# Value Parameters

Earlier in the chapter we made a procedure called drawtriangle. We said it had some weaknesses. It would always draw the triangle in the same place and in the same colour. By using *value parameters* we can make the procedure a lot more flexible. We can pass information to the procedure about where to draw the triangle and what colour to use. As a first example let's modify our original drawtriangle to allow us to specify the colour.

|  |  |
| --- | --- |
| |  | | --- | | **procedure** drawtriangle (triangleColour : **int**)  drawline(100, 100, 150, 150, triangleColour)  drawline(150, 150, 100, 200, triangleColour)  drawline(100, 200, 100, 100, triangleColour)  **end** drawtriangle  *% main program*  **var** s : **string**  drawtriangle(blue)  **put** "press enter to change the triangle to green"  **get** s:\*  drawtriangle(green) | |

Notice how we call the procedure now. We use the name of the procedure followed by brackets with the colour inside. This *argument* gets copied to the parameter in the procedure (the variable called triangleColour). So the first time the procedure is called triangleColour is blue. The second time it has the value green.

Now let's modify the procedure again so we can specify the location of the bottom vertex of the triangle. In the original procedure that vertex was at coordinates (100, 100).

|  |  |
| --- | --- |
| |  | | --- | | **procedure** drawtriangle (x : **int**, y : **int**, triangleColour : **int**)  drawline(x, y, x + 50, y + 50, triangleColour)  drawline(x + 50, y + 50, x, y + 100, triangleColour)  drawline(x, y + 100, x, y, triangleColour)  **end** drawtriangle  *% main program*  **var** s : **string**  drawtriangle(100, 100, blue)  **put** "press enter to change the triangle to green and a new location"  **get** s:\*  drawtriangle(300, 200, green) | |

Copy this program and try running it. Try modifying the colour and location in the procedure calls.

You have to be careful that the type of the arguments and the type of the parameters match. The following procedure calls are all incorrect:

|  |  |
| --- | --- |
| |  | | --- | | drawtriangle(40, 300, "green") *% third argument should be a colour, ie. an int*  *% not a string*  drawtriangle(green, 40, 300) *% tries to draw a triangle with lower vertex of (2, 40)*  *% since green has a value of 2 and a colour of 300 which*  *% is not allowed (maxcolor is 256)*  drawtriangle(40, 300) *% wrong number of arguments. Need to specify the colour.*  drawtriangle(40.5, 300, green) *% first argument should be an int* | |

In the first line of a procedure definition we list the parameters. We can use the same shortcuts that we use when we declared variables. In the drawTriangle example all the parameters are of type int. So we could change the first line of the procedure to: drawTriangle (x, y, triangleColour: **int**).

Here is an example where the types of the parameters are different:

|  |  |
| --- | --- |
| |  | | --- | | *%\*\*\*\*\*\* formatReal \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  *% Purpose: allows you to format a real number by specifying the \**  *% number of decimal places you want and whether you \**  *% want to go to a new line after printing \**  *% Parameters: 1) number - the number to be formatted \**  *% 2) decimalPlaces - how many decimal places required \**  *% 3) newLine - false if you want to stay on the current \**  *% line \**  *%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  **procedure** formatReal (number : **real**, decimalPlaces : **int**, newLine : **boolean**)  **if** newLine **then**  **put** number:1:decimalPlaces  **else**  **put** number:1:decimalPlaces..  **end** **if**  **end** formatReal  *% main program*  **var** i : **int** := 5  **var** x : **real** := 123.456  **var** b : **boolean** := **true**  formatReal(x, i, b)  formatReal(12.4, 3, **false**)  formatReal(i, 2, b)  formatReal(543.2109, 3, **true**) | |

Here is the output from this program:

|  |
| --- |
| 123.45600  12.4005.00  543.211 |

Note that the first argument can be an int or a real (since a real variable can hold either type). The second argument must be an int and the third argument must be a boolean. Also note the comments at the start of the procedure. It is a good idea to do this for all your procedures. By describing what the procedure does and what its parameters are it makes your procedure easier to use.