

TP 01

Introduction

Ce TP vise à vous initier à Java, en abordant à la fois sa syntaxe et les bases de la programmation orientée objet. Java est un langage fortement typé, utilisé dans de nombreux domaines, du développement d'applications desktop aux systèmes embarqués, en passant par le développement web et mobile. Ce premier TP est structuré en 3 parties :

- L'exercice d'introduction, qui permet de pratiquer la syntaxe Java avec des cas simples sans introduire les objets.
- L'exercice 1 qui introduit progressivement les classes, les objets et les principes fondamentaux de la POO.
- L'exercice 2 qui servira à mettre en pratique le tout à travers un cas pratique.

Pour démarrer

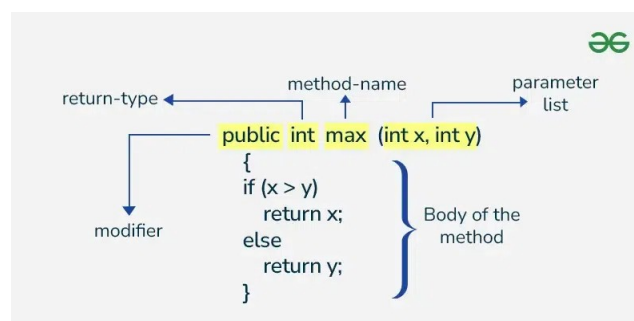
Avant de commencer à coder, je vous invite fortement à voir les prochaines vidéos d'introduction à java. Ces notions peuvent vous paraître floues au départ, mais celles-ci seront abordées durant les cours de la semaine.

1. **Qu'est-ce que Java ?** : [Vidéo d'introduction à Java](#)
2. **Installation de IntelliJ IDEA** : Pour coder en Java, nous utiliserons l'IDE IntelliJ IDEA, disponible via JetBrains.
3. **Synthèse de java en 3min pour comprendre les bases** : [Vidéo explicative](#)

En java, la syntaxe de déclaration d'une variable et d'une méthode est atypique de python.

Les variables doivent être tout le temps typées et nommées, mais pas forcément initialisées. Il existe plusieurs types de variables que l'on verra durant le semestre. Sur le lien suivant, vous aurez quelques détails pour mieux comprendre comment les variables en java fonctionnent, vous n'êtes pas obligés de tout lire (de préférence au moins jusqu'au chapitre des différents types compris), mais cela vous aidera à définir une bonne base pour votre apprentissage (ces notions seront aussi bien évidemment revues en cours) : <https://www.guru99.com/fr/java-variables.html>

La syntaxe pour la déclaration d'une méthode est la suivante :



Parfois, le terme “Static” sera aussi ajouté après le “public” afin de définir une méthode qui sera rattachée à une classe et utilisable par celle-ci (*un petit peu comme vous aviez vu en python pour faire appel aux méthode d'un module à travers le main “module.VotreFonction()”*)

Exercice d'introduction

Pour démarrer votre apprentissage en java, un premier exercice d'introduction est à réaliser et présent dans le fourni à l'intérieur du package “_IntroJava”. Les fichiers sont déjà créés, ainsi que les méthodes, il ne vous reste qu'à compléter les différentes fonctionnalités en suivant les consignes en commentaires (*Commencez par “ExerciceIntroduction.java”*).

L'idée de cet exercice est de vous familiariser avec les bases en java. Essayez d'expérimenter différentes façons de faire l'exercice sans viser une manière précise ou forcément la plus juste ou efficace, utilisez la logique que vous avez assimilé durant le premier semestre, la difficulté principale de ces fonctions réside dans votre ignorance à la syntax java, si vous ne savez pas comment coder une condition ou afficher une valeur, votre premier reflexe en tant qu'informaticien est d'utiliser le site suivant pour y trouver votre réponse :

[Google](#)

(*Ce n'est pas une blague...prenez VRAIMENT le réflexe de faire vos recherches quand vous êtes bloqué et pas de crier à l'aide à vos voisins de cours ou à l'assistant...*)

Exercice 1 : Programmation orientée objet

Introduction à la POO en Java

Maintenant que nous avons vu les bases de Java sans la POO, nous allons introduire les concepts d'objets et de classes.

Vidéo recommandée

- **Apprendre Java (objet, classe, héritage) :** [Lien vers la vidéo](#)

Cette vidéo (plus longue et dense que les premières) est un premier pas vers l'apprentissage de l'orienté objet en plus des différents cours que vous aurez les deux prochaines semaines. Je vous invite à regarder toute la vidéo, mais la partie sur l'héritage ne sera pas tout de suite traité dans ce cours (vous pouvez tout de même regarder cette partie, mais attention à ne pas trop vous enfoncer dans du contenu nouveau alors même que la base n'a pas été acquise). La partie instanceof est à oublier, vous pouvez l'ignorer.

Exercice : Modélisation d'une entreprise et d'un employé

L'objectif de cet exercice sera de vous familiariser avec la création d'objets java, et de mettre en place les bonnes pratiques le plus vite possible. En général, il est possible de déclarer des attributs sans avoir à définir l'accès (public ou private), mais je vous invite à prendre au plus tôt le réflexe de définir en privé tous les attributs d'une classe du domaine (C'est les objets métiers, comme une Personne, une Maison, etc...). Pour cet exercice, rien ne vous est fourni, c'est à vous de tout créer. Vous pouvez aussi rajouter une classe Main pour effectuer vos tests et vous familiariser avec la création d'instances et l'appel de fonction depuis un objet.

Classe Entreprise

L'objectif est de créer une classe « Entreprise » afin de mettre en pratique vos connaissances fraîchement acquises sur les classes. L'exercice est séparé en 3 parties :

- Les variables d'instances ou attributs
- Le constructeur
- Les méthodes de classe

Voici le détail de la classe représentant une entreprise. Une entreprise possède les champs suivants :

- nom
- adresse
- type (Publique, Privée, Semi-publique)
- nombreBâtiments
- domaine (Informatique, Médical, Logistique, etc.)
- nombreEmployes
- detenteurMajoritaire

Méthodes à implémenter :

1. Modifier le nombre de bâtiments par une valeur donnée
2. Modifier le nombre d'employés par une valeur donnée
3. Afficher les informations de l'entreprise de deux manières :

- Affichage complet :

Nom de l'entreprise, adresse
Entreprise "type" détenue par "detenteurMajoritaire" dans le domaine yyyyyy
L'entreprise possède X bâtiments pour Y employés

- Affichage partiel :

Nom de l'entreprise et adresse

4. Modifier le type de l'entreprise (si publique, le détenteur majoritaire est obligatoirement "L'État")

Exemples de sorties :

```
Agrofi, Chemin de l'usine 82b
Entreprise Privée détenue par Charles Agrofi dans le domaine Alimentaire
L'entreprise possède 3 bâtiments pour 42 employés
```

```
Après avoir mis le nombre de bâtiments à 5 via la méthode :
Agrofi, Chemin de l'usine 82b
Entreprise Privée détenue par Charles Agrofi dans le domaine Alimentaire
L'entreprise possède 5 bâtiments pour 42 employés
```

```
Après avoir passé le type de l'entreprise en « Publique » grâce à la méthode :
Agrofi, Chemin de l'usine 82b
Entreprise Publique détenue par L'Etat dans le domaine Alimentaire
L'entreprise possède 5 bâtiments pour 42 employés
```

Classe Employe

De plus, nous souhaiterions pouvoir modéliser un employé également. Un employé possède les attributs suivants :

- nom
- prenom
- adresse
- fonction
- anneeDansEntreprise
- salaireMensuel

Méthodes à implémenter :

1. Modifier l'adresse en cas de déménagement
2. Modifier le nombre d'années dans l'entreprise
3. Calculer le bonus en fonction de l'ancienneté :
 - Salaire mensuel + (annéeDansEntreprise % du salaire mensuel)
 - Exemple : pour un employé avec 7 ans d'ancienneté et un salaire de 10'000 CHF :
$$10'000 \text{ CHF} + 7\% \text{ de } 10'000 \text{ CHF} = 10'700 \text{ CHF}$$
4. Afficher les informations complètes de l'employé

Exemple d'affichage :

```
Dupont Jean, Chemin de la route 35. Fonction : Comptable
7 années dans l'entreprise. Salaire : 10700.0Chf par mois, 10000Chf sans bonus
```

Exercice 2 : Programme de gestion d'une bibliothèque

Ce deuxième exercice a pour but de concrétiser votre apprentissage des différentes notions à travers un cas pratique déjà vu le semestre précédant, mais adapté à Java. L'objectif principal est de comprendre l'utilisation du static, la création et l'appel de méthodes ainsi que la gestion des variables et d'objets java à travers différents fichiers. Pour cet exercice, un fichier vous est fourni dans le package "Exercice2" sur lequel vous ajouterez les différentes fonctionnalités demandées.

Je vous invite à aller la voir la vidéo suivante afin d'assimiler concrètement la bonne utilisation des variables, la notion de référence ainsi que l'utilisation des variables dans une structure de donnée : [Lien de la vidéo](#)

Contexte

La HEG souhaite développer un logiciel de gestion de bibliothèque. Vous êtes chargé.e d'implémenter un module simple en Java permettant d'ajouter des livres, de gérer leur disponibilité et d'afficher les informations.

Données

Les données seront stockées dans une *ArrayList<Livre>*, où chaque élément est un objet représentant un livre avec trois attributs :

- id : un entier unique identifiant le livre
- titre : une chaîne de caractères représentant le titre du livre
- disponible : un booléen indiquant si le livre est disponible ou non

Cela permet une gestion plus simple des livres et facilite l'ajout, la suppression et la modification des données.

Vous devez ainsi créer la classe Livre afin de pouvoir l'utiliser correctement dans votre logiciel de Bibliothèque en ajoutant les bons attributs et méthodes.

La variable "livres" est une liste qui sera accessible dans tout le fichier

Nous allons coder ce programme dans un unique fichier Java en utilisant des méthodes statiques pour manipuler la liste. Une méthode main pour tester vos fonctions est fournie, dans laquelle vous trouverez la création de 3 livres et l'ajout de ceux-ci dans la liste.

Fonctions à implémenter

1. Ajouter un nouveau livre

- Nom : ajouterLivre
- Paramètres : l'id et le titre du livre à ajouter
- Retourne : un message indiquant que le livre a bien été ajouté

- Remarque : un nouveau livre est toujours disponible

2. Vérifier la disponibilité d'un livre

- Nom : livreDisponible
- Paramètres : l'id du livre concerné
- Retourne : un booléen indiquant la disponibilité du livre
- Remarque : un livre est disponible s'il existe dans la liste et que son champ disponibilites contient true

3. Modifier la disponibilité d'un livre

- Nom : modifierDisponibilite
- Paramètres : l'id du livre à modifier et la nouvelle disponibilité
- Retourne : un message indiquant que la liste a été mise à jour

4. Afficher la liste des livres

- Nom : afficherLivres
- Paramètres : aucun
- Retourne : rien
- Affiche chaque livre sur une ligne sous la forme :
[ID] - Titre – Disponible/Indisponible

Exemple d'affichage :

```
Le livre avec l'id 4 a été ajouté avec succès.  
Le livre avec l'id 2 est disponible: true  
La disponibilité du livre avec l'id 3 a été modifiée avec succès.  
[1] L'île mystérieuse - Disponible: true  
[2] Voyage au centre de la Terre - Disponible: true  
[3] De la Terre à la Lune - Disponible: true  
[4] Histoire d'un casse-noisette - Disponible: true
```