A.13 Exceptions in Input-Output

(1)

The package IO_Exceptions defines the exceptions needed by the predefined input-output packages.

Static Semantics

(2)

The library package IO_Exceptions has the following declaration:

(3)

```
package Ada.IO_Exceptions is
   pragma Pure(IO_Exceptions);
```

(4)

```
   Status_Error : exception;
   Mode_Error   : exception;
   Name_Error   : exception;
   Use_Error    : exception;
   Device_Error : exception;
   End_Error    : exception;
   Data_Error   : exception;
   Layout_Error : exception;
```

(5)

```
end Ada.IO_Exceptions;
```

(6)

If more than one error condition exists, the corresponding exception that appears earliest in the following list is the one that is propagated.

(7)

The exception Status_Error is propagated by an attempt to operate upon a file that is not open, and by an attempt to open a file that is already open.

(8)

The exception Mode_Error is propagated by an attempt to read from, or test for the end of, a file whose current mode is Out_File or Append_File, and also by an attempt to write to a file whose current mode is In_File. In the case of Text_IO, the exception Mode_Error is also propagated by specifying a file whose current mode is Out_File or Append_File in a call of Set_Input, Skip_Line, End_Of_Line, Skip_Page, or End_Of_Page; and by specifying a file whose current mode is In_File in a call of Set_Output, Set_Line_Length, Set_Page_Length, Line_Length, Page_Length, New_Line, or New_Page.

(9)

The exception Name_Error is propagated by a call of Create or Open if the string given for the parameter Name does not allow the identification of an external file. For example, this exception is propagated if the string is improper, or, alternatively, if either none or more than one external file corresponds to the string.

(10)

The exception Use_Error is propagated if an operation is attempted that is not possible for reasons that depend on characteristics of the external file. For example, this exception is propagated by the procedure Create, among other circumstances, if the given mode is Out_File but the form specifies an input only device, if the parameter Form specifies invalid access rights, or if an external file with the given name already exists and overwriting is not allowed.

(11)

The exception Device_Error is propagated if an input-output operation cannot be completed because of a malfunction of the underlying system.

(12)

The exception End_Error is propagated by an attempt to skip (read past) the end of a file.

(13)

The exception Data_Error can be propagated by the procedure Read (or by the Read attribute) if the element read cannot be interpreted as a value of the required subtype. This exception is also propagated by a procedure Get (defined in the package Text_IO) if the input character sequence fails to satisfy the required syntax, or if the value input does not belong to the range of the required subtype.

(14)

The exception Layout_Error is propagated (in text input-output) by Col, Line, or Page if the value returned exceeds Count'Last. The exception Layout_Error is also propagated on output by an attempt to set column or line numbers in excess of specified maximum line or page lengths, respectively (excluding the unbounded cases). It is also propagated by an attempt to Put too many characters to a string.

```
      procedure Get_Line(File : in File_Type; Item : out String; Last :
out Natural);
      procedure Get_Line(Item : out String;   Last : out Natural);
```

(19)

Reads successive characters from the specified input file and assigns them to successive characters of the specified string. Reading stops if the end of the string is met. Reading also stops if the end of the line is met before meeting the end of the string; in this case Skip_Line is (in effect) called with a spacing of 1. The values of characters not assigned are not specified.

(20)

If characters are read, returns in Last the index value such that Item(Last) is the last character assigned (the index of the first character assigned is Item'First). If no characters are read, returns in Last an index value that is one less than Item'First. The exception End_Error is propagated if an attempt is made to skip a file terminator.

(21)