```pascal
unit HUValidations;

{$mode objfpc}{$H+}

//==========================================================================================
//
// Unit : HUValidations.pas
//
// Description :
//
// Called By :
//
// Calls :
//
// Ver. : 1.0.0
//
// Date : 10 Feb 2019
//
//==========================================================================================

interface

uses
  Classes, SysUtils, Dialogs,
  // Application units
  // HULib units
  HUConstants;


//function ValidAlphaCharacter( Key: char) : char;
function ValidNameCharacter( Key: char) : char;
function ValidCallsignCharacter( Key: char) : char;
function ValidEmailCharacter( Key: char) : char;
function ValidDirectoryCharacter( Key: char) : char;
function ValidDigitCharacter( Key: char) : char;

implementation

//==========================================================================================
//          PRIVATE CONSTANTS
//==========================================================================================

//==========================================================================================
//          PUBLIC CONSTANTS
//==========================================================================================

//==========================================================================================
//          PRIVATE VARIABLES
//==========================================================================================

//==========================================================================================
//          PUBLIC VARIABLES
//==========================================================================================

//==========================================================================================
//          PRIVATE ROUTINES
//==========================================================================================

//==========================================================================================
//          PUBLIC ROUTINES
```

```
//========================================================================================
{function ValidAlphaNumericCharacter( Key: char) : char;
begin
    // Returns only Valid Alphabetic Characters. Non-valid characters are converted
    // into Null (#0) characters.
    //Valid Alpha C haracters are:
    // <BS>
    // <SP>
    // [A..Z]
    // [a..z]
    // [0..9]
    Result := Key;
    case Key of
      keyBS : Exit; // <BS>
      keySpace : Exit; // <SP>
      key0..key9 : Exit; // [0..9]
      keyA..keyZ : Exit; // [A..Z]
      key_a..key_z : Exit; // [a..z]
    else
      Result := keyNull;
    end;// case Key of
end;// function ValidAlphaNumericCharacter(var Key: char);}

//========================================================================================
function ValidCallsignCharacter( Key: char) : char;
begin
    // Returns only Valid Callsign Characters. Non-valid characters are converted
    // into Null (#0) characters.
    //Valid Alpha C haracters are:
    // <BS>
    // </>
    // [0..9]
    // [A..Z]
    // [a..z] Converted to Uppercase
    Result := Key;
    case Key of
      K_BS : Exit; // <BS>
      K_ForwardSlash : Exit;  // </>
      K_0..K_9 : Exit; // [0..9]
      uK_A..uK_Z : Exit; // [A..Z]
      lK_a..lK_z : begin
                    Result := UpCase(Key);
                    Exit; // [a..z]
                  end;
    else
      Result := K_NULL;
    end;// case Key of

end;// function ValidCallsignCharacter(var Key: char);

//========================================================================================
function ValidDigitCharacter( Key: char) : char;
begin
  // Returns only Valid Digits. Non-valid characters are converted
  // into Null (#0) characters.
  //Valid Digit Characters are:
  // <BS>
  // <DEL>
  // [0..9]

  Result := Key;
  case Key of
    K_BS : Exit; // <BS>
    K_0..K_9 : Exit; // [0..9]
  else
    Result := K_NULL;
  end;// case Key of

end;// function ValidDigitCharacter(var Key: char);

//=======================================================================================
{Function ValidFloatCharacter( Key: char) : char;
begin
  // Returns only Valid Digits and the Decimal Point. Non-valid characters are converted
  // into Null (keyNull) characters.
  // Valid Digit Characters are:
  // <BS>
  // [.]
  // <DEL>
  // [0..9]
  Result := Key;
  case Key of
    keyBS : Exit; // <BS>
    keyDecimalPoint : Exit; // <.>
    keyDEL : Exit; // <DEL>
    key0..key9 : Exit; // [0..9]
  else
    Result := keyNull;
  end;// case Key of
end;// Function ValidFloatCharacter( Key: char)}

//========================================================================================
{function ValidFilenameCharacter(Key: char) : char;
begin
    // Returns only Valid Filename Characters. Non-valid characters are converted
    // into Null (#0) characters.
    //Valid Alpha C haracters are:
    // <BS>
    // <SP>
    // [A..Z]
    // [a..z]
    // [0..9]
    // <_>
    Result := Key;
    case Key of
      keyBS : Exit; // <BS>
      keySpace : Exit; // <SP>
      key0..key9 : Exit; // [0..9]
      keyA..keyZ : Exit; // [A..Z]
      keyUScore : Exit; // <_>
      key_a..key_z : Exit; // [a..z]
    else
      Result := keyNull;
    end;// case Key of
end;// function ValidFilenameCharacter }

//========================================================================================
function ValidDirectoryCharacter (Key: char) : char;
begin

```

```
    // Returns only Valid Directory Characters. Non-valid characters are converted
    // into Null (#0) characters.
    //Valid Alpha Characters are:
    // <BS>
    // <SP>
    // [A..Z]
    // [a..z]
    // [0..9]
    // <_>
    Result := Key;
    case Key of
      K_BS : Exit; // <BS>
      K_SP : Exit; // <SP>
      K_0..K_9 : Exit; // [0..9]
      uK_A..uK_Z : Exit; // [A..Z]
      k_UnderScore : Exit; // <_>
      lK_a..lK_z : Exit; // [a..z]
    else
      Result := K_NULL;
    end;// case Key of

end;// function ValidDirectoryCharacter

//===========================================================================
function ValidNameCharacter (Key: char) : char;
begin

    // Returns only Valid Name Characters. Non-valid characters are converted
    // into Null (#0) characters.
    //Valid Name Characters are:
    // <BS>
    // <SP>
    // [A..Z]
    // [a..z]
    // <->
    Result := Key;
    case Key of
      K_BS : Exit; // <BS>
      K_SP : Exit; // <SP>
      uK_A..uK_Z : Exit; // [A..Z]
      lK_a..lK_z : Exit; // [a..z]
      K_UnderScore : Exit; // <_>
    else
      Result := K_NULL;
    end;// case Key of

end;// function ValidName Characters

//===========================================================================
function ValidEmailCharacter (Key: char) : char;
begin

    // Returns only Valid Name Characters. Non-valid characters are converted
    // into Null (#0) characters.
    //Valid Name Characters are:
    // <BS>
    // <SP>
    // [A..Z]
    // [a..z]
    // <->
    Result := Key;
    case Key of
      K_BS : Exit; // <BS>
      K_SP : Exit; // <SP>
      uK_A..uK_Z : Exit; // [A..Z]
      lK_a..lK_z : Exit; // [a..z]
      K_0..K_9 : Exit; // [0..9]
      K_ExclamationMark : Exit; // !
      K_HashMark : Exit; // #
      K_DollarSign : Exit; // &
      K_PercentSign : Exit; // %
      K_Ampersand : Exit; // &
      K_SingleQuote : Exit; // '
      K_Asterix : Exit; // *
      K_PlusSign : Exit; // +
      K_ForwardSlash : Exit; // /
      K_EqualSign : Exit; // =
      K_QuestionMark : Exit; // ?
      K_Caret : Exit; // ^
      K_UnderScore : Exit; // _
      K_RightSingleQuote : Exit; // '
      K_LeftBracket : Exit; // {
      K_VerticalBar : Exit;
      K_RightBracket : Exit; // }
      K_Tilde : Exit; // ~
      K_Dash : Exit; // -
     else
      Result := K_NULL;
    end;// case Key of

end;// function ValidEmailCharacter

//========================================================================================
//          PROPERTY ROUTINES
//========================================================================================

//========================================================================================
//          MENU ROUTINES
//========================================================================================

//========================================================================================
//          FILE ROUTINES
//========================================================================================

//========================================================================================

end.// unit HUValidations

```