Devoir surveillé

novembre 2015

Eléments de correction

1 - Type abstrait

[*barême indicatif :*]

Question 1-1 :

Le paquetage **cartes_paq** n'est pas visible dans la procédure `prog1`. Il faut indiquer qu'on utilise ce paquetage (clause `with`) et rendre ce paquetage visible (clause `use`). Ainsi il faut ajouter en début du fichier `prog1.adb` :

```
with cartes_paq;
use cartes_paq;
```

Question 1-2 :

Puisque le type `Carte` est privé il faut ajouter au paquetage **cartes_paq** les sélecteurs et constructeurs nécessaires à la manipulation de variables du type `Carte`. Par exemple on va ajouter :

- une fonction `la_carte` qui construit une carte avec une `Valeur` et une `Couleur` données,
- des fonctions `val` et `coul` qui donnent respectivement la valeur et la couleur d'une carte donnée.

Dans le fichier **Cartes_paq.ads** on ajoute les profils suivants (ligne 13) :

```
function la_carte (v:Valeur; c:Couleur) return Carte;
function val (c:Carte) return Valeur;
function coul (c:Carte) return Couleur;
```

De plus il faut créer un fichier **Cartes_paq.adb** qui contient la réalisation des fonctions précédentes :

```
package body carte_paq is
   function la_carte (v:Valeur; c:Couleur) return Carte is
   ca : Carte;
   begin
     ca.v := v; ca.c := c;
     return ca;
   end la_carte;

   function val (c:Carte) return Valeur is
   begin
     return c.v;
   end val;

   function coul (c:Carte) return Couleur is
   begin
     return c.c;
   end coul;
end cartes_paq;
```

Les modifications à apporter au fichier **prog1.adb** sont les suivantes :

- remplacer les lignes 11 et 12 par `c1 := la_carte (huit, trefle)`
- remplacer les lignes 14 et 15 par `c2 := la_carte (dame, coeur)`
- remplacer la ligne 18 par `put_line("Valeur de c1 " & Valeur'image(val(c1)))`
- remplacer la ligne 20 par `put_line("Couleur de c2 " & Couleur'image(coul(c2)))`
- remplacer le test de la ligne 23 par `if val(c1) < val(c2)`
- remplacer le test de la ligne 24 par `elsif val(c1) > val(c2)`

2 - Entrées-Sorties [*barême indicatif :*]

Question 2-1 :

En instanciant le paquetage générique d'entrées-sortie des types énumérés :

```
with cartes_paq, ada.text_io;
use cartes_paq, ada.text_io;

package body donne_paq is

package ES_Valeur is new ada.text_io.enumeration_io(Valeur);
package ES_Couleur is new ada.text_io.enumeration_io(Couleur);
use ES_Valeur;
use ES_Couleur;

procedure lire_donne (nom_f : in string; D : out donne) is
v : Valeur;
c : Couleur;
f : file_type;
begin
   open(f, in_file, nom_f);
   for i in Joueur loop
      for j in Num_carte loop
         get(f, v);
         get(f, c);
         D(i, j) := la_carte(v, c);
         skip_line(f);
      end loop;
   end loop;
   close(f);
end lire_donne;

procedure ecrire_donne (nom_f : in string; D : in donne) is
f : file_type;
begin
   create(f, out_file, nom_f);
   for i in Joueur loop
      for j in Num_carte loop
         put(f, val(D(i,j)));
         put(f, coul(D(i,j)));
         new_line(f);
      end loop;
   end loop;
   close(f);
end ecrire_donne;

end donne_paq;
```

Au lieu d'utiliser le paquetage générique d'entrée-sortie des types énumérés, comme précédemment, on peut par analogie avec le programme **prog1** utiliser l'attribut `image` des types `Valeur` et `Couleur`.

```
procedure ecrire_donne (nom_f : in string; D : in donne) is
f : file_type;
begin
   create(f, out_file, nom_f);
   for i in Joueur loop
      for j in Num_carte loop
         put(f, Valeur'image(val(D(i,j))));
         put(f, Couleur'image(coul(D(i,j))));
         new_line(f);
      end loop;
   end loop;
   close(f);
end ecrire_donne;
```

Question 2-2 :

```
with ada.text_io. donne_paq, ada,commande_line;
use ada.text_io. donne_paq, ada,commande_line;

procedure prog2 is
d : donne;
begin
   if argument_count /= 2 then
      put ("donnez deux noms de fichiers en arguments";
      new_line;
   else
      lire_donne(argument(1), d);
      ecrire_donne(argument(2), d);
   end if;
end prog2;
```

3 - Assertions, couverture d'un programme

[*barême indicatif :*]

Question 3-1 :

On complète l'assertion avec un message indiquant quelle propriété a été violée :

```
assert (d>=f, "d < f" );
assert (e>f, "e <= f" );
```

Question 3-2 :

Notons a_0, b_0, c_0, d_0 les valeurs initialement lues pour les variables `a, b, c` et `d`.

1. Pour mettre en défaut l'assertion `d>=f`, ligne 20 il faut atteindre cette ligne et que l'expression booléenne soit fausse. Les conditions à remplir sont :
   (a) `d<e` ligne 12, c'est-à-dire $d_0 < a_0 * b_0$
   (b) `f>0` ligne 14, c'est-à-dire $c_0 - a_0 > 0$
   (c) `g <= a` ligne 19, d'où $a_0 * b_0 - b_0 \leq a_0$
   (d) et enfin `d<f` ligne 20, c'est-à-dire $d_0 * 2 < c_0 - a_0$

   Par exemple il suffit de prendre :
   – $b_0 = 1$ satisfait la condition (c)
   – $c_0 = a_0 + 1 = 2$ pour la condition (b)
   – la condition (d) devient : $2 * d_0 < 1$, prenons par exemple $d_0 = 0$
   – avec $a_0 = 1$ la condition (a) est vérifiée
   ce qui donne $a_0 = 1, b_0 = 1, c_0 = 2, d_0 = 0$.
2. Pour mettre en défaut l'assertion `e>f`, ligne 29 il faut atteindre cette ligne et que l'expression booléenne soit fausse. Les conditions à remplir sont :
   (a) `d<e` ligne 12, c'est-à-dire $d_0 < a_0 * b_0$
   (b) la condition sur `f>0` est indifférente
   (c) `e<=f ligne 29`, c'est-à-dire $a_0 * b_0 \leq c_0 - a_0$

   si nous prenons $a_0 = b_0 = 1$ par exemple, alors $a_0 * b_0 = 1$; avec $d_0 = 0$ nous satisfaisons la condition (a) ($d_0 < 1$) et avec $c_0 = 2$ nous satisfaisons la condition (c) ($c_0 \geq a_0 + 1$).
3. Un jeu minimal permettant de couvrir l'ensemble des instructions du programme sans mettre en défaut aucune assertion requiert au minimum 3 tests :
   – un test pour lequel les conditions $d < e$, ligne 12, $f > 0$, ligne 14, $g > a$, ligne 17 soient vraies. Dans ce cas l'assertion $e > f$ doit aussi être vraie.
   – un test pour lequel les conditions $d < e$, ligne 12, $f > 0$, ligne 14 soient vraies et $g > a$, ligne 17 fausse. De plus les assertions $e > f$ et $d \geq f$ doivent être vérifiées.
   – un test pour lequel la condition $d < e$, ligne 12, soit vraie et la condition $f > 0$, ligne 14 soit fausse. De plus la condition $e > f$ doit aussi être vraie.