

Liste des diagrammes de classes :

- ✓ *ClassDia_Complet.pdf* : Diagramme de classe complet de l'application.
- ✓ *ClassDia_CompletLight.pdf* : Diagramme de classe complet de l'application sans les 'identifieurs' des patrons de conception.
- ✓ *ClassDia_PatronCommand01.pdf* : Diagramme de classe de l'utilisation du patron 'Command' entre l'horloge et la présentation.
- ✓ *ClassDia_PatronCommand02.pdf* : Diagramme de classe de l'utilisation du patron 'Command' entre l'horloge et le moteur.
- ✓ *ClassDia_PatronCommand03.pdf* : Diagramme de classe de l'utilisation du patron 'Command' entre le moteur et le contrôleur.
- ✓ *ClassDia_PatronObserver01.pdf* : Diagramme de classe de l'utilisation du patron 'Observer' entre la présentation et le contrôleur.
- ✓ *ClassDia_PatronObserver02.pdf* : Diagramme de classe de l'utilisation du patron 'Observer' entre le moteur et le contrôleur.

Détail des patrons utilisés :

- 1) Le patron 'Command' entre l'horloge et le moteur permet de déclencher les actions de marquage du temps sur le moteur. L'implémentation de l'horloge reçoit la commande 'CmdImp_TopSynchro' et l'exécute à une fréquence donnée par le moteur. Cette commande appelle la fonction `marquagePériodique()` du moteur.
- 2) Le patron 'Command' entre le moteur et le contrôleur permet de déclencher les actions de marquage du temps, et si nécessaire de la mesure, sur le contrôleur. La classe `MoteurMetronomeImple` reçoit les commandes 'CmdImp_marqueTps' et 'CmdImp_marqueMes' et les exécute à chaque marquage périodique, en fonction du paramètre de nombre de temps dans une mesure. Ces commande appelle les fonctions `marqueTemps()` et `marqueMesure()` sur le contrôleur.
- 3) Le patron 'Command' entre l'horloge et la présentation permet d'éteindre les Leds sur la présentation au bout d'un temps donné. L'implémentation de l'horloge reçoit la commande 'CmdImp_stopLed' et l'exécute après un délai passé en paramètre. Cette commande appelle la fonction `eteindreLesLeds()` de la présentation (classe `IHMImple`).
- 4) Le patron 'Observer' entre la présentation et le contrôleur permet de surveiller les modifications des valeurs de tempo, de nombre de temps dans une mesure, et de mise en marche effectuer par l'IHM. Lorsqu'une modification est notifiée, la fonction `actualise()` du contrôleur est appelée pour récupérer et traiter les nouvelles valeurs.
- 5) Le patron 'Observer' entre le moteur et le contrôleur permet de surveiller les modifications des valeurs de tempo et de nombre de temps dans une mesure effectuer par le moteur. Lorsqu'une modification est notifiée, la fonction `actualise()` du contrôleur est appelée pour récupérer et traiter les nouvelles valeurs.