

TD 2 - Amitié et surcharge d'opérateurs

Programmation Orientée Objet

Objectif

- Comprendre le principe des fonctionnalités amies
- Comprendre ce qu'est la surcharge des opérateurs ;

1 Fonctions amies

Créer une classe **Avion** ayant pour données privées le fabricant, le type, l'altitude et le cap de l'avion. Créer une fonction *non membre* **afficheFiche**, amie de la classe **Avion**, permettant d'afficher les informations caractéristiques d'un avion sous le format ci-dessous :

```
**** AVION ****
* Airbus
* A320
* Alt : 10000
* Cap : 345
*****
```

2 Classes amies et surcharge

Créez une classe **TourDeContrôle**. Cette classe contiendra un tableau d'**Avion** de taille fixe, le nombre d'avions enregistrées dans ce tableau. Cette classe sera une classe amie de la classe **Avion**.

- 1) Que faut-il faire pour rendre **TourDeContrôle** amie de la classe **Avion** ?
- 3) Créez une fonction membre **AjouterTabAvionNormal** ajoutant un avion au tableau en utilisant les accesseurs de la classe **Avion**.
- 4) Créez une fonction membre **AjouterAvionViaAmie** ajoutant une personne au tableau en utilisant les accès directs à la classe **Avion**.
- 5) Comment faudrait-il faire pour ajouter un avion sans passer par les accesseurs ni par un accès direct ? Proposez une solution.
- 6) Créez un programme principal manipulant la classe **TourDeContrôle**. Pour cela créez 3 avions différents, ajoutez les en utilisant les différentes méthodes, et affichez les avions du tableau à l'aide de la méthode **affiche()**.
- 7) Surchargez la méthode **cout** pour afficher directement les éléments d'un avion.