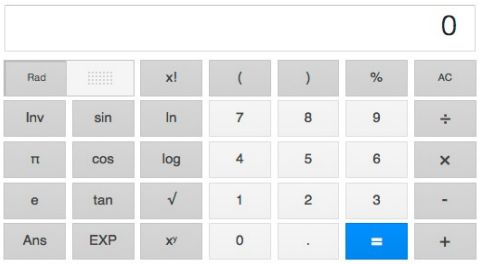
Université Grenoble Alpes
Licence 2 MIASHS
Inff3 – Algorithmique et programmation par objets
Jérôme David-
2024-2025

TD1 : Introduction aux concepts de la programmation par objets

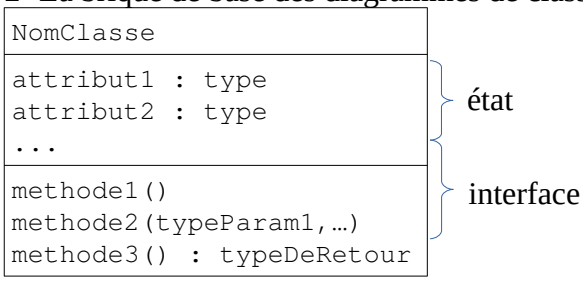
Le but de ce TD est de vous initier à la conception par objets. Pour cela, on s'appuiera sur des diagrammes de classe, notation graphique issue d'un langage de modélisation et conception objet appelé UML (*Unified Modeling Language*).

1. Objets, classes, attributs et méthodes



1- Identifiez les classes d'**objets graphiques (visuels)** et leurs instances sur cette interface de calculatrice. Imaginez les attributs et des méthodes que ces objets possèdent.

2- La brique de base des diagrammes de classe UML est le pictogramme représentant une classe :



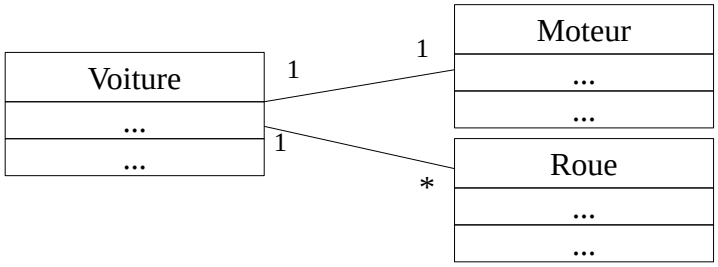
Pour chaque attribut, on donnera un nom et son type. Les types prédéfinis sont *entier*, *réel*, *booléen* et *chaîne* (de caractères). Une méthode a un nom et peut avoir éventuellement des paramètres. Dans ce cas, on indiquera le type des paramètres. Finalement, une méthode peut retourner un résultat dont le type sera indiqué.

a) Modélisez le fonctionnement d'une classe « bouteille » en suivant la description suivante. Une bouteille a une capacité (volume maximal contenu), un volume actuel, elle peut être fermée ou ouverte. On peut l'ouvrir, la fermer, ajouter ou enlever du liquide, la vider, la remplir, et calculer le niveau restant.

b) Une chaîne de caractères peut être vue comme une séquence (une suite ordonnée) de caractères. D'après votre expérience, donnez au moins 5 méthodes (avec leurs arguments et types de retour) qu'une chaîne de caractères peut proposer.

2. Relation entre classes

Les relations entre classes sont représentées de la manière suivante dans un diagramme de classe :

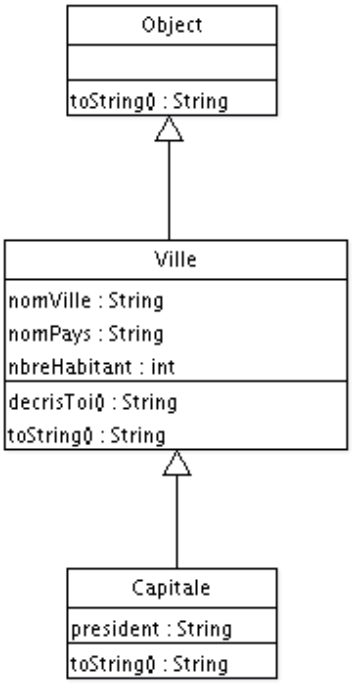


« une voiture est composée d'un moteur et de plusieurs roues » (*=plusieurs).
« une roue est associée une et une seule voiture ».

Exercice :
Vous voulez réaliser un gestionnaire d'albums photos. Vos photos peuvent être annotées (date, ville de la prise de vue, personnes présentes sur la photo). Les personnes peuvent être catégorisées (amis, famille, etc). Les localisations peuvent être également hiérarchisées (villes dans des régions, et régions dans pays).
Proposez un diagramme de classe modélisant les données (les attributs mais pas les méthodes) de cette application.

3. Héritage, redéfinition et polymorphisme

L'héritage est représenté de la manière suivante dans un diagramme de classe :



Les liens composés d'un triangle et d'un trait indiquent que la classe Ville est une sous-classe de (étend) la classe Objet et que Capitale est une sous-classe de Ville. Par transitivité, Capitale est une sous-classe d'Objet.
Par héritage, tous les attributs et méthodes sont hérités dans les sous-classes : une capitale possède les attributs nomVille, nomPays et nbreHabitants et les méthodes decrisToi(), toString().
Dans une sous-classe, on ne répète le nom d'une méthode héritée que lorsque la méthode en question doit être redéfinie (son implémentation change). Dans l'exemple, la méthode decrisToi() est héritée telle quelle dans la classe Capitale. Cependant, la méthode toString() de la classe Object est redéfinie dans les classes Ville et Capitale.

Une boutique en ligne vous demande de modéliser son catalogue avec des objets. Cette boutique vend principalement des livres, mais propose également de nombreux autres articles tels que des jeux vidéos, des consoles, etc. L'entreprise s'intéresse principalement au prix des articles, leur quantité en stock et au calcul des frais de port qui dépendent du poids et des dimensions de l'objet. Afin générer les pages web des articles de sa boutique elle a besoin d'informations spécifiques (qui dépendent des types d'objets).

Lors de la discussion avec le commerçant, vous avez noté en vrac les notions suivantes :
`article, prix, livre, epaisseur, console, auteur, nombrePages, calculerFraisPort, genererPageWeb, annee, marque, poids, titre, modele, jeuVideo, qteEnStock, commander, dimension, largeur, hauteur, editeur.`

1- Identifiez dans cette liste les termes représentant les classes, les attributs et les méthodes.
2- Organisez les classes en hiérarchie d'héritage et ventilez les attributs et méthodes dans ces classes. Identifier pour chaque méthode ses éventuels paramètres et types de retour.
3- Identifiez les méthodes qui seront redéfinies.
(4- Certains attributs sont déclarés dans différentes classes indépendamment (i.e. ils ne sont pas déclarés dans une super-classe commune). Proposez une solution permettant de factoriser ces attributs. Les classes que vous venez d'ajouter peuvent-elles avoir des instances ?)