moteur

DemarrageMetronome
+main()

CommandInterf
+executer() : Boolean

+marqueMesure 0..1
+marqueTemps 0..1
+cmdPeriodique 0..1
utilGenerale
+moteur 1

MoteurMetronomeInterf
+setEnMarche(val : boolean) : void
+getEnMarche() : Boolean
+setNbTempsParMesure(nbTemps : Integer) : void
+getNbTempsParMesure() : Integer
+setTempo(tempoBpm : Integer) : void
+getTempo() : Integer
+marquagePeriodique() : Boolean
+setCommandesCtrl(cmdTemps : CommandInterf, cmdMesure : CommandInterf) : void

CmdImp_topSynchro
+executer() : Boolean

+cmdStopLed 0..1

MoteurMetronomeImple
-enMarche : boolean
-nbTempsParMesure : Integer
-tempo : Integer
...

+moteur 1
-moteur 0..1

Invoker
ConcreteCommand
Receiver / client
ConcreteSubject

controleur

-ctrl 1

GestionEvtsInterf
+marquerTemps() : Boolean
+marquerMesure() : Boolean

+eteindreLed 0..1
Receiver / Client

CmdImp_marqueTps
+executer() : Boolean

ControleurMetronome
-dureeAllumageLed : float
+marquerMesure() : Boolean
+marquerTemps() : Boolean
+getMoteur() : MoteurMetronomeInterf
+setDureeAllumage(dureeAllumageLed : float) : void
+getDureeAllumageLed() : float

-contoleur 0..1
-contoleur 0..1

CmdImp_marqueMes
+executer() : Boolean

-monContoleur 1

Command
ConcreteCommand
ConcreteObserver

Observer

utilGenerale

ObservateurInterf
+actualise()

-observateurs 0..*
-observateurs 0..*

Observer
Observer
Subject

SujetObservableInterf
+ajoutObservateur(obs : ObservateurInterf)
+suppObservateur(obs : ObservateurInterf)
+notificationModif()

outillageExterne

+horloge 1

HorlogeInterf
+activerPeriodiquement(cmd : CommandInterf, periodeSec : float) : void
+activerApresDelai(cmd : CommandInterf, delaiMilliSec float) : void
+arretHorloge() : void

-horloge 1

HorlogeImple
-cmdPeriodique : CommandInterf

Invoker
Subject

presentation

IHMInterf
+flasherLed1(tpsMilliSec : float) : void
+flasherLed2(tpsMilliSec : float) : void
+emettreSon() : void
+setPositionMolette(position : Integer)
+getPositionMolette() : integer
+setMarche(val : Boolean)
+getMarche() : Boolean
+setTempsParMesure(val : Integer)
+getTempsParMesure() : Integer
+setInfosTempo(val : Integer)

-ihm 0..*
Command
ConcreteObserver
Receiver / Client
ConcreteCommand

1 -presentation

IHMImple
...
+eteindreLesLeds() : void

ConcreteSubject

CmdImpl_stopLed
+executer() : Boolean

Favereau M. et Legendre L.