```
/*
 * Classe qui permet de gérer un parking privé avec un seul accès.
 * (il s'agit de la vraie barrière)
 * Le nombre de place est limité à la valeur que vous désirez,
 * et l'accès se fait par un code secret qui est à préciser.
 */

#ifndef _PARKING_H
#define _PARKING_H

#include <ServiceBarriere.h>
#include <barriere.h>
#include <clavier.h>

//--------------------------------------------------------------------------------
class Parking
{
  ServiceBarriere service;
  Afficheur afficheur;
  Clavier clavier;
  Barriere barriere;
  const int codeSecret;
  const int nombrePlace;
  int disponible;
public:
  Parking(int places, int code);
  void run();
private:
  void affichage();
  void entrer();
  void sortir();
};
//--------------------------------------------------------------------------------

/* Constructeur de la classe qui permet d'initaliser correctement
 * l'ensemble des éléments de la vraie barrière en spécifiant en même temps
 * le nombre de place maxi autorisé ansi que le code d'accès
 * ------------------------------------------------------------------------------*/
inline Parking::Parking(int places, int code)
: service("172.20.3.73"), afficheur(service), barriere(service), clavier(service),
  codeSecret(code), nombrePlace(places)
{
  disponible = places;
}


/* Gestion globale de la barrière avec un contrôle
 * sur l'entrée et la sortie des véhicules
 * ------------------------------------------------------------------------------*/
inline void Parking::run()
{
  // Faire descendre la barrière pour être dans les bonnes conditions initiales
  if (barriere.getPosition() != Barriere::enBas) barriere.setCommandeSensDescente();
  // Affichage d'invite
  affichage();

  while (true)
  {
    // Contrôle de la présence d'un véhicule pour entrer ou pour sortir
    switch (barriere.getSituation())
    {
```

```
      case Barriere::enEntree :
        if (disponible>0)
        {
          entrer();
          disponible--;
          affichage();
        }
        break;

      case Barriere::enSortie :
        sortir();
        disponible++;
        affichage();
        break;
    }
  }
}

/* Message d'invite qui indique le nombre de places disponibles ou parking complet
 * ----------------------------------------------------------------------------*/
inline void Parking::affichage()
{
  switch (disponible)
  {
    case 0 :
       afficheur.clear();
       afficheur.affiche(0, 0, "PARKING COMPLET");
       break;

    case 1 :
       afficheur.centre("Bienvenue", "1 place libre");
       break;

    default:
       {
         char chaine[17];
         sprintf(chaine,"%d places libres", disponible);
         afficheur.centre("Bienvenue", chaine);
       }
       break;
   }
}

/* Procédure pour entrer complètement dans le parking avec saisie du code d'accès
 * ----------------------------------------------------------------------------*/
inline void Parking::entrer()
{
  // solliciter la saisie du code
  afficheur.affiche("Bonjour", "Code ? ");

  // récupérer le bon code secret
  do {
    // si le code est incorrect, avertir l'automobiliste
    if (!clavier.codeBon(codeSecret)) afficheur.affiche("Code incorrect", "Code ?
");
  }
  while (!clavier.isValide());

  // afficher un message précisant la bonne valeur du code
  afficheur.affiche("Code bon", "Entrez...");
```

```
  // lever la barrière pour permettre l'entrée du véhicule
  barriere.setCommandeSensMontee();

  // attendre que le véhicule pénètre complètement dans le parking
  while (barriere.getSituation() != Barriere::enSortie);
  while (barriere.getSituation() == Barriere::enSortie);

  // descendre la barrière une fois que le véhicule est bien entré
  barriere.setCommandeSensDescente();
}

/* Procédure pour sortir complètement du parking
 *--------------------------------------------------------------------------------*/
inline void Parking::sortir()
{
  // lever la barrière pour permettre la sortie du véhicule
  barriere.setCommandeSensMontee();

  // attendre que le véhicule sorte complètement du parking
  while (barriere.getSituation() != Barriere::enEntree);
  while (barriere.getSituation() == Barriere::enEntree);

  // descendre la barrière une fois que le véhicule est bien sortie
  barriere.setCommandeSensDescente();
}

//--------------------------------------------------------------------------------
#endif  /* _PARKING_H */
```