ED 12
Programmation par tâches et rendez-vous Ada

Exercice : Tampon bicase

On souhaite implanter un paquetage qui gère l'échange de messages entre processus. Ce paquetage est utilisé concurremment par des processus qui en appellent les procédures :

```
procedure POSER(X : in MESSAGE);
procedure OTER(Y : out MESSAGE);
```

La programmation de la concurrence sera réalisée par tâches et rendez-vous, c'est-à-dire sans mémoire commune.

On veut construire un paquetage gérant deux cases . L'interface de ce paquetage `BICASE` s'écrit :

```
package BICASE is
      procedure POSER(X : in MESSAGE);
      procedure OTER(Y : out MESSAGE);
end BICASE;
```

Le corps du paquetage contient deux cases T1 et T2 . Le comportement du paquetage est le suivant :

- pour `POSER` :

  quand la case `T1` est vide, l'action est : `T1 := X; -- ceci remplit T1`

  une tâche `T_T1` réalise la gestion de `T1`.

- pour `OTER` :

  quand la case `T2` est pleine, l'action est : `Y := T2; -- ceci vide T2`

  une tâche `T_T2` réalise la gestion de `T2`;

- Quand `T1` est pleine et `T2` est vide, l'action est : `T2 := T1; -- ceci vide T1 et remplit T2`

  une tâche `CANAL` réalise cette action.

Question

Compléter, au moyen de rendez-vous ADA, la programmation donnée ci-dessous.

```
package BICASE is
      procedure POSER(X : in MESSAGE);
      procedure OTER(Y : out MESSAGE);
end BICASE;
package body BICASE is
```

```
task T_T1 is
       entry POSE(X : in MESSAGE);
end T_T1;
task T_T2 is
       entry OTE(X : out MESSAGE);
end T_T2;
task CANAL is
       entry EMETTRE(EMIS : in MESSAGE);
       entry RECEVOIR(RECU : out MESSAGE);
end CANAL;
procedure POSER(X : in MESSAGE) is
       begin T_T1.POSE(X); end POSER;
procedure OTER(Y : out MESSAGE) is
       begin T_T2.OTE(Y); end POSER;
task body T_T1 is
*** à programmer ***
task body CANAL is
*** à programmer ***
task body T_T2 is
*** à programmer ***
begin null; end BICASE;
```