

TD 4 : Héritage

POO Avancée

Objectifs

Prendre la mesure de l'héritage et des nouveaux concepts vus en cours Héritage et Polymorphisme.

Les mots clefs :, base, new, virtual, override, sealed et abstract doivent être assimilés

Exercice 1

Un salarié est une personne qui a un salaire, un email professionnel et une date d'entrée dans la compagnie

On définit une Personne par un nom, un prénom, une date de naissance (DateTime), un sexe (char), une adresse (string) et un numéro de téléphone

- Décrire le diagramme UML
- Créer la classe Personne
 - a. Avec les attributs privés
 - b. Définir le constructeur, la propriété sur le nom en lecture,
 - c. Définir les méthodes suivantes :
 - i. qui calcule l'âge de la personne à partir de l'année courante *int Age()*
 - ii. la méthode *string ToString()* héritée de la classe Object (override)
 - iii. la méthode *string Coordonnees()* qui retourne le numéro de téléphone sous forme de string

Créer dans le Main, 2 instances de la classe Personne et affichez leur âge et leur identité

- Créer la classe Salarie qui hérite de la classe Personne
 - a. Définir le constructeur
 - b. Changez l'accessibilité des attributs de la classe Personne en « protected ». Adaptez le constructeur
 - c. Définir les méthodes suivantes
 - i. La méthode *string ToString()* héritée de la classe Object
 - ii. La méthode qui calcule l'ancienneté du salarié *int Anciennete()*
 - iii. La méthode *string Coordonnees()* qui retourne la chaine de caractères compilant le numéro de téléphone et l'email professionnel (utiliser le mot clef new)

Créer dans le Main, 2 instances de la classe Salarie et affichez leur identité complète ainsi que le nombre d'années d'ancienneté et ses coordonnées complètes

- Créer dans le main
 - a. Salarie p = new Personne(...);

- i. Que constatez-vous ? Pourquoi ?
- b. `Personne p = new Salarie(...)`
 - i. Que constatez-vous ? Pourquoi ?
- c. Dans le cas b), appliquez la méthode `Coordonnees`
 - i. Que constatez-vous ? Pourquoi ?
- d. Transformez la méthode `Coordonnees` de la classe `Personne` en virtual et celle de la classe `Salarie` en override, appliquez la méthode `Coordonnees`
 - i. Que constatez-vous et pourquoi ?

Exercice 2

On définit une figure par une aire, un périmètre et un point d'origine. En tant que telle, nous ne créerons jamais d'instance de la classe figure.

On définit un rectangle comme une figure, le point d'origine étant le coin haut droit.

On définit un cercle comme une figure dont le point d'origine est le centre.

Le rectangle et le cercle ont des aire et périmètre calculés de façon particulière.

- Définir le diagramme UML
- Ecrire les classes de telle sorte que nous puissions calculer l'aire et le périmètre d'un rectangle et d'un cercle dans le main

Exercice 3

Une petite compagnie aérienne vous demande un logiciel pour gérer sa flotte d'avions. Une première étape repose sur la modélisation objet de ses avions et de son personnel. Chaque avion est repéré par un identifiant unique (X5743, par exemple) lié à l'avion, une référence liée à la classe d'avion (Airbus A340 par exemple), un rayon d'action (en kilomètre) et la longueur de piste nécessaire (identique pour son envol et son atterrissage). Deux grands types d'avions existent : les avions cargos et les avions pour voyageurs. Pour les avions cargos, il faut alors définir la charge utile en poids et en m3 et le nombre de pilotes nécessaires. Les avions transportant des voyageurs définissent aussi le nombre de pilotes mais aussi les informations sur la cabine : nombre de sièges, taille de l'équipage (en dehors des pilotes). Il existe deux types d'avions transportant des voyageurs : ceux dits d'affaires destinés à la location privée et ceux dits de ligne destinés au transport de voyageurs avec billets. Une information de niveau de luxe (une chaîne) est ajoutée pour les avions d'affaires. Le nombre de places de classe normale et de classe premium est noté pour les avions de ligne.

Chaque personnel est repéré par son nom, son prénom et un identifiant unique (un numéro). Il existe deux types de personnels : les navigants et le personnel au sol. Les navigants sont soit des pilotes soit des stewards. Les pilotes possèdent un numéro de licence ainsi que la liste des avions qu'ils peuvent piloter. Les stewards possèdent la liste des avions pour lesquels ils sont formés ainsi que la liste des

langues qu'ils maîtrisent. Une information sur le nombre de vols effectués est définie pour les pilotes et les stewarts. Chaque personnel au sol est attaché à un aéroport (repéré par son nom). Les personnels au sol sont divisés en trois sous-groupes : les mécaniciens, le personnel de piste et le personnel d'accueil. Les mécaniciens possèdent une liste d'avions sur lesquels ils peuvent intervenir. Le personnel de piste possède une liste d'habilitations (une habilitation est une chaîne). Le personnel d'accueil est en particulier défini par son affectation (une chaîne) et la liste de ses langues maîtrisées.

Aussi, certains avions équipés de balises GPS sont localisables : les cargos. La localisation est alors accessible sous forme d'une fonction retournant une chaîne "coordonnées GPS".

Exercice 1

Proposez une modélisation UML afin de représenter les avions et les personnels. Proposez une classe Compagnie intégrant les avions et les personnels

Exercice 2

Proposez le code complet C# de la classe Cargo

Exercice 3

Justifiez, le cas échéant, les classes abstraites et des collections.

Proposez une méthode permettant d'afficher la position de tous les avions localisés.