Introduction to Computers and Programming
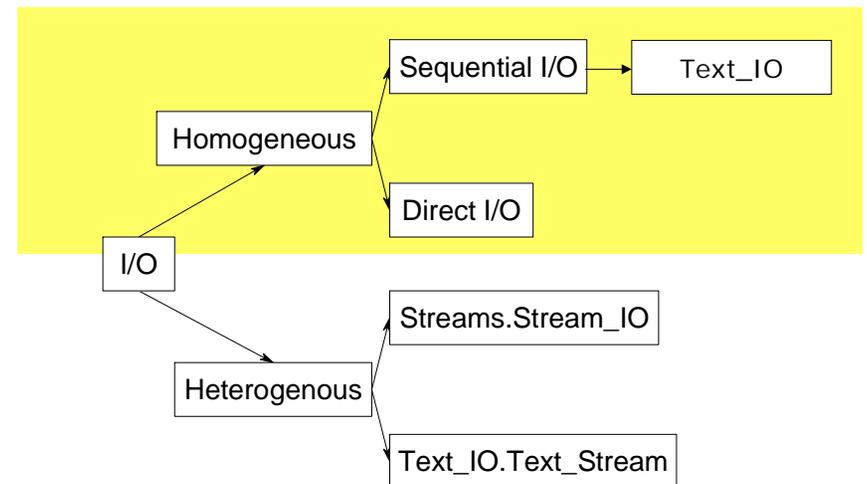
Prof. I. K. Lundqvist

Ada.Strings.Fixed (A.4.3)

- **procedure** Move (Source : **in** String;
  Target : **out** String;
  Drop   : **in** Truncation := Error;
  Justify : **in** Alignment := Left;
  Pad   : **in** Character := Space);

- **function** Insert (Source : **in** String;
  Before   : **in** Positive;
  New_Item : **in** String) **return** String;

- **function** Delete (Source : **in** String;
  From   : **in** Positive;
  Through : **in** Natural) **return** String;

Input/Output

- Related I/O packages

- Text files
  - Formatting input
  - Formatting output files

Input/Output Operations (A.6)

Files (A.7)

- **External files**: Values input from the external environment of the program, or output to the external environment

- **File**
  - Name (string)
  - Form   (string that gives system defined characteristics such as access rights)

Language Defined Types (A.7)

- **File_Type**: associates an object through which the program can access the external file

```
My_File : Ada.Text_IO.File_Type;
```

- For Sequential_IO, Text_IO, Wide_Text_IO and Stream_IO

```
type File_Mode is (In_File, Out_File,
                   Append_File);
```
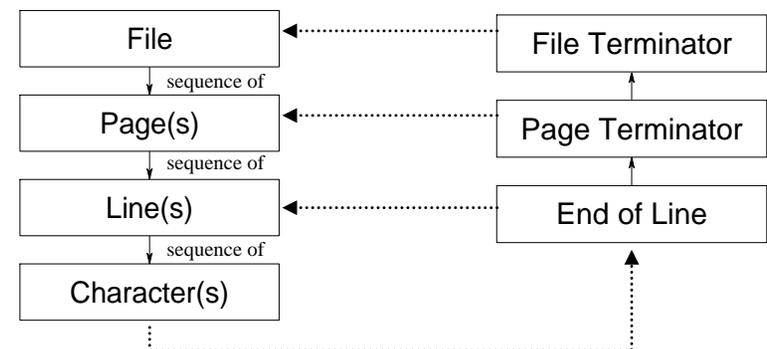
- For Direct_IO

```
type File_Mode is (In_File, Inout_File,
                   Out_File);
```

Sequential vs. Direct (A.8)

- **Sequential Files**
  - Sequential_IO
  - Files viewed as a sequence of values
  - Transfer occurs in order of appearance

- **Direct Files**
  - Direct_IO
  - File viewed as a set of elements occupying consecutive positions in linear order
  - *index*: location of current element(>=1)
  - *current size*: number of elements in file

Logical View of Text Files (A.10)

- Terminators are generated
  - Explicitly by procedures/ functions
  - Implicitly as part of other operations

Ada.Text_IO (A.10.1)

```ada
procedure Create (File : in out File_Type;
                  Mode : in File_Mode := Out_File;
                  Name : in String := "";
                  Form : in String := "");
```

```ada
procedure Open (File : in out File_Type;
                Mode : in File_Mode;
                Name : in String;
                Form : in String := "");
```

```ada
procedure Close  (File : in out File_Type);
procedure Delete (File : in out File_Type);
procedure Reset  (File : in out File_Type;
                  Mode : in File_Mode);
procedure Reset  (File : in out File_Type);
```

`formats.adb`

Text Input/Output

- Ada and I/O
  - Embedded real-time systems ✓
  - Business oriented applications ✗

- Package instantiation
  - Ada.Text_IO.Integer_IO; (generic package)
    - Ada.Integer_Text_IO; (preinstantiated for the type Integer)
  - Ada.Text_IO.Float_IO;
    - Ada.Float_Text_IO;
- Brute force example: Formatted output data

`formats.adb`

Output to a File 1(2)

- Need 2 file names to read and write to a file
  - Internal name (used in our program)
  - External name (used by OS)

- Create -- associate names with each other
  - Create(Internal_Name, Out_File,
          "TEST.TXT");
- Writing to a file
  - Put(Internal_Name, "Some text.");
  - New_Line(Internal_Name, 2);

`fileout1.adb`

Output to a File 2(2)

- Redirecting output
  - Set_Output(Internal_Name);
    Put_Line("Some other text.");
    New_Line(2);
    Put_Line(Standard_Output, "On screen");
    Set_Output(Standard_Output);
- Closing a file
  - Close(Internal_Name);

Files, cont.

- Example: How to open multiple files at once
- Input read from a file
  - `Open(My_File, In_File, "CHARACTS.TXT");`
    - File pointer set to beginning of file
  - `End_Of_File(My_File);`
    - Returns TRUE when next char to be read is EOF character
  - `Reset(My_File);`
    - File pointer moved to beginning of file again
  - `End_Of_Line(My_File)`
    - Returns TRUE when next char in buffer is EOL character

`multfileout.adb`
`charin.adb`

Examples

- String Input and Output
- Read Integers from file
- Read Floating point numbers from file
  - What has to be changed?
- Input from keyboard
  - Internal filename: `Standard_Input`
  - No `End_Of_File` when reading from keyboard

`stringio.adb`
`integerin.adb`