

## TP POO – Implémentation du digicode

Le projet à partir duquel vous commencez votre travail contient du code et certaines classes que vous pourrez utiliser :

- une fonction `C _kbhit()` qui vérifie si une touche a été saisie au clavier. Elle retourne 0 quand rien n'a été saisi ou le code ASCII du caractère sinon ;
- la classe *Chrono* qui permet de décompter le temps écoulé ;
- la classe *BD* qui lit le fichier `Base.txt` pour vérifier si un code d'entrée saisi y existe ;
- la classe *Archivage* qui enregistre toute entrée dans le bâtiment dans un fichier `Archive.txt` ;
- la classe *InterfaceDeSaisie* que vous aurez à compléter en partie D.

### Partie A

Dans cette partie, vous devez écrire les classes *Alarme*, *Capteur*, *Voyant*, *Clavier*, en suivant les indications ci-dessous.

Vous testerez dans la fonction *main* chacune des méthodes au fur et à mesure de l'écriture de vos classes.

N'oubliez pas de sauvegarder chaque version de votre travail, au moins après l'ajout d'une classe.

- La méthode *Alarme::declencher()* démarre une boucle infinie qui affiche un message d'alarme ("Alarme : la porte n'a pas été refermée") puis fait une pause de 1s ; vous utiliserez la fonction *sleep* définie dans `<unistd.h>` ;
- la méthode *Capteur::detecter()* renvoie 0 si rien n'a été saisi au clavier, autre chose que 0 sinon ; vous utiliserez la fonction *\_kbhit* décrite plus haut ;
- la méthode *Clavier::saisirChiffre()* renvoie -1 si si rien n'a été saisi au clavier, et l'entier correspond au caractère saisi sinon ; vous utiliserez la fonction *\_kbhit* décrite plus haut. Vous considèrerez que le seul clavier disponible est le clavier numérique ;
- dans le fichier *Voyant.h*, déclarez 2 énumérations *Couleur* (valeurs *rouge* et *vert*) et *Etat* (valeurs *allume* et *eteint*), que vous utilisez comme type des données membre d'un *Voyant* ;
- la méthode *Voyant::getStatut()* retourne la chaîne "ON" ou "OFF" suivant l'état.

### Partie B

Dans cette partie, vous allez écrire la classe *ObjetTempo*, qui utilisera la classe *Chrono* qui est fournie :

- on transmet au constructeur un entier, le nombre de secondes pendant lequel cet objet est temporisé (en plus du chronomètre...) ;
- la méthode *ObjetTempo::finTempo()* arrête l'attente (=0).

### Partie C

Vous implémentez maintenant la classe *Porte* (qui est un *ObjetTempo*), en reprenant les diagrammes de séquence que vous avez travaillés dans le TP correspond de M2104.

- écrivez le constructeur ;
- la méthode *Porte::ouvrir()* passe en attente (=1), démarre le chronomètre, puis demande en boucle au capteur s'il a détecté une entrée, et ce tant que le chronomètre ne l'arrête pas. Dès qu'une entrée est détectée, la méthode *Porte::fermer()* est appelée ;
- la méthode privée *Porte::fermer()* interroge aussi en boucle le capteur, et, le cas échéant, arrête le chronomètre ;
- si la porte n'est pas refermée, la procédure d'ouverture échoue.

## Partie D

Vous implémentez maintenant la classe *InterfaceDeSaisie* (qui est un *ObjetTempo*), en reprenant les diagrammes de séquence que vous avez travaillés dans le TP correspond de M2104.

Dé commentez ce qu'il faut dans la méthode *Interface::statut()* pour pouvoir l'utiliser.

## Partie E

Ecrivez enfin la classe *Système*, qui est composée de tous les objets et qui coordonne leurs responsabilités.

Dans la fonction *main*, vous créerez un objet de classe *BD*, un de classe *Archivage*, et enfin un objet de la classe *Système* qui démarrera son travail.