Міністерство освіти і науки, молоді та спорту України

Національний Технічний Університет України

“Київський Політехнічний Інститут”

*Факультет прикладної математики*

*Кафедра СПіСКС*

**Лабораторна робота № 3**

*з дисципліни*

“Інженерія програмного забезпечення. Основи проектування трансляторів”

Тема: “Розробка генератора коду”

Виконав:

cтудент групи КВ-42

Сахнік Іван

Перевірив:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ 2017 р.

**Постановка завдання**

1. Разробити програму генератора коду (ГК) для подмножини мови програмування SIGNAL, заданої за варіантом.

2. Програма має забезпечувати: читання дерева розбору та таблиць, створених синтаксичним аналізатором, який було розроблено в розрахунково-графічній роботі; виявлення семантичних помилок;

генерацію коду та/або побудову внутрішніх таблиць для генерації коду.

3. Входом генератора коду (ГК) мають бути: дерево розбору; таблиці ідентифікаторів та констант з повною інформацією, необхідною для генерації коду; вхідна програма на підмножині мови програмування SIGNAL згідно з варіантом (необхідна для формування лістингу програми).

4. Виходом ГК мають бути: асемблерний код згенерований для вхідної програми та/або внутрішні таблиці для генерації коду; внутрішні таблиці генератора коду (якщо потрібні).

5. Зкомпонувати повний компілятор, що складається з розроблених раніше лексичного та синтаксичного аналізаторів і генератора коду, який забезпечує наступне: генерацію коду та/або побудову внутрішніх таблиць для генерації коду; формування лістингу вхідної програми з повідомленнями про лексичні, синтаксичні та семантичні помилки.

6. Входом компілятора має бути програма на підмножині мови програмування SIGNAL згідно з варіантом;

7. Виходом компілятора мають бути: асемблерний код згенерований для вхідної програми та/або внутрішні таблиці для генерації коду; лістинг вхідної програми з повідомленнями про лексичні, си-

нтаксичні та семантичні помилки.

8. Для програмування може бути використана довільна алгоритмічна мова програмування високого рівня. Якщо обрана мова програмування має конструкції або бібліотеки для роботи з регулярними виразами, то використання цих конструкцій та/або бібліотек строго заборонено.

***Варіант 15***

1.<signal-program> --> <program>

2.<program> --> PROGRAM <procedure-identifier> ;

<block> ;

3.<block> --> BEGIN <statements-list> END

4.<statements-list> --> <statement> <statements-list> | <empty>

5.<statement> --> <unsigned-integer> : <statement>|

<variable-identifier> := <unsigned-in-teger> ; |

<procedure-identifier> <actual-arguments> ; |

GOTO <unsigned-integer> ; |

LINK <variable-identifier> , <unsigned-integer> ; |

IN <unsigned-integer>; |OUT <unsigned-integer>; |

RETURN ; |; |

($ <assembly-insert-file-identifier> $)

6.<actual-arguments> --> ( <variable-identifier>

<actual-arguments-list> ) |

<empty>

7.<actual-arguments-list> --> ,<variable-identi-fier>

<actual-arguments-list> | <empty>

8.<variable-identifier> --> <identifier>

9.<procedure-identifier> --> <identifier>

10.<assembly-insert-file-identifier> --> <identi-fier>

11.<identifier> --> <letter><string>

12.<string> --> <letter><string> |

<digit><| 8 | 9string> | <empty>

13.<unsigned-integer> --> <digit><digits-string>

14.<digits-string> --> <digit><digits-string> | <empty>

15.<digit> --> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7

16.<letter> --> A | B | C | D | ... | Z

**Результат виконання програми**

code SEGMENT

ASSUME cs:code

begin:

01:

mov A, 10

mov B, 20

push A

push B

call MYPROC

jmp 01

draw\_point proc

push bp

mov bp, sp

mov ax, 0A000h

mov es, ax

mov si, crt\_y

mov di, crt\_x

cmp si, y\_res\_

jae @nd

cmp di, x\_res\_

jae @nd

mov ax, x\_res\_

mul si

add ax, di

mov bx, ax

mov dx, [bp+4]

mov byte ptr es:[bx], dl

ret 0

mov ax, 4c00h

int 21h

code ENDS

end begin

**Стандартний вміст таблиць**

**Таблиця роздільників**

|  |  |
| --- | --- |
| **Код** | **Значення** |
| 59 | ; |
| 58 | : |
| 44 | , |
| 40 | ( |
| 41 | ) |
| 301 | := |
| 302 | ($ |
| 303 | $) |

**Таблиця ключових слів**

|  |  |
| --- | --- |
| **Код** | **Значення** |
| 401 | PROGRAM |
| 402 | BEGIN |
| 403 | END |
| 404 | GOTO |
| 405 | LINK |
| 406 | IN |
| 407 | OUT |
| 408 | RETURN |