

```
/*
 * Classe spécifique à la gestion du clavier de la borne d'accès
 * avec gestion d'un historique
 */
#ifndef _CLAVIER_H
#define _CLAVIER_H

#include <Socket.h>
#include <string.h>
#include <afficheur.h>

//-----
class Clavier
{
    Socket &barriere;
    int derniereTouche;
    bool valide;
public:
    Clavier(Socket &service) : barriere(service) {}
    int readClavier();
    int toucheEnfoncée();
    bool codeBon(int codeSecret);
    int getDerniereTouche();
    bool isValid();
};
//-----

// Permet de scanner le clavier physique et de récupérer le code de la touche enfoncée.
inline
int Clavier::readClavier()
{
    barriere.envoyer("Clavier:readClavier");
    return atoi(barriere.recevoir());
}

// Cette méthode fait appel à la méthode readClavier(). Elle est bloquante.
// Elle attend qu'une touche soit effectivement enfoncée.
inline
int Clavier::toucheEnfoncée()
{
    int touche;
    while ((touche = readClavier()) == -1);
    return derniereTouche = touche;
}

// Contrôler la saisie du code de l'automobiliste (4 chiffres).
// Il doit correspondre au code secret passé en argument de la fonction
inline
bool Clavier::codeBon(int codeSecret)
{
    Afficheur afficheur(barriere);
    int code = 0;
    for (int i=0; i<4; i++) {
        code *= 10;
        code += toucheEnfoncée();
        afficheur.afficheCaractere('#');
    }
    return valide = code == codeSecret;
}
```

```
// Retourne la dernière touche qui a été enfoncée
inline
int Clavier::getDerniereTouche()
{
    return derniereTouche;
}

// Indique la validité du code secret après coup
bool Clavier::isValide()
{
    return valide;
}

#endif
```