



LE LANGAGE C++

ESILV S6 ∞ 2013-2014

Devoir maison

「 Parc informatique 」

- Ce travail est à réaliser **en binôme**.
- Il est à rendre au plus tard à la fin de votre prochaine et dernière séance de TP.
 - Vous fournirez vos **fichiers sources (.h et .cpp)**, soit par clé USB soit par e-mail à votre chargé de TP ;
 - au sein d'une archive au format **.zip** ;
 - en précisant votre groupe et vos noms et prénoms.
- L'ensemble des codes sera analysé par un système anti-plagiat. Un plagiat entraîne une note de 0 au module.
- Un programme ne compilant pas ou ne s'exécutant pas entraîne une note de 0 à ce contrôle continu.

L'objectif ici est de montrer votre habilité à modéliser « proprement » selon le paradigme orienté objet et à programmer un C++ avec une « bonne gestion de la mémoire ».

Exercice

Un **ordinateur** est constitué d'un ensemble de **composants**, qui peuvent être :

- un **processeur**, caractérisé par une fréquence (en giga-Hertz) ;
- une ou plusieurs barrettes de **mémoire**, caractérisée par une capacité (*) ;
- zéro ou plusieurs **disques durs**, caractérisé par une capacité (*). Le cas particulier d'un disque dur **SSD** est qu'il est en plus caractérisé par un nombre de cycles d'écriture.

(*) Une capacité s'exprime soit en octets (o), soit en kilo-octets (Ko), soit en méga-octets (Mo), soit en giga-octets (Go), soit en téra-octets (To) ; avec $1000\text{ o} = 1\text{ Ko}$, $1000\text{ Ko} = 1\text{ Mo}$, $1000\text{ Mo} = 1\text{ Go}$ et $1000\text{ Go} = 1\text{ To}$.

Il est donc possible d'**ajouter** (via l'opérateur +=) des composants à un ordinateur, sachant que :

- il ne peut y avoir qu'un processeur
- il ne peut y avoir au maximum que 4 barrettes de mémoire

- un ordinateur ne peut pas **démarrer** sans processeur ou sans mémoire, cela lèvera une exception de type **MissingComponentException**.

Étape 1 Codez en C++ les classes (et leurs attributs et méthodes éventuels) nécessaires pour modéliser cet exercice.

De plus,

- chaque classe disposera d'une méthode **to_string** offrant une représentation « libre » sous forme de *string* de l'état de l'objet ;
- l'opérateur << sera surchargé pour la classe **Ordinateur**, faisant appel à sa méthode **to_string**
- le démarrage de l'ordinateur affichera sur la sortie standard (**std::cout**) le contenu de l'ordinateur.

Étape 2 La classe **ParcInformatique** permet de regrouper un ensemble d'ordinateurs.

Cette classe offre trois méthodes :

- **chargement** qui à partir d'un fichier texte formaté comme ci-dessous permet de « remplir » le parc.

```
1 ordinateur;1;2;2;3
2 1;processeur;1.7
3 2;memoire;2;Go
4 3;ssd;500;Go;10000
5 ordinateur;1;1;4;5;5
6 4;memoire;4;Go
7 5;disquedur;1;To
```

- un ligne commençant par **ordinateur** signifie qu'il s'agit d'un nouvel ordinateur pour le parc. Les nombres suivants (séparés par ';') correspondent aux identifiants dans le fichier des composants qui composent cet ordinateur.
- un ligne commençant par nombre désigne un composant, identifié par ce nombre, dont le type est le second élément de la ligne. Les éléments suivants caractérisent ce composant.

- **démarrer_tout** qui démarre tous les ordinateurs du parc.
- **rechercher** qui recherche tous les ordinateurs selon une capacité de mémoire donnée (toutes barrettes confondues pour un même ordinateur). Cette méthode retourne donc un accès aux ordinateurs de parc correspondant au critère de recherche.

Étape 3 Mettez en situation un parc informatique dans le programme principal.