CS370 Compilers Lab 7

Ziad Arafat - Apr 11 2021

In this lab we will re-implement our symbol table to check if variable have been declared and to check for matching operator sides.

How to run

```
make clean
make
./lab7 < lab7_sub.decaf
```

OUTPUT

lab7_sub.decaf

LABEL	OFFSET	SIZE	LEVEL	TYPE	SUBTYPE
print_int	0	0	0	VOID	Extern Method

LABEL	OFFSET	SIZE	LEVEL	TYPE	SUBTYPE
print_string	0	0	0	VOID	Extern Method
print_int	0	0	0	VOID	Extern Method

LABEL	OFFSET	SIZE	LEVEL	TYPE	SUBTYPE
_T4	18	1	2	INT	Scalar
_T3	17	1	2	INT	Scalar
_T2	16	1	2	INT	Scalar
_T1	15	1	2	INT	Scalar
_T0	14	1	2	INT	Scalar
x	4	10	2	INT	Array
y	3	1	1	INT	Scalar
f	0	0	0	INT	Method
b	2	1	1	INT	Scalar
z	201	1	0	INT	Scalar
Y	101	100	0	BOOL	Array
Z	1	100	0	INT	Array
y	0	1	0	INT	Scalar
print_string	0	0	0	VOID	Extern Method
print_int	0	0	0	VOID	Extern Method

LABEL	OFFSET	SIZE	LEVEL	TYPE	SUBTYPE
_T9	18	1	2	INT	Scalar
_T8	17	1	2	INT	Scalar
_T7	16	1	2	INT	Scalar
_T6	15	1	2	INT	Scalar
_T5	14	1	2	INT	Scalar
x	4	10	2	INT	Array
y	3	1	1	INT	Scalar
f	0	0	0	INT	Method
b	2	1	1	INT	Scalar
z	201	1	0	INT	Scalar
Y	101	100	0	BOOL	Array
Z	1	100	0	INT	Array
y	0	1	0	INT	Scalar
print_string	0	0	0	VOID	Extern Method
print_int	0	0	0	VOID	Extern Method

LABEL	OFFSET	SIZE	LEVEL	TYPE	SUBTYPE
y	3	1	1	INT	Scalar
f	0	0	0	INT	Method
b	2	1	1	INT	Scalar
z	201	1	0	INT	Scalar
Y	101	100	0	BOOL	Array
Z	1	100	0	INT	Array
y	0	1	0	INT	Scalar
print_string	0	0	0	VOID	Extern Method
print_int	0	0	0	VOID	Extern Method

```
LABEL       OFFSET SIZE   LEVEL  TYPE      SUBTYPE
_T15        9      1      1      INT       Scalar
_T14        8      1      1      INT       Scalar
_T13        7      1      1      INT       Scalar
_T12        6      1      1      INT       Scalar
_T11        5      1      1      INT       Scalar
_T10        4      1      1      INT       Scalar
main        0      0      0      INT       Method
arg1        3      1      1      INT       Scalar
arg2        2      1      1      INT       Scalar
f           0      19     0      INT       Method
z           201    1      0      INT       Scalar
Y           101    100    0      BOOL      Array
Z           1      100    0      INT       Array
y           0      1      0      INT       Scalar
print_string0      0      0      VOID      Extern Method
print_int   0      0      0      VOID      Extern Method

Parsing completed
LABEL       OFFSET SIZE   LEVEL  TYPE      SUBTYPE
main        0      10     0      INT       Method
f           0      19     0      INT       Method
z           201    1      0      INT       Scalar
Y           101    100    0      BOOL      Array
Z           1      100    0      INT       Array
y           0      1      0      INT       Scalar
print_string0      0      0      VOID      Extern Method
print_int   0      0      0      VOID      Extern Method

 EXTERN FUNC print_int
  EXTERN Type  INT
END EXTERN with Type:
  VOID  EXTERN FUNC print_string
  EXTERN Type  STRING
END EXTERN with Type:
  VOID  Package : foo
     Variable  y with type  INT  = 7
  Variable  Z[100]  with type  INT
  Variable  Y[100]  with type  BOOLEAN
  Variable  z with type  INT  = 10
  METHOD FUNCTION  'f' with type  INT
    (
          Method ARGb INT
    )
    BLOCK STATEMENT
     Variable  y with type  INT
     BLOCK STATEMENT
      Variable  x[10]  with type  INT
      ASSIGNMENT STATEMENT
       Variable Left x
```

```
      [
      EXPR  -
       EXPR  +
        CONSTANT INTEGER  2
        CONSTANT INTEGER  3
       CONSTANT INTEGER  5
      ]
     EXPR  +
      Variable Right b
       METHOD CALL name: f
      (
      METHOD ARG
       EXPR  +
        CONSTANT INTEGER  5
        EXPR  *
         Variable Right x
           [
           CONSTANT INTEGER  2
           ]
         Variable Right b
      )
   BLOCK STATEMENT
    Variable  x[10]  with type  INT
    ASSIGNMENT STATEMENT
     Variable Left x
       [
       EXPR  -
        EXPR  +
         CONSTANT INTEGER  2
         CONSTANT INTEGER  3
        CONSTANT INTEGER  5
       ]
     EXPR  +
      Variable Right b
       METHOD CALL name: f
      (
      METHOD ARG
       EXPR  +
        CONSTANT INTEGER  5
        EXPR  *
         Variable Right x
           [
           CONSTANT INTEGER  2
           ]
         Variable Right b
      )
METHOD FUNCTION  'main' with type  INT
  (
       Method ARGarg1 INT
       Method ARGarg2 INT
  )
```

```
    BLOCK STATEMENT
      METHOD CALL name: print_int
     (
     METHOD ARG
       METHOD CALL name: f
      (
      METHOD ARG
       EXPR  +
        EXPR  -
         EXPR  +
          Variable Right arg1
          Variable Right arg2
         Variable Right z
        Variable Right y
      )
     )
      METHOD CALL name: print_int
     (
     METHOD ARG
       METHOD CALL name: f
      (
      METHOD ARG
       EXPR  +
        EXPR  -
         EXPR  +
          Variable Right arg1
          Variable Right arg2
         Variable Right z
        Variable Right y
      )
     )
      METHOD CALL name: print_string
     (
     METHOD ARG
      CONSTANT STRING  "hello world\n"
     )

Finished printing AST
```