Appendix B

Custom subroutines

Mathematical functions

The following table contains examples and explanations of some mathematical functions in Delphi:

<table>
<tr><th colspan="2">Delphi</th><th>Comment</th></tr>
<tr><td colspan="2">iVal := Round(rX);</td><td rowspan="5">Rounds a real value and returns an integer value rounded to the nearest whole number.
If the number is halfway between two integers, it will always be rounded to the nearest even number. E.g.
Round(7.5) = 8 and also
Round(8.5) = 8</td></tr>
<tr><td>Round(8.8)</td><td>9</td></tr>
<tr><td>Round(8.3)</td><td>8</td></tr>
<tr><td>Round(7.3)</td><td>7</td></tr>
<tr><td>Round(6.5)</td><td>6</td></tr>
<tr><td colspan="2">iVal := Trunc(rX);</td><td rowspan="3">Truncates a real number to an integer. It 'chops' off the decimal part and returns only the integer part.</td></tr>
<tr><td>Trunc(8.5)</td><td>8</td></tr>
<tr><td>Trunc(8.3)</td><td>8</td></tr>
<tr><td colspan="2">rVal := Frac(rX);</td><td rowspan="3">Returns only the decimal part of a real number.</td></tr>
<tr><td>Frac(74.89)</td><td>0.89</td></tr>
<tr><td>Frac(0.3728)</td><td>0.3728</td></tr>
<tr><td colspan="2">iAnswer := Ceil(rX) ;</td><td rowspan="4">Returns the largest integer value nearest to the number.
This function is stored in the Unit Math.</td></tr>
<tr><td>Ceil(2.1)</td><td>3</td></tr>
<tr><td>Ceil(2.9)</td><td>3</td></tr>
<tr><td>Ceil(-3.1)</td><td>-3</td></tr>
</table>

<table>
<tr><td colspan="2">iAnswer := Floor(rX) ;</td><td rowspan="4">Returns the smallest integer value nearest to the number.
This function is stored in the Unit Math.</td></tr>
<tr><td>Floor(2.1)</td><td>2</td></tr>
<tr><td>Floor (2.9)</td><td>2</td></tr>
<tr><td>Floor (-3.1)</td><td>-4</td></tr>
<tr><td colspan="2">rVal := Sqrt(rX);</td><td>Returns the square root of a number. The result is always of type real.
Note: The square root of a negative number can not be calculated.</td></tr>
<tr><td colspan="2">rVal := Sqr(rX);
iVal := Sqr(iY);</td><td>Returns the square of a number – the result is of the same type as the number.</td></tr>
<tr><td colspan="2">rVal := Abs(rX);
iVal := Abs(iY);</td><td>Returns the absolute value of a number – the result is of the same type as the number.</td></tr>
<tr><td colspan="2">rC:= 2 * PI * iR;</td><td>PI returns a value for the mathematical constant PI.</td></tr>
<tr><td colspan="2">rAnswer := Power(iBase, iExponent);</td><td>Raises the first number to the power of the second number. All variables must be compatible with the data type extended.
This function is stored in the Unit Math.</td></tr>
<tr><td colspan="2">iVal := Random(iX);</td><td>Returns an integer value in the range [0..(iX-1)].</td></tr>
<tr><td colspan="2">rVal := Random;</td><td>Returns a real number >=0 but < 1.</td></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><td colspan="3">Tip
• Random (100) has 100 possible answers (from 0 to 99). To generate a number between 1 and 100 (both included) do the following:
iNo := Random (100) + 1;
• To make the numbers generated by the random function less predictable use the Randomize procedure. Randomize simply scrambles the results of the Random function. You only have to use this procedure once in your program, preferably when the program starts (use the OnActivate event of the form).</td></tr>
</table>

Procedures that do calculations

Inc(iNumber, iAdd); Inc(iNumber); Inc(cChar,2);	Increases the value of iNumber with iAdd Increases the value of iNumber with 1 Change the character to two places after this character in the ASCII table. (Increases the ordinal value of the character by 2)
Dec(iNumber,2); Dec(iNumber); Dec(cChar,2);	Decreases the value of iNumber with 2. Increases the value of iNumber with 1 Change the character to two places before this character in the ASCII table. (Decreases the ordinal value of the character by 2)

String manipulation functions

A string is represented in memory as a sequence of numbered spaces.

```
Var sName : string ;
sName := 'Peter' ;
```

```
sName [1] has the value 'P'
sName[4] has the value 'e'
```

Data in memory

P	e	t	e	r
[1]	[2]	[3]	[4]	[5]

We can do manipulations on strings such as to delete individual characters in a string, change some characters to uppercase, add characters any place in a string, or extract a group of characters from the string. String manipulation functions can also provide us with information regarding strings, for example: How long is a string (how many characters does it consists of)? Where is the space in the string? Is the word 'house' part of the string?

Here are some examples of functions that can be used to provide us with information on strings:

Delphi function	**Comment**
lblOut.Caption:=**Concat**('Hello ',sName); lblOut.Caption := 'Hello ' + sName;	The Concat function can be used instead of the '+' operator.
iVal := **Length**(sName);	Returns the number of characters that a string value consist of as an integer.

<table>
<tr><td>iPos := Pos(sString, sName);</td><td>Returns the position of the first letter of sString in sName.</td></tr>
<tr><td></td><td>Study the following examples, assume that
sName := 'Software Application';
<table>
<tr><th>Example</th><th>Value returned</th></tr>
<tr><td>iPos := Pos('a',sName);</td><td>6</td></tr>
<tr><td>iPos := Pos('war',sName);</td><td>5</td></tr>
<tr><td>iPos := Pos(' ',sName);</td><td>9</td></tr>
<tr><td>iPos := Pos('A',sName);</td><td>10</td></tr>
</table></td></tr>
<tr><td>sWord :=
Copy(sS, iStart, iLength);</td><td>Copies a substring from sS, from position iStart for iLength characters.</td></tr>
<tr><td></td><td>Study the following examples, assume that
sS := 'Software Application';
<table>
<tr><th>Example</th><th>String returned</th></tr>
<tr><td>sWord := Copy(sS,5,4);</td><td>ware</td></tr>
<tr><td>sWord := Copy(sS,2,3);</td><td>oft</td></tr>
</table></td></tr>
<tr><td>sWord := Uppercase(sWord);</td><td>Converts all letters in sWord to uppercase and assigns it to sWord.</td></tr>
<tr><td>sWord[1] := Upcase(sWord[1]);</td><td>The number in brackets refers to the position of a character in a string. The Upcase function is used to convert one character to uppercase. In this example the first character of the string is converted to uppercase.</td></tr>
</table>

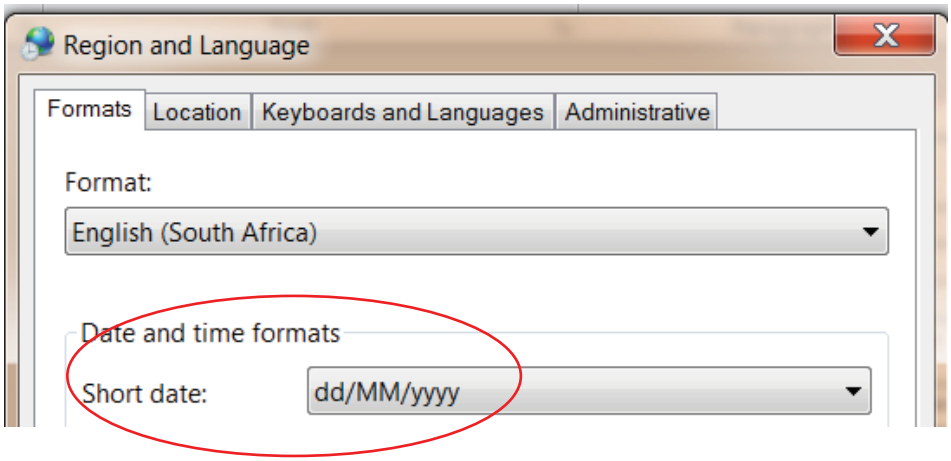
String manipulation procedures

We can also manipulate strings using the procedures *Insert* and *Delete*. These two procedures receive a number of values and then make changes to one of the values.

<table>
<tr><td colspan="3">Delete(sString, iStart, iLength);
Deletes a substring from a string.
sString: The string that should be changed.
iStart: The position of the first character that should be deleted.
iLength: The number of characters to be removed from the string – the starting one included.</td></tr>
<tr><td>Example:</td><td>sS := 'Lets start from scratch';
Delete(sS,12,5);</td><td>Value of sS is now: 'Lets start scratch'</td></tr>
<tr><td colspan="3">Insert(sSub, sString, iPosition);
Inserts a substring into a string in a given position.
sSub: the string that should be inserted.
sString: The string that should be changed.
iPosition: The position where the substring should start in the changed string.</td></tr>
<tr><td>Example:</td><td>sS := 'Bread tastes good';
Insert('Hot ',sS,1);</td><td>'Hot Bread tastes good'</td></tr>
<tr><td colspan="2">Val(sString, iNumber,iCode);</td><td>Converts sString to a number and stores the ‘number’ in variable iNumber. If the variable iCode is 0, the conversion was successfull, if not, the value of iCode shows the position of the character in the string that caused the problem.</td></tr>
<tr><td colspan="2">Str(rNumber,sString:5:2);</td><td>Converts the value stored in rNumber to a string value and stores the string in sString. In this case the number should consist of 5 places including 2 decimals.</td></tr>
<tr><td colspan="3">Consult the Delphi Help for more examples of using Val and Str.</td></tr>
</table>

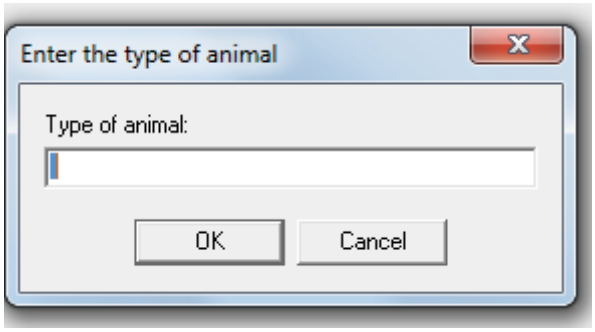
Time and date functions

sTime := **TimeToStr**(Time);	Converts the system time to a string.
Date	Returns the current system date.
pnlDate.Caption := **DateToStr**(Date);	Converts the system date to a string. The format will be determined by the *Short Date* format in the *Region and Language* settings. For example, 23 March 2013 will be displayed as 23/03/2013 if the settings is as follows:



bLeap := IsLeapYear(2000) ;	The *IsLeapYear* function returns true if a given calendar value is a leap year. The year supplied as the argument can be a value between 0 and 9999.

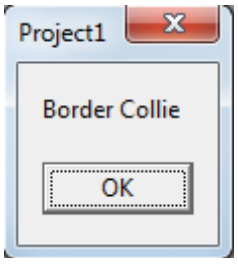
Other functions

sAnimal := **InputBox**('Enter the type of animal','Type of animal:',' ');	A function to input a string that creates a pop up: 
frmScreen.Color := **RGB**(iRed, iGreen, iBlue);	Sets the colour of the screen using a combination of red, green and blue.
bAnswer := **Odd** (iValue) ;	Determines whether the argument is odd or not. Returns Boolean value.

iNum := **ord**('A') ;	The *ord* function is usually used to determine the ordinal value (ASCII code) of a character.
	iNum := **ord**('A') ; //Returns the value 65
	iNum := ord('1') ; //Returns the value 49

cChar := chr(65) ;	The *chr* function can return the character associated with an ASCII code.
	cChar := chr(65) ; //Returns the character 'A'
	cChar := chr(50) ; //Returns the character '2'

Other procedures

ShowMessage(sAnimal);	A procedure to display a message, without using a specific object on the form: 
Randomize;	Scrambles the results of the Random function to make the results less predictable
Beep;	Makes a beep noise
Sleep(200);	Delays the program for 2 seconds (200 milli seconds)