

```
1  /**
2   * PC0 2015 Ex. 11 (c) YTA Yann Thoma
3   * Barrière de synchronisation (rendez-vous de N threads)
4   * Objectif: réalisé. Commentaire: solution optimale.
5   */
6
7  class PcoBarrier{
8
9  private:
10
11     const unsigned int nbToWait; // nombre max à attendre
12     unsigned int waiting = 0; // nombre en attente actuellement
13     QSemaphore* mutex; // protège waiting
14     QSemaphore* attente; // bloque/file d'attente
15
16 public:
17
18     PcoBarrier(unsigned int nbToWait): nbToWait(nbToWait) {
19         // équiv à: this.nbToWait = nbToWait; mais const !
20         mutex = new QSemaphore(1); // ouvert
21         attente = new QSemaphore(0); // fermé
22     }
23
24     ~PcoBarrier() {
25         delete mutex;
26         delete attente;
27     }
28
29     void wait() {
30         mutex->aquire();
31         waiting++;
32
33         if (waiting == nbToWait) {
34             // libération de tous les autres en attente
35             // !! ne pas compter soi-même et revenir à l'état initial
36             for(; waiting > 1; waiting--) {
37                 attente->release();
38             }
39             waiting = 0;
40             mutex->release();
41         } else {
42             // waiting < nbToWait: mise en attente
43             mutex->release();
44             attente->aquire();
45         }
46     }
47 }; // fin de la classe PcoBarrier
```