Acting Out ADTs

Daniel M. Berry

© 1996 Daniel M. Berry

Example of Use of ADTs

Situation:

Read a specially structured line of characters from the (only) input file.

And later ...

Dig out the i^{th} character from the line

© 1996 Daniel M. Berry


```
package Specially_Structured_Lines is
        type LINE is private;
        function make_line(s:STRING) return LINE;
        procedure insert_char(l:in out LINE,c:CHAR);
        function get_char(l:LINE,i:INTEGER) return CHAR;
        function length(l:LINE) return INTEGER;
        function index_of_end_of_word(i:INTEGER,l:LINE)
            return INTEGER;
        function no_of_words(l:LINE) return INTEGER;
        function index_of_punct_symbol(i:INTEGER,l:LINE)
            return INTEGER;
        function no_of_punct_symbols(l:LINE)
            return INTEGER;
end;
```

© 1996 Daniel M. Berry


```
with Specially_Structured_Lines, stdio;
package Input_File is
        procedure open;
        function read_line return LINE;
        function is_at_eof return BOOL;
        procedure close;
end Input_File;
```

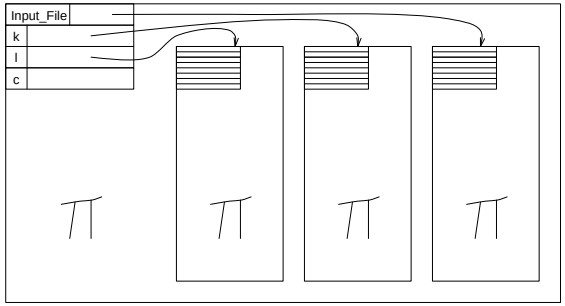
© 1996 Daniel M. Berry

```
with Specially_Structured_Lines, Input_File;
procedure main is
        k,l:LINE
        c: CHAR;
begin
        open;
        while (not is_at_eof) loop
            l:=read_line;
            for i from 1 to length(l) loop
                c:=get_char(l,i)
            end;
        end;
        close;
end main;
```

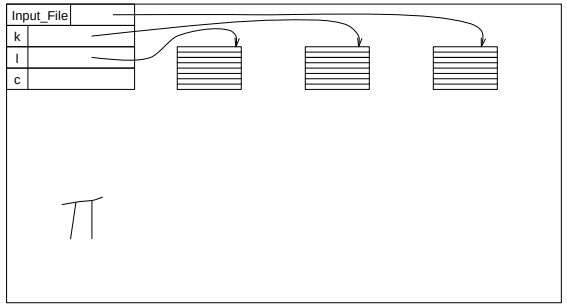

Processes

User's View

Supported by compiler:



Object Code View



```
procedure main is
        k,l:definition[LINE]
        c: CHAR;
begin
        open;
        while (not body[is_at_eof]) loop
            l:=body[read_line];
            -- body[read_line] contains
                -- body[insert_char]
            for i from 1 to body[length(l)] loop
                c:=body[get_char](l,i)
            end;
        end;
        close;
end main;
```