ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

Examen ILO

14 décembre 2009

Durée : 1h30

Documents autorisés : Polycopiés de cours, TDs et TPs. Pas de calculatrice

Exercice 1 : Constructeurs (3 points)

a) Qu'est ce qu'un constructeur ? (1 point)

b) Les constructeurs n'indiquent pas leur type de retour : Quel est t'il ? (1 point)

c) En supposant que la variable `c1` de type `CompteBanque` est correctement initialisée, quelles sont les interprétations de l'instruction `CompteBanque c3 = c1;` en Java et en C++ ? (1 points)

Exercice 2 : Polymorphisme et Héritage (5 points)

a) En Java et en C++ on utilise le terme « polymorphisme » pour décrire deux mécanismes distincts : quels sont-ils ? (2 points)

b) Pourquoi est t'il important en C++ pour un opérateur + de renvoyer une référence vers l'objet qui déclenche cet opérateur (vers lui-même donc) ? (1 point)
Exemple : `MonObjet & operator+(const double valeurAjout);`

c) Expliquez la notion de lien dynamique (sans utiliser la définition du cours), utilisez des exemples au besoin. (2 points)

Exercice 3 : Gestion memoire (4 points)

a) Quelle est la durée de vie d'une variable (en Java ou en C++) ? (1 point)

b) Si une variable pointe vers une instance d'un objet (sous forme de référence en Java ou de pointeur en C++) que se passe-t-il lorsque cette variable disparaît en Java et en C++ (1 point).

c) Pour approfondir la question b) présentez les mécanismes de création et de destruction d'objets en Java et en C++ (2 points).

Exercice 4 : Conception (8 points)

On souhaite concevoir une hiérarchie d'objet pour gérer les mesures effectuées dans une station météo. Pour ce faire on peut représenter les capteurs comme des objets informatiques qui collectent les mesures faites par des sondes et les affichent à l'utilisateur (les capteurs informatiques sont donc considérés ici comme des afficheurs de mesure plutôt que comme le dispositif matériel qui effectue la mesure).

Capteurs

Les capteurs récupèrent une mesure d'une grandeur météorologique sur une sonde. Chaque capteur est caractérisé par un nom (thermomètre, baromètre, hygromètre, anémomètre, etc.) la valeur actuellement mesurée, l'unité dans laquelle cette valeur est mesurée ainsi que la sonde qui a fourni la mesure. Les capteurs possèdent néanmoins une liste d'unités dans lesquelles peut être exprimée la mesure : par exemple une mesure de température peut être faite en degrés Celsius mais exprimée (affichée ou transmise) en degrés Fahrenheit ou en degrés Kelvin.

Certains capteurs ont une plage de fonctionnement : c'est-à-dire que les mesures qu'ils récupèrent ne sont valides que dans une certaine plage (par exemple un thermomètre normal indique une valeur fiable si la température réelle est comprise entre -40°C et +50°C). Ces capteurs ont donc des bornes min et max au-delà

desquelles la mesure n'est plus fiable, les valeurs affichées par un tel capteur sont donc bornées par leur plage de fonctionnement.

Certains capteurs doivent pouvoir fournir à tout moment la mesure la plus basse effectuée ainsi que la mesure la plus haute (indépendamment de la notion de bornes de fonctionnement). Pour chaque mesure valeurs enregistrées min et max sont donc mises à jour. On doit aussi pouvoir remettre à zéro les valeurs enregistrées min et max lorsqu'on le souhaite.

Enfin certains capteurs doivent pouvoir calculer une tendance de mesure d'après les n dernières mesures faites et exprimer si celle-ci est à la hausse ou à la baisse.

Sondes

Les objets de type Sonde récupèrent les mesures sur les capteurs physiques (à ne pas confondre avec nos capteurs logiciels) à intervalles réguliers mais doivent être capable de fournir la dernière mesure effectuée à tout moment au capteur qui la demande (les sondes et les capteurs fonctionnent donc de manière asynchrone). Les sondes fournissent la mesure dans une unité de base qui pourra être convertie en une autre unité par les capteurs.

Unités

Une unité peut être décrite par une chaine représentant l'unité (par exemple « km/h » ou bien « $m \cdot s^{-1}$ ») et par un format qui permet de formater la valeur affichée (un angle en degré sera affiché sans décimales alors qu'un angle en radians nécessitera des décimales (120° ⇔ 2,094 radians)).

On distinguera les unités de bases (celles dans lesquelles sont effectuées les mesures) et les unités dérivées qui permettent d'exprimer une mesure dans une autre unité (par exemple une température mesurée en degrés Celsius (20 °C) peut être convertie en degrés Fahrenheit (68 °F).

Par ailleurs, on peut aussi distinguer les unités numériques et les unités symboliques. On peut ainsi exprimer la direction du vent en degrés (120°) ou bien en directions cardinales (Sud-Est). Néanmoins on considèrera que les sondes fournissent des mesures numériques.

Travail demandé

Concevez le diagramme de classes UML pour représenter les capteurs suivants : Girouette, Anémomètre, Hygromètre, Baromètre et Thermomètre en créant les classes pour les capteurs, les unités et les sondes avec les caractéristiques suivantes :

Capteur	*Grandeur*	*Unité de base*	*Plage de fonctionnement*	*Enregistrement min et max*	*Calcul tendance*	*Unités dérivées*
Girouette	Direction du vent	radians	-	non	non	• degrés, • Directions cardinale (Nord, Nord-Est, Est, Sud-Est, Sud, Sud-Ouest, Ouest, Nord-Ouest)
Anémomètre	Vitesse du vent	m/s	1 - 100	oui	non	• km/h, • Nœuds, • Miles/h, • Echelle Beaufort (Calme, Très légère brise, Légère brise, Petite brise, Jolie brise, Bonne brise, Vent frais, Grand frais, Coup de vent, Fort coup de vent, Tempête, Violente tempête, Ouragan)
Hygromètre	Humidité	%	0 - 100	oui	non	-
Baromètre	Pression	hPa (HectoPascals)	800 - 1100	oui	oui	• mm Hg (mm de mercure),
Thermomètre	Température	°C	-40 - +50	oui	oui	• °F (Fahrenheit) • °K (Kelvin)

A savoir :

- L'héritage multiple n'existe pas en Java, néanmoins une classe peut implémenter plusieurs interfaces; on parle alors d'héritage multiple de comportements.
- Les membres publics sont précédés de : +
- Les membres protégés sont précédés de : #
- Les membres privés sont précédés de : -
- Les relations d'héritage sont indiquées par les flèches en traits pleins.
- Les relations d'implémentation sont indiquées par les flèches en traits pointillés.
- Les membres ou les classes concrèt(e)s sont indiqués en texte droit.
- Les membres ou les classes abstrait(e)s sont indiqués en texte italique.
- On précisera pour les classes abstraites s'il s'agit d'interfaces ou de classes abstraites.
- Les compositions sont indiquées par un losange plein : par exemple la classe AB possède une instance de la classe « Element » et elle est responsable de la création et de la destruction de cette instance.
- Les agrégations sont indiquées par un losange vide : par exemple la classe ACD possède une collection d'instances de la classe « Element », mais elle n'est pas responsable de l'instanciation de cette collection (on supposera qu'elle lui est donnée par la méthode addElements).

Exemple :