULL, 0}, // 0 - for

{\_myEOS, &token[2], NULL, 0, 0, 0, 0, 0, NULL, 0}, // 1 - ; (last)

{\_asAdd, &token[9], &token[3], 0, 0, 0, 0, 0, NULL, 0}, // 2 - +=

{\_ixbr, &token[12], &token[11], 0, 0, 0, 0, 0, NULL, 0}, // 3 - []

{\_brkt, NULL, &token[5], 0, 0, 0, 0, 0, NULL, 0}, // 4 - ()

{\_myEOS, &token[6], &token[7], 0, 0, 0, 0, 0, NULL, 0}, // 5 - ; (left)

{\_ass, &token[9], &token[10], 0, 0, 0, 0, 0, NULL, 0}, // 6 - =

{\_myEOS, &token[11], &token[8], 0, 0, 0, 0, 0, NULL, 0}, // 7 - ; (right)

{\_dcr, &token[11], NULL, 0, 0, 0, 0, 0, NULL, 0}, // 8 - --

{ \_nam, (struct lxNode\*)imgs[0], NULL, 0, 0, 0, 0, 0, NULL, 0 }, // 9 - b

{ \_nam, (struct lxNode\*)imgs[1], NULL, 0, 0, 0, 0, 0, NULL, 0 }, // 10 - 0

{ \_nam, (struct lxNode\*)imgs[2], NULL, 0, 0, 0, 0, 0, NULL, 0 }, // 11 - n

{ \_nam, (struct lxNode\*)imgs[3], NULL, 0, 0, 0, 0, 0, NULL, 0 } // 12 - a

};

// управляючі таблиці мов програмування

extern char \*oprtrC[], \*oprtrP[], \*oprtrV[], \*cprC[], \*cprP[], \*cprV[];

char \*\*oprtr = oprtrC, \*\*cpr = cprC,

modeP=0, // тип роздільника операторних дужок для Паскаля

modeC=1, // тип роздільника операторних дужок для С

modeL=modeC;

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

{

prLxTxt(token);

printf("\n\n");

printf("S0");

enum autSgn ASgn[7] = {sg0, sg0, sg0, sg0, sg0, sg0, sg0};

for(int n = 0; n < 7; n++)

printf("%5d->%2d ", ASgn[n], nxtStat(ASgn[n]));

printf("\n\n");

system("pause");

}

## automat.h

enum autStat

{

S0,

S1,

S2,

S3,

S4,

S5,

Se

};

enum autSgn

{

sg0, // sg0 - Нульовий сигнал

dlm, // dlm - Перший сигнал

ltr // ltr - Другий сигнал

};

enum autStat nxtStat(enum autSgn sgn);

## automat.cpp

#include "stdafx.h"

#include "automat.h"

enum autStat nxtSts[Se+1][ltr+1] =

{

{S1, S0, S0},

{S2, S1, S1},

{S3, S2, S2},

{S4, S3, S3},

{S5, S4, S4},

{Se, S5, S2},

{Se, Se, Se}

};

enum autStat nxtStat(enum autSgn sgn)

{

static enum autStat s=S0;// поточний стан лексеми

return s=nxtSts[s][sgn];// новий стан лексеми

}