

# Typage & Classes

## Exercice 1 : Les débuts de l'informatique moderne : Charles Babbage, Ada Lovelace, Alan Turing & Co...

Finir cet exercice dont l'énoncé complet se trouve dans le TD précédent 😊 ....

## Exercice 2 : Un robot fortement typé ...

### Rappel du cahier des charges pour cette séance :

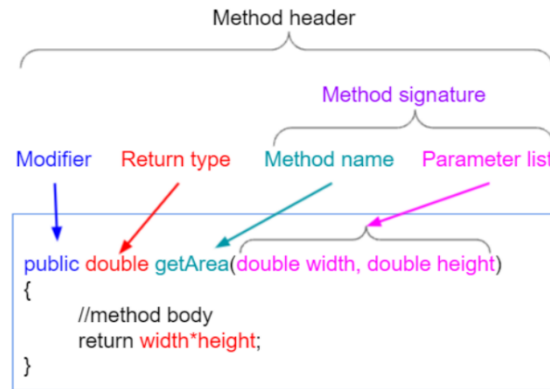
*Vous allez participer au développement du projet qui va piloter le prochain robot (rover) sur la lune.  
Pour simplifier le problème, on considère dans un premier temps que le robot se déplace sur une carte 2D. Le robot démarre toujours du centre de la carte c.-à-d. en (0,0) et il est dirigé vers le nord.  
Le robot (rover) doit pouvoir avancer, tourner à droite et gauche.*



*Proposez un diagramme de classes qui permet de modéliser ce problème.  
Vous dessinerez ce diagramme **à la main** sur une feuille libre où vous mentionnerez nom, prénom et groupe.*

1. Discutons ensemble des différentes propositions de **conception** que vous avez suggérées pour résoudre ce problème et proposons « en équipe » une première ébauche du **diagramme de classes**.
2. **L'implémentation** de cette conception se fera **en TP**, toutefois nous allons réfléchir ensemble avant aux points suivants :
  - ➔ 2.a Zoom sur les **attributs** :
    - **final** est un mot-clé de java, que signifie-t-il ? Indiquez à quel endroit il peut être utilisé dans le code et quel est alors son impact sur le code ?
    - Indiquez pour chaque attribut de votre modèle, si vous ferez apparaître le modificateur **final** lors de la déclaration de cet attribut. Justifiez bien sûr vos réponses 😊

Rappel sur la *signature* d'une méthode :



## → 2.b Zoom sur les constructeurs

- Au niveau de la déclaration, quelle est la différence au niveau du **header** entre un **constructeur** et une **méthode** (opération/service qui apporte un comportement métier à l'objet) ?
- En principe, **plusieurs constructeurs sont déclarés dans une classe** avec des paramètres différents pour donner plus de flexibilité à votre code et permettre d'instancier un objet de différentes manières selon les besoins c.-à-d. en passant des paramètres d'entrée différents (en nombre, en type,...)

**Proposez plusieurs signatures de constructeurs pour la classe Rover pour pouvoir instancier un robot de différentes façons.**

**Rappel** : en Java, si vous n'écrivez pas de constructeur dans une classe, le **constructeur par défaut** (sans paramètre d'entrée) est disponible. Mais à partir du moment où vous implémentez un constructeur dans une classe, le constructeur par défaut n'est plus disponible. Si vous souhaitez utiliser le constructeur par défaut, il faut alors l'écrire.

**Le constructeur par défaut fera donc également partie des constructeurs que vous devez proposer 😊**

- Implémentez les constructeurs proposés précédemment.
- **Utilisation des constructeurs** :  
Pour chaque constructeur implémenté précédemment, écrire une instruction qui permet de montrer comment **instancier un robot à l'aide de ce constructeur**.  
Vous pouvez par exemple appeler les robots (objets) : **viper, python, anaconda, ...**

## Travail à rendre pour la prochaine séance de TD :

Votre nouveau client souhaite vous confier le développement de différents jeux de cartes qui se jouent à partir d'un jeu de cartes à jouer de 52 cartes. Il vous détaillera plus tard le(s) jeux qu'il souhaite vous faire développer.



Pour gagner du temps, vous décidez de vous familiariser avec ce nouveau *contexte métier* (celui du jeu de cartes).

Pour commencer, vous décidez de vous focaliser sur **l'élément de base** de cette application : **la carte à jouer**, puisque **sans carte, pas de jeu à développer ! 😊**

Proposez un diagramme de classes pour modéliser une carte à jouer.

Ce diagramme devra être réalisé **à la main** et **en anglais**.

Conseil : Pour vous familiariser avec votre contexte métier et adopter la *bonne* terminologie métier, n'hésitez pas à utiliser votre moteur de recherche favori 😊