

TP n5 : Encapsulation

Exercice 1 : Temps

Soit la classe `Temps` constituée des attributs `sec`, `min`, `heure`, `jour`, `mois`, et `an` (correspondants aux secondes, minutes, heures, jours, mois et années), de constructeurs, de méthodes d'accès et des méthodes suivantes :

- `void modifieJMA(int, int, int)` : modifiant la valeur des attributs
- `void modifieHMS(int, int, int)` : modifiant la valeur des attributs
- `void ajouteA()` : ajoutant 1 à l'attribut `an`
- `void ajouteM()` : ajoutant 1 à l'attribut `mois`
- `void ajouteJ()` : ajoutant 1 à l'attribut `jour`
- `void ajouteH()` : ajoutant 1 à l'attribut `heure`
- `void ajouteMin()` : ajoutant 1 à l'attribut `min`
- `void ticTac()` et `void ticTac(int n)` : ajoutant 1 ou `n` à l'attribut `sec`. Faites attention à ne pas mettre plus de 60 secondes (au besoin, rajouter une minute...).
- `void modifieTemps(int, int, int, int, int, int)` : modifiant les secondes, les minutes, l'heure, le jour, le mois et l'année
- `void afficheTemps()` : affichant les valeurs des attributs

Écrire la classe `Temps` en séparant interfaces et implémentations (c'est-à-dire faire un `.h` et un `.cpp`). Rajouter éventuellement les méthodes et attributs qui vous semblent nécessaires. Respecter les règles d'encapsulation en déclarant public **un minimum de choses** (les méthodes utilisées dans le main) et en contrôlant au maximum les possibilités d'erreur d'utilisation de la classe au sein de la classe (par exemple mettre une date négative).

Écrire la fonction `main` réalisant les instructions suivantes :

- création une instance de `Temps` avec les valeurs d'attributs de votre choix
- affichage du temps
- appel à la méthode `modifieTemps` avec les arguments (58, 59, 23, 31, 12, 1999)
- 3 appels successifs à la méthode `tic`
- affichage du temps
- appel à la méthode `modifieTemps` avec les arguments (58, 59, 23, 28, 2, 2000)
- appel à la méthode `ticTac` avec la valeur 5 passée en argument
- affichage du temps

Rappel : une année est bissextile si elle est divisible par 4 mais non divisible par 100 (sauf si elle est divisible par 400). 1900 n'était pas bissextile, 2000 oui.

Exercice 2 : Variable et méthode de classe

Vous ajouterez une variable de classe (ou `static`) à la classe `Temps` qui permet de savoir combien d'horloges ont été déclarées dans le programme. Vous modifierez donc le constructeur pour qu'à chaque création d'objet, on incrémente ce compteur, et à chaque destruction, on décrémente ce compteur. Faites aussi la fonction de classe (ou `static`) `afficheCpt` pour pouvoir afficher la valeur du compteur.

Modifiez le `main` pour créer 3 variables de type `Temps` puis affichez la valeur du compteur.

```
void main(){
    Temps t1,t2,t3 ;
    Temps::afficheCpt() ;
}
```

Le `main` précédent doit afficher 3 à l'écran...