

(ストアド) パッケージの作成方法と、(PL/SQL) プロシージャ間で共通に使う変数とカーソル

(ストアド) パッケージは、仕様部 (呼出しインターフェイス定義) と本体部 (プログラム・コード) の2つに分かれている。
この2つを別々に作成する

仕様部の作成と実装部の作成

らくらく Oracle とってもやさしい PL/SQL プログラミング塾

ストアドパッケージ (全体)

仕様部 (インターフェイス定義)

```
CREATE OR REPLACE PACKAGE          パッケージ名 AS ⇨  
  FUNCTION   ファンクション名 ( 引数 ) RETURN 戻り値の型 ;  
  PROCEDURE プロシージャ名 ( 引数 ) ;  
  共通変数名 データ型 ;  
END パッケージ名 ;
```

説明は、下部ページ

本体部 (プログラム・コード)

```
CREATE OR REPLACE PACKAGE BODY パッケージ名 AS ⇨  
  FUNCTION   ファンクション名 ( 引数 ) RETURN データ型 AS ⇨  
  BEGIN  
    プログラミング部  
  
    RETURN 戻り値 ;  
  END ファンクション名 ;  
  
  PROCEDURE プロシージャ名 ( 引数 ) AS ⇨  
  BEGIN  
    プログラミング部  
  
    RETURN ;  
  END プロシージャ名 ;  
END パッケージ名 ;
```

/ 登録操作の実行のために、行頭に「/」を入れる

(作成例)

```
/* 仕様部 */
CREATE OR REPLACE PACKAGE      package_c AS
  FUNCTION func_b ( i1 IN NUMBER , i2 IN NUMBER )
    RETURN NUMBER ;
  PROCEDURE proc_a ( i1 IN NUMBER , i2 IN NUMBER,
    o1 OUT NUMBER ) ;

  wk_num NUMBER;

END package_c ;

/* 本体部 */
CREATE OR REPLACE PACKAGE BODY package_c AS
  FUNCTION func_b ( i1 IN NUMBER , i2 IN NUMBER )
    RETURN NUMBER AS
  BEGIN
    /* プログラミング部 */
    wk_num := wk_num + 5 ;
    RETURN i1 + i2 + wk_num ;
  END func_b ;

  PROCEDURE proc_a ( i1 IN NUMBER , i2 IN NUMBER, o1
    OUT NUMBER ) AS
  BEGIN
    /* プログラミング部 */
    wk_num := wk_num + 3 ;
    o1 := i1 + i2 + wk_num ;
    RETURN ;
  END proc_a ;

END package_c ;
```

呼出し例)

DECLARE

func_modori NUMBER;

procedure_outparameter NUMBER;

BEGIN

package_C.wk_num := 0; -- 共通変数への値セット

func_modori := package_c.func_b(5, 7);

DBMS_OUTPUT.PUT_LINE('ファンクション戻り値: ' || func_modori);

package_c.proc_a(8, 6, procedure_outparameter);

DBMS_OUTPUT.PUT_LINE('プロシージャ出力引数: ' ||

procedure_outparameter);

END;

/

---- 実行結果 ----

ファンクション戻り値: 17

プロシージャ出力引数: 22

(作成例)

```
/* 仕様部 */
CREATE OR REPLACE PACKAGE      package_c IS
  FUNCTION func_b ( i1 IN NUMBER , i2 IN NUMBER )
    RETURN NUMBER ;
  PROCEDURE proc_a ( i1 IN NUMBER , i2 IN NUMBER,
    o1 OUT NUMBER ) ;

END package_c ;

/* 実装部 */
CREATE OR REPLACE PACKAGE BODY package_c IS
  FUNCTION func_b ( i1 IN NUMBER , i2 IN NUMBER )
    RETURN NUMBER IS
      変数名 1 データ型 ; ⚠      -- ローカル変数定義
      変数名 2 データ型 ;

BEGIN
  /* プログラミング部 */

  RETURN i1 + i2 ;
END ;

PROCEDURE proc_a ( i1 IN NUMBER , i2 IN NUMBER,
  o1 OUT NUMBER ) IS
  変数名 1 データ型 ; ⚠      -- ローカル変数定義
  変数名 2 データ型 ;

BEGIN
  /* プログラミング部 */
  o1 = i1 + i2
  RETURN ;
END ;
END package_c ;
/
```

呼出し例)

```
DECLARE
  wk_out NUMBER;
BEGIN
  package_C.wk_num := 0 ;
  package_c.proc_a( 5, 5, wk_out );
  DBMS_OUTPUT.PUT_LINE( 'proc_a 答え' || wk_out ) ;
  wk_out := package_c.func_b( 3, 3 );
  DBMS_OUTPUT.PUT_LINE( 'func_b 答え' || wk_out ) ;
END;
/
```

---- 実行結果 ----

```
proc_a 答え 10
func_b 答え 6
```

パッケージ内のプロシージャ間で、共通で使う変数と共通で使うカーソル（共通変数と共通カーソル）

CREATE PACKAGE 文（パッケージ仕様部、本体部のどちらでもよい）直後に、変数を定義すると、すべてのプロシージャで変数のインスタンスを共有された共通変数となる

※ 有効期間

この変数やカーソルが初期化される時期は、セッションが切断される時です。それまでは、定義したパッケージ内のプロシージャ間で共通の値やレコード位置（カーソル変数）が有効です。

ただし、SQL Developer では、テスト実行のたびにセッションが切断されるので使えません。

```
/* 仕様部 */
```

```
CREATE OR REPLACE PACKAGE      package_c AS
  wk_common_hensuu NUMBER := 0 ; -- 共通変数の定義
  FUNCTION func_b ( i1 IN NUMBER , i2 IN NUMBER )
    RETURN NUMBER ;
  PROCEDURE proc_a ( i1 IN NUMBER , i2 IN NUMBER,
    o1 OUT NUMBER ) ;

END package_c ;
```

どちらでも良いが、
どちらか一方

```
/* 本体部 */
```

```
CREATE OR REPLACE PACKAGE BODY package_c AS
  wk_common_hensuu NUMBER := 0 ; -- 共通変数の定義
  FUNCTION func_b ( i1 IN NUMBER , i2 IN NUMBER )
    RETURN NUMBER AS
  BEGIN
    .
    .
    .
  END;
```

共通変数の値（初期値）セットは、外部からの無名 PL/SQL ブロックから行う

```
DECLARE
```

```
BEGIN
```

```
  package_C.wk_common_hensuu := 0 ;
```

```
END;
```

```
/
```

パッケージ内のストアドプログラムだけが利用できるプライベート・プロシージャの定義

仕様部がない（パッケージ・ボディーのみの）プロシージャやファンクションは、パッケージ内のプロシージャ内からだけ呼出せ、それ以外の外部のストアドプログラムや無名パッケージからは呼び出せません

```
/* 仕様部 */
CREATE OR REPLACE PACKAGE package_c AS
FUNCTION func_b ( i1 IN NUMBER , i2 IN
NUMBER ) RETURN NUMBER ;
PROCEDURE proc_a ( i1 IN NUMBER , i2 IN
NUMBER, o1 OUT NUMBER ) ;

END package_c ;
```

```
/* 本体部 */
CREATE OR REPLACE PACKAGE BODY package_c AS
→FUNCTION func_b ( i1 IN NUMBER , i2 IN
NUMBER ) RETURN NUMBER AS
BEGIN
.
.
.
END ;

PROCEDURE proc_a ( i1 IN NUMBER , i2 IN
NUMBER, o1 OUT NUMBER ) AS

b1 NUMBER ;
BEGIN
.
.
b1 := func_b ( 1 , 2 ) ;
.
END ;
```

