Simula Standard - Errata

==========

NOTE: Parameter Specification

Simula Standard 5.4.2 states:

```
specifier
      = type [ array | procedure ]
      | label
      | switch
```

This is obviously wrong. If we follow the syntax, no-type procedure is illegal, and that was probably not the intention. No-type array is also not allowed, which inherited from Algol should mean real array.

Simula Standard should have been changed to:

```
specifier
      = type
      | [type] array
      | [type] procedure
      | label
      | switch
```

==========

NOTE: Simula Standard , Chapter 12 Simulation - RANK_IN_SQS

The Simula Standard (under the definition of Class Simulation) states:

```
ref (EVENT_NOTICE) procedure pred;   pred:- PRED;

  procedure RANK_IN_SQS (afore); Boolean afore;
  begin ref (EVENT_NOTICE) evt;
     evt:- SQS.last;
     while evt.EVTIME > EVTIME do    evt :- evt.pred;
     if afore then
        while evt.EVTIME = EVTIME do evt :- evt.pred;
     follow(evt)
  end RANK_IN_SQS;
```

Here procedure pred is redefined so that 'pred' of the first element is the head itself
and not **none** like in Simset.

In Simula, this should give 'qua-check failed'.
Which led to 'cast' error in Java.

The procedure RANK_IN_SQS must be rewritten.

NOTE: lowten

Simula Standard (Chapter 9 Class "ENVIRONMENT") states:

```
character procedure lowten(c);  character c;
      if ... ! c is illegal as lowten;
         then  error("..." ! Lowten error ;)
      else begin
            lowten:= CURRENTLOWTEN; CURRENTLOWTEN:= c
      end lowten;

Changes the value of the current lowten character to that of the
parameter. The previous value is returned. Illegal parameters are
digits, plus ("+"), minus ("-"), dot ("."), comma (","),
control characters (i.e. ISO code<32), DEL (ISO code 127),
and all characters with ISO code greater than 127.
```

Here, ISO code 32 (SPACE) is indicated as a legal LOWTEN character.

Correct the condition in the parentheses to: `(i.e. ISO code<=32)`

==========

Note:

Simula Standard (Chapter 10 Class "INPUT-OUTPUT") states:

The user's main program acts as if it were embedded as follows:

```
BASICIO (inlength, outlength) Note ! prefixed block;
      inspect SYSIN do
      inspect SYSOUT do
      begin <external-head> <program> end
end prefixed block
```

The word **Note** must be replaced by **begin**

==========