

פונקציות ופרוצדורות בשפת MINY

כללי

פונקציה – רוטינה שמחזירה ערך.

פרוצדורה – רוטינה שלא מחזירה ערך.

פונקציות ופרוצדורות בשפת MINY, מוגדרות בתוך התוכנית הראשית, או בתוך פונקציות אחרות לאחר שלב ההגדרות ולפני הבלוק של הקטע הביצועי בתוכנית. לפיכך קיימת אפשרות לקינון פונקציות ופרוצדורות בשפה. טווח ההכרה של הפונקציות, זהה לטווח ההכרה של המשתנים בתוכנית, כלומר ניתן להפעיל אותם מתוך התוכנית שהגדירה אותם, מתוך עצמם או מתוך תת-פונקציות שלהם או של התוכנית שהגדירה אותם.

מיקום ההגדרה

לדוגמא :

```
PROGRAM example
  a FIXED;
  b FIXED;

  PROCEDURE sort ( ..... )
  {
    .
    .
  }
{
  .
  .
}
```

הגדרת פרוצדורה

פרמטר by reference	פרמטרים by Value	שם	מילה שמורה
↓	↓	↓	↓
PROCEDURE proc (a FIXED, b FIXED, c FIXED IDENTICAL)			
d FIXED;			
{			
}			

הגדרת פונקציה

ערר מוחזר	פרמטר by reference	פרמטר by Value	שם	מילה שמורה
↓	↓	↓	↓	↓
FIXED	IDENTICAL	FIXED	func	FUNCTION
:	(a	FIXED,	c
			FIXED	IDENTICAL)
			d	FIXED;
			{	
			func = d ;	
			}	

בסוף הגדרת הפונקציה יש להוסיף " : " ואחריהם את הטיפוס של הערך המוחזר מהפונקציה.

החזרת הערך מתבצעת בסוף הפונקציה כאשר מבצעים השמה של הערך המוחזר לתוך שם הפונקציה, כפי שמופיע בדוגמא. למעשה כאילו מבצעים השמה למשתנה ששמו זהה לשם הפונקציה ואין צורך להגדירו.

פרמטרים

קיימות שתי צורות להעברת פרמטרים :

1. by Value - העתקת הערך של המשתנה למחסנית (כלומר ערכו של הפרמטר לא משתנה ביציאה מהפונקציה, גם אם שינו אותו בתוך הפונקציה).
הגדרת פרמטר כזה : טיפוס הפרמטר שם הפרמטר.

דוגמא : FIXED param

2. by Reference - העתקת הכתובת של המשתנה למחסנית (כלומר שינוי הפרמטר בתוך הפונקציה משנה את ערכו גם ביציאה מהפונקציה).

כדי לזהות פרמטר שמועבר בצורה זו משתמשים במילה השמורה IDENTICAL – בסוף ההצהרה על הפרמטר.

הגדרת פרמטר כזה : IDENTICAL טיפוס הפרמטר שם הפרמטר.

דוגמא : IDENTICAL FIXED param

קריאה לפונקציה/פרוצדורה

הפעלת פונקציות ופרוצדורות זהה לשפות אחרות. הקריאה מתבצעת ע"י כתיבת שם הרוטינה ואחריה הפרמטרים המועברים.

קריאה לפרוצדורה היא פקודה בשפה , לדוגמא :

swap (a, b)

בעוד שפונקציה יכולה להחזיר ערך ולכן ניתן להציב את הערך המוחזר לתוך משתנה ואז הקריאה לפונקציה תהיה כחלק מביטוי השמה בתוכנית, לדוגמא:
 ReturnValue = func (a,b);

מבנה עץ התוכנית

הגדרת פרוצדורה

type = PROC_DECL
 string = Procedure name

אם אין פרמטרים :

children = 1

child[0]
 |
 type = begin
 (procedure's block :
 declarations
 sub-proc/func declarations
 code).

אם יש פרמטרים :

children = 2
 child[1]
 type = begin
 (procedure's block :
 declarations
 sub-proc/func declarations
 code).

child[0]
 type = parameter_list
 (list of parameters
 detailed later)

הגדרת פונקציה

type = FUNC_DECL
 string = Function name

אם אין פרמטרים :

children = 2
 child[1]
 |
 type = BEGIN
 (procedure's block :
 declarations
 sub-proc/func declarations
 code).

child[0]
 |
 type of the return
 value type.

אם יש פרמטרים :

children = 3

child[0]	child[1]	child[2]	
type = parameter_list (list of parameters detailed later)	type of the return value type.	type = begin (procedure's block : declarations sub-proc/func declarations code).	

הגדרת פרמטרים

type = parameter_list
 children = 2
 (parameter_list is defined as any list -
 contains the list of parameters of current function/procedure.
 Each parameter in the list can be byValue, or byRef as defined below)

byValue - פרמטר

type = by_value
 string = parameter's name.
 children = 1
 child[0]
 |
 type = any type. (The parameter type).

byReference - פרמטר

type = by_Ref
 string = parameter's name.
 children = 1
 child[0]
 |
 type = any type. (The parameter type).

קריאה לפונקציה/פרוצדורה

type = PROC_CALL

string = procedure/function name.

אם אין פרמטרים :

children = 0

אם יש פרמטרים :

children = 1

child[0] (parameter list

|

type = ExprList

(ExprList is defined as any list (StatList, CaseList etc.) -
contains the list of parameters of current function/procedure.
Each parameter in the list is an expr (var , expression etc).