```
monitor DP {
      enum ( THINKING, HUNGRY, EATING) state [5] ;
      condition self [5];

      void pickup (int i) {
             state[i] = . . .;
             . . .;
             if (state[i] != . . .)
                 ...;
      }

      void putdown (int i) {
             state[i] =  . . .;
             // test left and right neighbors
             test((i + 4) % 5);
             test((i + 1) % 5);
      }

      void test(int i)
      //If my neighbors are not in eating state and I am
      //hungry, then I can eat!

      {
         if ( (state[(i + 4) % 5] != . . .) &&
               (state[i] == HUNGRY) &&
               (state[(i + 1) % 5] != . . .) )
         {

               state[i] = . . .;
                self[i].signal () ;
          }

      }
      initialization_code() { //initial state is THINKING
         for (int i = 0; i < 5; i++)
                state[i] = . . .;
            }
      }
`
```

```
//Usage

dp.pickup (i)

//EAT

dp.putdown (i)

 //THINK
```