Département de génie électrique et de génie informatique
Faculté des sciences et de génie

Génie logiciel orienté objet

IFT-20829

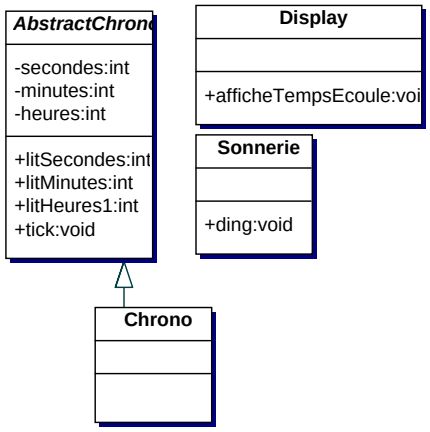
Examen partiel #1

20% note du trimestre
Documentation permise

QUESTION 1 **(25 points au total) Design patterns (1)**

Le diagramme de classes de la Figure 1 montre une classe abstraite *AbstractChrono* définissant une interface décrivant le fonctionnement d'un chronomètre. La méthode *tick()* représente le passage d'une seconde.

Figure 1 Diagramme de la Question1.



Les méthodes *litSecondes*, *litMinutes* et *litHeures* permettent d'accéder aux valeurs des secondes, minutes et heures respectivement. La classe Chrono est une classe concrète implantant l'interface de la classe *AbstractChrono*. Lorsqu'une seconde passe (tick()), le Display doit afficher les heures, les minutes et les secondes et la Sonnerie doit faire entendre un tintement. Comment utiliseriez-vous le pattern *Observer* pour paramétriser les interactions entre Chrono, Display et Sonnerie de manière à ce que le fonctionnement de ce système simple soit le plus flexible possible au run-time. Vous pouvez ajouter les classes que vous désirez à celles montrées à la Figure 1.

Donnez le diagramme de classe de votre réponse.

QUESTION 2 **(25 points au total) Design patterns - Collaboration entre classes (2)**

Vous devez concevoir une application pour laquelle un choix de menu doit enclencher l'impression d'un document. On souhaiterait éviter de relier directement l'objet implantant le menu à l'objet responsable de l'impression. De plus, dépendant du contexte de l'application au run-time, deux implantations de l'impression peuvent être possibles. La première implante une impression en postscript sur une imprimante laser, tandis que l'autre implante une impression en bitmap sur une imprimante à point. Finalement, dans le cas de l'impression en bitmap, si le document a une longueur d'une page, le contenu du document s'imprime alors que lorsque le document a une longueur de plus d'une page, le contenu du document s'imprime normalement *et* un cadre délimite les marges de la deuxième page et des pages suivantes.

Utilisez les patterns *mediator*, *bridge* et *decorator* pour modéliser cette application. Donnez le diagramme de classes de votre réponse. Identifiez clairement les patterns sur le diagrammes.

QUESTION 3 **(25 points au total) Design patterns (3)**

Une application doit utiliser un environnement offrant des widgets[1]de type Button et Menu. On voudrait qu'il soit possible de choisir lors du lancement de cette application le type de Button et de Menu que l'on désire utiliser, soit ceux fournis dans la librairie de Windows ou soit ceux de la librairie Motif. Quelle que soit la librairie utilisée, un Button et un Menu est composé des propriétés suivantes: color (int), shape (int) et behavior (int). Utilisez le pattern Abstract Factory pour modéliser ce type d'application. Votre modèle doit respecter les contraintes suivantes:

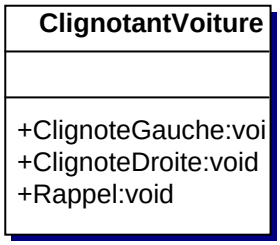
1) votre abstract factory doit être un singleton;
2) la production d'un objet Button ou Menu doit s'effectuer en utilisant une Factory Method
3) La classe implantant la Factory Method doit utiliser le pattern Builder pour construire les objets Button ou Menu (i.e. construire leur différents descripteurs soit color, shape et behavior)

Dessinez le diagramme satisfaisant ces exigences.

QUESTION 4 **(25 points au total) Diagramme d'état**

Supposons que la classe très simple de la Figure 2 serve à décrire un clignotant de voiture pour signaler à gauche ou à droite. Quand la voiture est démarrée, le clignotant entre dans un état d'attente de commande. Quand la méthode ClignoteGauche est appelée, le clignotant entre dans un état pour lequel le feu arrière gauche clignote. Quand la méthode ClignoteDroite est appelée, le clignotant entre dans un état pour lequel le feu arrière droit clignote. Dans ces deux états de clignotement, le feu droit ou gauche est éteint et le clignotant retourne à son état d'attente de commande lorsque la méthode Rappel est appelée.

Figure 2 Classe de la Question 4



Dessinez le diagramme d'états de la classe Clignotant.

1. widget: windows gadgets