

```

package dexpressions is
  type expression_type is (e_null, e_const, e_var, e_un, e_bin);
  type un_op           is (neg, sin, cos, exp, ln);
  type bin_op          is (add, sub, prod, quot, power);

  type expression is private;

  --Operacions de construcció
  function b_null           return expression;
  function b_constant (n: in integer) return expression;
  function b_var      (x: in character) return expression;
  function b_un_op    (op: in un_op; esb: in expression) return expression;
  function b_bin_op   (op: in bin_op; e1, e2: in expression) return expression;

  --Operacions per llegir components
  function e_type (e: in expression) return expression_type;
  function g_const(e: in expression) return integer;
  function g_var  (e: in expression) return character;
  procedure g_un (e: in expression; op: out un_op; esb: out expression);
  procedure g_bin(e: in expression; op: out bin_op; esb1, esb2: out expression);
private
  type node;
  type expression is access node;
  subtype pnode is expression;
  type node (et: expression_type) is record
    case et is
      when e_null =>
        null;
      when e_const =>
        val: integer;
      when e_var =>
        var: character;
      when e_un =>
        opun: un_op;
        esb: pnode;
      when e_bin =>
        opbin: bin_op;
        esb1, esb2: pnode;
    end case;
  end record;
end dexpressions;

```