

Tiny Set Language

Oh조

권성철(200924413)

조용래(201224540)

목차

1. 개요
2. 어휘
3. 문법
4. 예시 프로그램

개요

Tiny Set Language(TSL)

C의 문법과 C++의 일부
기능

집합/논리

기타 잡다한 곳에서 본
것

개요

- 입출력 : 표준 입출력

표준 입력 : `read(id | id_list)`

표준 출력 : `write(id | id_list | NUM | BIT | LETTER | STRING)`

`(id_list -> id | ", " id_list)`

- 타입

Elementary Type	Set	Tuple
int	intset	inttup
char	charset	chartup
bool	bitset	

개요

- 제어문
 - If (expr) body| if(expr) body else body;
 - while(expression) body
 - for(assign_stmt; relational_expr; assign_stmt) body
 - repeat(number | identifier) body
- set 타입에 대한 새로운 순회 방법 적용
 - touch(set; condition; sort-criteria) body
 - set : 집합 타입의 변수
 - condition : 추출 함수(대상 집합에서 부분 집합을 추려내기 위한 함수)
 - sort-criteria : 정렬 함수(ex 오름차순)

어휘

프로그래밍 언어의 어휘 구성

키워드

명칭

상수

연산자

구두점

공백문자

주석

어휘

Keyword : 언어상 용도가 정해져 변수나 함수 이름으로 사용 불가

Keyword	Category
int	Type
char	Type
bool	Type
intset	Type
charset	Type
Inttup	Type
chartup	Type
bitset	Type
If	Control
else	Control
for	Control
while	Control
return	Function Return
touch	Selection + Iteration

어휘

명칭(Identifier)

- letter [A-Za-z]
- digit -> [0-9]
- id -> {letter}({letter} | {digit})*

상수(Constant)

- 정수 상수 : {digit}+
- 문자 상수 : {letter}
- 문자열 상수 : {letter}{{letter} | {digit}}+
- 논리형 상수 : true | false
- 비트형 상수 : {b}{0 | 1}+

어휘

구두점(Punctuation)

- ()
- {}
- []
- ;
- ,

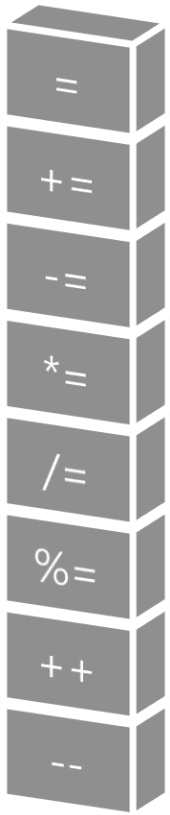
공백(White Space)

- " "
- \n

주석(Comment)

- //
- /* */

어휘



대입 연산자



산술 연산자



비교 연산자



비트 연산자



논리 연산자

문법

함수 선언/정의

⋮

```
START() {  
    변수선언...  
    CODE...  
    ...  
}
```

⋮

함수 선언/정의

프로그램 구성

문법

```
24 start_program : func_dec WHITE_SPACE START "(" body WHITE_SPACE func_dec
25                | START "(" body;
26 id_list : ID
27          | id_list ", " ID;
28 type    : INT
29          | CHAR
30          | INTSET
31          | CHARSET
32          | BITSET
33          | INTTUP
34          | CHARTUP
35          | BOOL;
36 id_num : ID | NUM;
37 dec : id_dec | func_dec;
38 id_dec : type " " ID;
39 func_dec : type " " ID "(" id_list ")" | type " " ID "(" id_list ")" body;
40
41 stmt : assign_stmt
42       | while_stmt
43       | for_stmt
44       | touch_stmt
45       | if_stmt
46       | repeat_stmt
47       | read | write
48       | function_call;
```

문법

```
51 body : "{" stmt_list "}" | dec;
52 assign_stmt : ID "=" assign_term
53              | ID "+=" assign_term
54              | ID "-=" assign_term
55              | ID "*=" assign_term
56              | ID "%=" assign_term
57              | ID "&=" assign_term
58              | ID "|=" assign_term
59              | assign_term;
60 assign_term : "++" ID | "--" ID | ID | NUM;
61
62 repeat_stmt : REPEAT "(" NUM ")" body | REPEAT "(" ID ")" body;
63 while_stmt : WHILE "(" stmt ")" body;
64 for_stmt : FOR "(" assign_stmt ";" relational_expr ";" assign_stmt ")" body;
65 touch_stmt : TOUCH "(" ID ";" ID ";" ID ")" body;
66 if_stmt : IF "(" expr ")" body | IF "(" expr ")" body ELSE body;
67
68 read      : READ "(" id_list ")";
69 write     : WRITE "(" id_list ")";
70 function_call : ID "(" id_list ")";
71
72 expr :    logical_expr
73        |  arithmetic_expr
74        |  relational_expr
75        |  bit_expr;
76
```

문법

```
77 bit_expr : bit_expr "|" bit_term
78           | bit_expr "~|" bit_term
79           | bit_expr "^" bit_term
80           | bit_term;
81 bit_term : bit_term "&" bit_factor
82           | bit_term "~&" bit_factor
83           | bit_term "&~" bit_factor
84           | bit_factor;
85 bit_factor : "~" bit_end | bit_end;
86 bit_end : BIT | "(" bit_expr ")";
87
88
89 relational_expr : id_num "==" id_num
90                  | id_num ">" id_num
91                  | id_num "<" id_num
92                  | id_num ">=" id_num
93                  | id_num "<=" id_num
94                  | id_num "!=" id_num;
95
96 arithmetic_expr : arithmetic_expr "+" arithmetic_term
97                  | arithmetic_expr "-" arithmetic_term
98                  | arithmetic_term;
99 arithmetic_term : arithmetic_term "*" arithmetic_factor
100                 | arithmetic_term "/" arithmetic_factor
101                 | arithmetic_term "%" arithmetic_factor
102                 | arithmetic_factor;
103
104 arithmetic_factor : ID | NUM | "(" arithmetic_expr ")";
105
106
107 logical_expr : logical_expr "||" logical_term | logical_term;
108 logical_term : logical_term "&&" logical_factor | logical_factor;
109 logical_factor : logical_factor "!" logical_end | logical_end;
110 logical_end : TRUE | FALSE | "(" logical_expr ")";
```


예제 프로그램

```
1 bool extract_even_num(int a) {
2     if(a % 2 == 0) return true;
3     else return false;
4 }
5
6 bool criteria_sort(int a, int b) {
7     if(a < b) return true;
8     else return false;
9 }
10
11 START() {
12     int a, i, temp, sum;
13     char b;
14     intset as = {15, 16, 23, 1, 3, 100};
15     intset bs = {};
16     a = 5;
17     for(i = 0 ; i < 10 ; ++i) {
18         read(temp);
19         bs += temp; // Adding an element to bs
20     }
21     intset cs = as + bs; // Union of as and bs
22     write(cs);
23     touch(cs, extract_even_num, criteria_sort) {
24         sum += touch.val; // touch.val contains current element.
25     }
26     write(sum);
27 }
```

감사합니다
