```
package indices is
        type index is non_negative_integer;
        constant index null=0;
        -- = and /= are available and that is all
end indices;

with indices
package line_numbers is
        type linenumber is index;
        -- reexport same ops
end line_numbers;

with indices
package character_pointers is
        type charptr is index;
        -- reexport same ops
end character_pointers;
```

```
with line_numbers character_pointers
package character_list is
        type charlist(size:linenumber);
        procedure create_charlist(cl:in out charlist);
        procedure insert_char_into_charlist(cl:in out charlist,cp:charptr,
                c:character);
        function number_of_chars(cl:charlist) returns index;
        function get_char_from_charlist(cl:charlist,cp:charptr)
                returns character;
end character_list;

with character_list
package indexing_table_for_charlist is
        type indextable(size:charptr);
        function create_indextable(it:in out indextable, cl:ptr charlist);
        procedure insert_pair_into_indextable(it:in out indextable,
                ln:linenumber, cp:charptr);
        function number_of_indices(it:indextable) returns index;
        function get_linenumber_from_indextable(it:indextable,
                i:index) returns linenumber;
        function get_charptr_from_indextable(it:indextable,
                i:index) returns charptr;
end character_list;
```

```
with indexing_table_for_charlist
package sorted_indexing_table_for_charlist is
        type sortedindextable
        function create_sortedindextable(sit:in out sortedindextable,
                cl:ptr charlist);
        procedure insert_pair_into_sortedindextable(
                sit:in out sortedindextable, ln:linenumber, cp:charptr);
        -- several charptrs can share the same linenumber
        function number_of_indices(it:sortedindextable) returns index;
        function get_linenumber_from_sortedindextable(sit:sortedindextable,
                i:index) returns linenumber;
        function get_charptr_from_sortedindextable(sit:sortedindextable,
                i:index) returns charptr;
        -- exceptions for when insertion does not preserve sortedness
end character_list;
```

-- by adjusting whether new charlists are used for circular_shifts and
-- alphabetized_circular_shifts, it is possible to change the
-- implementations of circular_shifts and alphabetized_circular_shifts
-- relatively independently.

Method 1:

```
with sorted_indexing_table_for_charlist
package line_storage is
                -- the internal storage consists of a new sortedindextable pointing
                -- to a new charlist
                --
end line_storage;

with sorted_indexing_table_for_charlist
package circular_shifts is
                -- the internal storage consists of a new sortedindextable pointing
                -- to an already allocated charlist, namely that of line_storage
                procedure cssetup(rcl:ref charlist);
                --
end circular_shifts;

with indexing_table_for_charlist
package alphed_circular_shifts is
                -- the internal storage consists of a new indextable pointing
                -- to an already allocated charlist, namely that of line_storage
                procedure alph(rcl:ref charlist);
                --
end alphed_circular_shifts;
```

Method 2:

```
with sorted_indexing_table_for_charlist
package line_storage is
                -- the internal storage consists of a new sortedindextable pointing
                -- to a new charlist
                --
end line_storage;

with sorted_indexing_table_for_charlist
package circular_shifts is
                -- the internal storage consists of a new sortedindextable pointing
                -- to a new charlist
                procedure cssetup();
                --
end circular_shifts;

with indexing_table_for_charlist
package alphed_circular_shifts is
                -- the internal storage consists of a new indextable pointing
                -- to an already allocated charlist, namely that of circular_shifts
                procedure alph(rcl:ref charlist);
                --
end alphed_circular_shifts;
```

Method 3:

```
with sorted_indexing_table_for_charlist
package line_storage is
                -- the internal storage consists of a new sortedindextable pointing
                -- to a new charlist
                --
end line_storage;

with sorted_indexing_table_for_charlist
package circular_shifts is
                -- the internal storage consists of a new sortedindextable pointing
                -- to a new charlist
                procedure alph(rcl:ref charlist);
                -- but that new charlist is initialized from first one
                --
end circular_shifts;

with indexing_table_for_charlist
package alphed_circular_shifts is
                -- the internal storage consists of a new indextable pointing
                -- to a new charlist
                procedure alph(rcl:ref charlist);
                -- but that new charlist is initialized from first one
                --
end alphed_circular_shifts;

-- Now it is possible to change the ways that characters are stored
-- relatively independently of all the above.
```