ADA PROGRAMMING LANGUAGE TRAINING

Section 02
Packages 01

Pedro Fernando Flores Palmeros

1 INTRODUCTION

In the prevvious example the Week package only had declarations and no body. In a package specification, you cannot declare bodies. Those have to be in the package body.

The next code is `operations.ads` this code is just the manifold or the skeleton of the package, only the function signature is declared but not implemented. In this example there are two main functions the first one is `Increment_By` that has two parameter, the fist one is `I` and it is needed, and the second parameter is `Incr` this parameter is optional, if it is not provided by the user, then it will use its default value as `0`. The second function is `Get_Increment_Value` this function does not have parameters and returns an Integer.

Code 1: operations.ads

```ada
package Operations is

   -- Declaration
   function Increment_By
      (I    : Integer;
       Incr : Integer := 0) return Integer;

   function Get_Increment_Value return Integer;

end Operations;
```

The next code is `operations.adb` in this code is the implementation of the functions declared in code 1. Coincidentally, introducing a body allows us to put the `Last_Increment` variable in the body and make then inaccessible to the user of the `Operations` package, providing the fist form of encapuslation. This works because entities declared in the body are *only* visible in the body.

Code 2: operations.adb

```ada
package body Operations is

   Last_Increment : Integer := 1;

   function Increment_By
      ( I    : Integer;
        Incr : Integer := 0) return Integer is
   begin
      if Incr /= 0 then
         Last_Increment := Incr;
      end if;

      return I + Last_Increment;
   end Increment_By;

   function Get_Increment_Value return Integer is
   begin
      return Last_Increment;
   end Get_Increment_Value;

end Operations;
```

In the `main` the `Operations` package is used and a procedure is declarad and implemented only to display de values. Note that in line 22 the procedure `Increment_By` is used with two arguments and in line 17 is invoked only with one value.

Code 3: operations.adb

```ada
1  with Ada.Text_IO; use Ada.Text_IO;
2  with Operations;
3
4  procedure Main is
5     use Operations;
6     I : Integer := 0;
7     R : Integer;
8
9     procedure Display_Udate_Values is
10       Incr : constant Integer := Get_Increment_Value;
11    begin
12       Put_Line(Integer'Image(I) & " incremented by "
         & Integer'Image(Incr) & " is " & Integer'Image(R));
      I := R;
   end Display_Udate_Values;
begin
   R := Increment_By(I);
   Display_Udate_Values;
   R := Increment_By(I);
   Display_Udate_Values;

   R := Increment_By(I,5);
   Display_Udate_Values;
   R := Increment_By(I);
   Display_Udate_Values;

   R := Increment_By(I, 10);
   Display_Udate_Values;
   R := Increment_By(I);
   Display_Udate_Values;
end Main;
```