Examen de Lenguajes de Alto Nivel

Febrero 2007

1) (2 puntos)

Implementar un procedimiento que obedezca a la especificación indicada abajo, y sirva para leer del fichero de texto denominado `Nombre` las coordenadas de los puntos de un polígono en el plano. El fichero contiene un número de líneas igual a los puntos del polígono, y en cada línea contiene las dos coordenadas del punto separadas por espacios en blanco. Realizar el tratamiento de las excepciones que se puedan producir.

```
type Punto is record
   X, Y : Float;
end  record;
subtype Num_Puntos is Integer range 0..20;
type Lista_Puntos is array (1..Num_Puntos'Last) of Punto;

type Poligono is record
   N : Num_Puntos:=0;
   Puntos : Lista_Puntos;
end record;

procedure Lee_Poligono (Nombre : String;
                        P : out Poligono);
```

2) (2 puntos)

Se dispone del tipo de dato `Coordenada_UTM` definido de la siguiente manera:

```
package Coordenadas is

   type Coordenada_UTM is record
      X : Float;
      Y : Float;
      Huso : Integer;
   end record;

   type Coordenada_UTM_Ref is access Coordenada_UTM;

   function Distancia (C1,C2 : Coordenada_UTM) return Float;
   -- Devuelve la distancia entre C1 y C2 en Km.

end Coordenadas;
```

Tomando como base en este tipo, se define una ruta como una lista de punteros a coordenadas de la siguiente manera:

```
with Coordenadas, Listas_Doblemente_Enlazadas;
use Coordenadas;

package Rutas is new
   Listas_Doblemente_Enlazadas(Coordenada_UTM_Ref,"=");
```

El paquete `Listas_Doblemente_Enlazadas` corresponde al ADT que se ha visto en clase.

Crear un procedimiento compilado separadamente al que se le pasa una ruta (lista del paquete `Rutas`) y una coordenada, y devuelve la distancia de la coordenada a la ruta (calculada como la menor de las distancias a cualquier punto de la ruta).

3) (4 puntos)

Se desea realizar una aplicación de gestión de la producción de una fábrica de galletas. La fábrica dispone de 10 líneas de producción capaces de producir un sólo tipo de galleta en un instante dado una vez que se ha configurado. La galletera es capaz de producir hasta 50 tipos de galletas diferentes. El objetivo es disponer de una base de datos capaz de contabilizar y registrar la producción por tipos de galleta y por línea de producción.

El tipo `Linea_Productora` se implementa en el paquete `Lineas_Productoras`, que funciona tal como se explica en los comentarios:

```
package Lineas_Productoras is

   type Codigo_Galleta is new integer range 0..49;

   type Linea_Productora is private;
   type Linea_Productora_Ref is access Linea_Productora;

   procedure Asigna_Galleta
     (L : in out Linea_Productora; C : Codigo_Galleta);
   -- Asigna el código de la galleta que producirá la línea
   -- desde ese instante.

   procedure Incrementa (L   : in out Linea_Productora;
                         Num : Natural);
   -- Incrementa en Num la cantidad de galletas producidas del
   -- tipo que tenga asignado la línea.
   -- Puede elevar Incorrecto si no se ha asignado ningún tipo.

   function Tipo_Galleta (L : Linea_Productora)
       return Codigo_Galleta;
   -- Devuelve el código de la galleta que está produciendo
   -- la línea.
   -- Puede elevar Incorrecto si no se ha asignado ningún tipo.

   function Num_Producidas (L : Linea_Productora;
                            C : Codigo_Galleta)
                           return Natural;
   -- Devuelve el número de galletas producidas por la línea
   -- del tipo indicado.

   Incorrecto : exception;

private

   type Linea_Productora is ...

end Lineas_Productoras;
```

Se pide realizar la implementación de la parte privada de este paquete así como del cuerpo para que funcione de acuerdo con la especificación.

Se pide además realizar el diseño de un paquete capaz de gestionar una fábrica de galletas cualquiera (en particular la propuesta, que tiene 10 líneas de producción). La base de datos de la fábrica tendrá las siguientes operaciones:

- Obtener el número de galletas producido de un tipo
- Obtener el número de galletas de producido por una determinada línea
- Obtener el número total de galletas producido por la planta.

En el diseño e implementación se deben utilizar los ADTs vistos en clase que se consideren más adecuados, justificando su uso.

4) (1 punto)

Se dispone del siguiente tipo etiquetado junto con una de sus operaciones:

```
subtype CM3 is Float range 0.0..500.0;

type Frigorifico is tagged record
   Fabricante : String(1..30);
   N_Serie : String(1..12);
   El_Volumen : CM3;
end record;

type Frigorifico_Ref is access Frigorifico'Class;

function Volumen (F : Frigorifico) return CM3;
```

Se pide definir un nuevo tipo etiquetado que sea frigoríco congelador que posee un campo más que es el volumen del congelador. Además, se debe redefinir la operación `Volumen` para que devuelva el volumen total (frigorífico más congelador).

5) (1 punto)

Indicar los fallos de compilación que daría el siguiente fragmento de programa:

```
procedure Principal is

   package P1 is
      procedure A;
      function B return Integer;
      J : Integer;
   end P1;

   package body P1 is
      E : exception;
      function B return Integer is
      begin
         return I+J;
      end B;
      I : Integer;
      procedure A is
      begin
         I:=I+1000.0;
         if I<0 then
            raise E;
         end if;
      end A;
   begin
      I:=0;
      J:=1000;
   end P1;

begin
   J:=2000;
   if B>3_000 then
      for I in 1..3_000 loop
         P1.A;
      end loop;
   end if;
exception
   when P1.E =>
      P1.J:=0;
      P1.I:=0;
end Principal;
```