

## Programmation par objets 2 - FLIN 606 - 10 mai 2011

2h - Tous documents autorisés

## 1 Station de mesure et capteurs

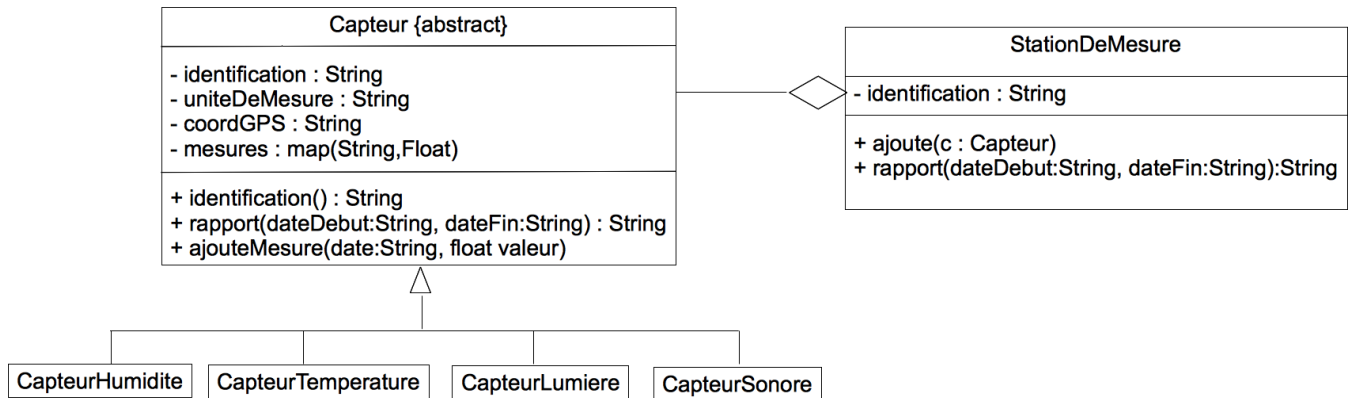


FIGURE 1 – Une station de mesure se compose de capteurs.

La figure 1 présente un modèle décrivant de manière succincte des stations de mesures. Une station de mesure est un dispositif composé d'un ou plusieurs capteurs proches. Ces capteurs peuvent mesurer le degré d'humidité, la température, la lumière ou le son. Un capteur dispose d'une identification, utilise une unité de mesure particulière, se trouve à un point localisé par des coordonnées GPS (ici codées dans une seule chaîne pour simplifier) et stocke pour chaque date (codée également pour des raisons de simplicité dans une string de format AAAAMMJJHHMnMnSS) la valeur mesurée. Deux méthodes sont disponibles sur les capteurs : une méthode **identification** qui retourne l'identification d'un capteur sous forme d'une chaîne et une méthode **rapport** qui retourne une chaîne contenant les résultats des mesures entre deux dates données. Nous supposons dans tout le sujet que les accesseurs existent pour les attributs, que ce soit en UML ou en C++, sans les écrire.

**Question 1** *Ecrivez l'interface (fichier header, d'extension .h) qui vous semble la plus appropriée pour la classe **Capteur**.*

**Question 2** *Ecrivez l'interface (fichier header, d'extension .h) qui vous semble la plus appropriée pour la classe **StationDeMesure**.*

**Question 3** *Ecrivez le/les constructeur(s) et le destructeur appropriés pour la classe **StationDeMesure**. Expliquez la solution choisie pour l'allocation/désallocation des objets de type capteurs et station qui justifie vos choix.*

**Question 4** *Ecrivez la méthode **ajoute** de la classe **StationDeMesure**. Elle doit signaler une exception (que vous définirez) lorsque l'on cherche à ajouter un objet capteur qui est déjà dans la station de mesure.*

**Question 5** *Ecrivez la méthode **rapport** de la classe **StationDeMesure**. Elle doit retourner une entête identifiant la station, puis pour chaque capteur, son identification, ses coordonnées GPS et les résultats des mesures entre les deux dates passées en paramètre.*

## 2 Réflexion autour de la généricité paramétrique

**Question 6** *Peut-on mélanger des capteurs de toutes sortes dans une station de mesure avec votre définition actuelle ? Et pourquoi ?*

**Question 7** *On désire à présent créer une station de mesure sonore (qui ne contient que des capteurs sonores). La solution que l'on vous demande d'explorer consiste à créer une classe `StationDeMesureSonore` comme sous-classe de `StationDeMesure`. Ecrivez la classe et discutez de la solution.*

**Question 8** *On veut de manière plus générale, créer des stations de mesure homogènes, n'intégrant qu'une catégorie de capteurs. Mettez en place une solution utilisant la généricité paramétrique (templates en C++).*

## 3 Héritage multiple sur les capteurs

On ajoute maintenant quelques classes de capteurs :

- une classe `CapteurHT` représentant les capteurs mesurant l'humidité et la température, sous-classe de `CapteurHumidite` puis de `CapteurTemperature`
- une classe `CapteurUniversel` représentant les capteurs mesurant les quatre paramètres, sous-classe de `CapteurHT` puis de `CapteurLumiere` puis de `CapteurSonore`.

**Question 9** *Pour le cas de l'héritage virtuel :*

- Écrivez les entêtes des classes (pas les attributs, ni les entêtes de méthodes)
- Indiquez dans quel ordre seront appelés les constructeurs pour une instance de `CapteurUniversel`.
- Dessinez un schéma mémoire d'une instance de `CapteurHT`.
- Discutez de la conception, est-elle judicieuse ? Imaginez une solution alternative à l'héritage multiple pour ce problème.

**Question 10** *Reprenez la question précédente dans le cas de l'héritage répété.*