ADA PROGRAMMING LANGUAGE TRAINING

Program 04
Check Direction

Pedro Fernando Flores Palmeros

1 INTRODUCTION

The main syntaxis of the `if` control structure is the next

```
if <condition> then
   statement 01;
   ...
   statement n;
elsif <condition 1> then
   statement 01;
   ...
   statement m;
elsif <condition q> then
   statement 01;
   ...
   statement p;
else
   statement 01;
   ...
   statement k;
end if;
```

in this case `<condition>` is the condition to be tested it does not require the <> symbols.

2 CODE

```
 1  with Ada.Text_IO; use Ada.Text_IO;
 2  with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
 3
 4  procedure Check_Direction is
 5     N : Integer;
 6  begin
 7     Put("enter an integer value");
 8     Get(N);
 9     Put(N);
10
11     if N=0 or N=360 then
12        Put_Line(" is due north");
13     elsif N in 1..89 then
14        Put_Line(" is in the northeast quadrant");
15     elsif N in 90 then
16        Put_Line(" is due east");
17     elsif N in 91 ..179 then
18        Put_Line(" is in the southeast quadrant" );
19     elsif N in 180 then
20        Put_Line(" is due south" );
21     elsif N in 181..269 then
22        Put_Line(" is in the southwest quadrant" );
23     elsif N = 270 then
24        Put_Line(" is due west");
25     elsif N in 271 .. 359 then
26        Put_Line(" is in the northwest quadrant" );
27     else
28        Put_Line(" is not in the range 0 ... 360" );
29     end if;
30  end Check_Direction;
```

3 Main parts

Observe that in this case in line 1 and 2 the packages that are going to be used are declared, and that is why in the lines 8, 15, 16, 17 the commands are without `Ada.Text_IO`. For further information in the first program is a detailed explanation of it.

3.1 Get(N)

This function is part of the package `Ada.Text_IO`, and it is useful for reading Integers from the keyboard, observe that a variable of Integer type has to be sent in the function argument.

3.2 Variable declaration

The variables and subsfunctions that are going to be used and implemented in the main procedure has to be defined in the lines between the procedure declaration and the begin of the implementation of the procedure. In this particular case between line 4 and 5.

3.3 IF - ELSIF - ELSE DIRECTIVES

Sometimes due the nature of the algorithm or problem to be solved it is required to use multiple conditions to evaluate a certain event, in this case the directive to be used is the `elsif`.

With this directive many conditions can be nested. In which if the first condition is not filled the second condition is tested and if the second is not filled, then it goes through the rest fo the conditions, if no condition has been accomplished then the conditions attached to the else are executed.