Modele Conception Partagé
TD

kelto

TD4

Utilisation des moniteurs de Hoare pour implémenter le modèle de Producteur-Consommateur.

Variante 1

```
int Nplace_libre;
Condition full;
Condition empty;
void Deposer(M:Message)
{
  if(Nplace_libre = 0)
    wait(full);
  inserer(M);
  NPlace_libre--;
  signal(empty);
}
void Retirer(M: Message)
{
  if(NPlace_libre == N)
    wait(empty);
  extraire(M);
    NPlace_libre++;
  signal(full);
}
```

Listing 1 – Corps moniteur Prod_Cons

Variante 2

```
Condition  Full, Empty;
Condition Ctype[2];
int nb_buffer;
Type next;

void Deposer(M:Message, type:Type)
{
  if(nb_buffer==0 || type != next)
    wait(Ctype[type]);

  inserer(M);
  next = (next == Blanc ? Noir:Blanc);
  nb_buffer--;
  if(nb_buffer != 0)
    Signal(Ctype[next]);
  Signal(Empty);



}

void Retirer(M:Message, type:Type)
{
  if(nb_buffer == N)
    wait(Empty);
  extraire(M);
    nb_buffer++;

  Signal(Ctype[next]);

}
```

Listing 2 – Corps moniteur Prod_Cons

Variante 3

```
1
  Condition  Full, Empty;
3 Condition Ctype[2];
  int nb_buffer;
5 Type next;

7 void Deposer(M:Message, type:Type)
  {
9   if(nb_buffer==0)
      Wait(Full);
11
    inserer(M);
13  nb_buffer--;
    Signal(Empty);
15



17
  }
19
  void Retirer(M:Message, type:Type)
21 {
    if(nb_buffer == N || peek_next() != type)
23    wait(Ctype[type]);
    extraire(M);
25    nb_buffer++;
    if(nb_buffer != N)
27    Signal(Ctype[peek_next()]);
    Signal(Full);
29
  }
```

Listing 3 – Corps moniteur Prod_Cons