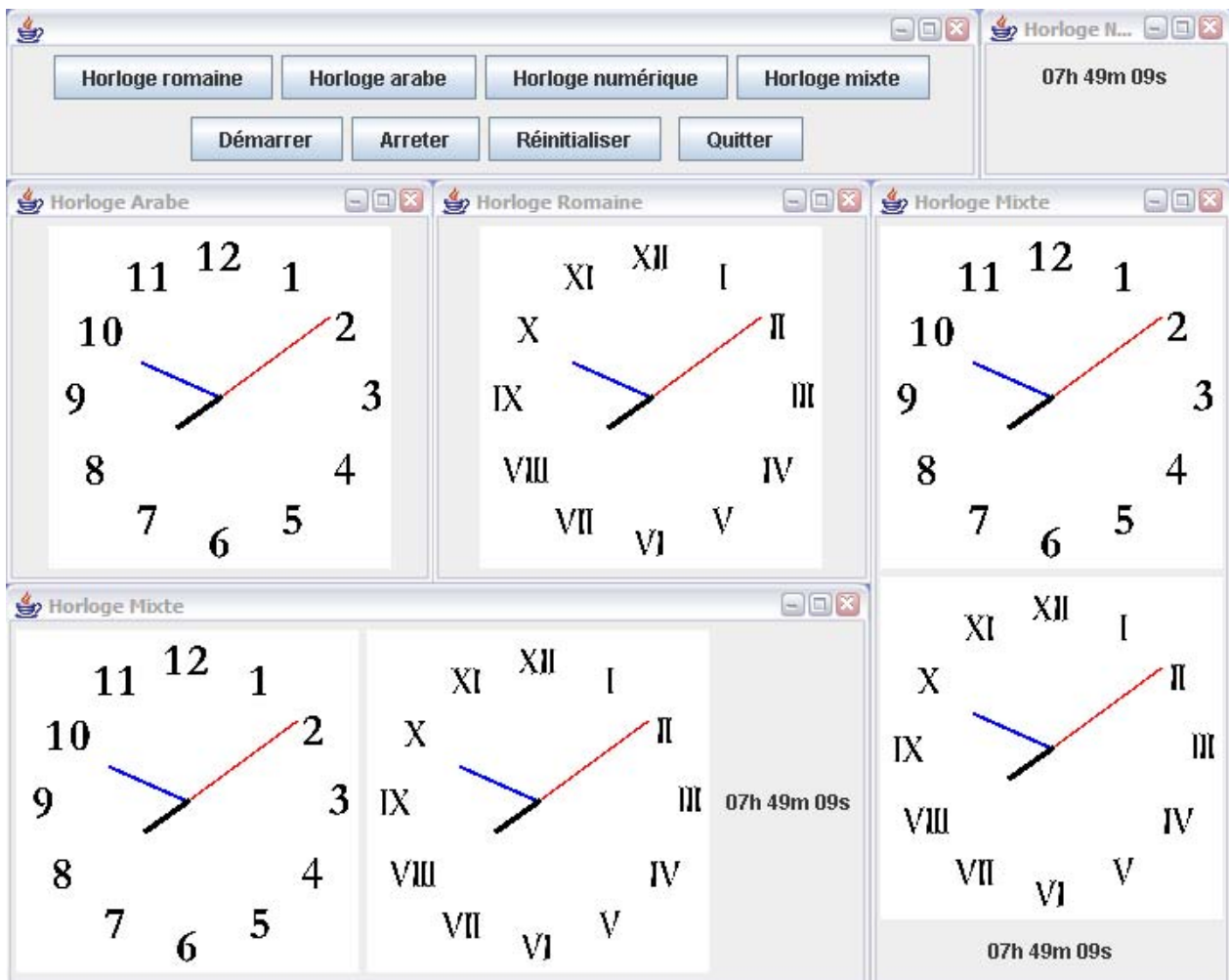


## Laboratoire 1: Observateur

*Durée du laboratoire: 2 séances. A rendre le jeudi 5 mars 2020, au début du cours sur Cyberlearn ainsi qu'une version imprimée.*



### 1. Objectifs

Concevoir une application utilisant le pattern Observateur (GoF).

### 2. Indications

Développer une application graphique qui permette de gérer un chronomètre. Les opérations possibles sur le chronomètre incluent le démarrage du chronomètre, son arrêt (pause) et sa remise à zéro. Il mesure le temps en secondes/minutes/heures.

Le chronomètre peut être visualisé de différentes façons (cf. figure):

- Sous forme numérique

- Sous forme analogique, avec chiffres romains
- Sous forme analogique, avec chiffres arabes
- Par combinaisons de ces vues

Le démarrage, l'arrêt et la réinitialisation du chronomètre seront contrôlés par une interface graphique supplémentaire. D'autre part, un click sur une vue met en pause le chronomètre. Un deuxième click le redémarre.

L'application doit permettre à l'utilisateur de créer n'importe quelle vue. Les vues sont redimensionnables.

### **3. Implémentation**

- Implémenter votre propre version du modèle Observateur (ne pas utiliser l'interface `IObserver` et classe `Observable` fournies dans l'API Java).
- Pour la gestion du temps utiliser la classe `Timer`.
- Pour charger une image depuis un fichier utiliser:  
`Toolkit.getDefaultToolkit().getImage(fileName);`
- Pour redimensionner une image utiliser:  
`imageScaled = image.getScaledInstance(width, height, mode);`

### **4. Travail à rendre**

- Diagramme de classes.
- Sources du programme.