

Programmation Orientée Objet (OBJET)

Modèle mémoire en Java

1 - Théorie

Jean-Marie Normand – Bureau E211

jean-marie.normand@ec-nantes.fr



Modèle de gestion de la mémoire en Java (1)

- Simplification → 2 « types » de mémoire en Java :
 - **Tas** (« *Heap* »)
 - **Pile** (« *Stack* »)
- En vrai plus compliqué → suffisant pour ce dont on parle ici

Modèle de gestion de la mémoire en Java (2)

- **Tas** = zone mémoire où sont créés les **objets**, les **tableaux**, les **chaines de caractères**, etc.
 - Même pour les tableaux de types primitifs (**int**, **double**, **char**, **boolean**)
 - P. ex. `int [] array = new int[10];`
- Zone mémoire **commune** à (et **partagée** par) toutes les **fonctions**, **threads**, etc. d'un **programme**
 - Mémoire libérée par le *Garbage Collector* (GC)

Modèle de gestion de la mémoire en Java (2)

- **Pile** = permet de stocker **les variables locales**, les **paramètres**, les **valeurs de retour** de chaque méthode invoquée par un programme (en fait par un thread)
- Chaque méthode crée un « **environnement/ contexte** » local dans la pile pour y stocker ses informations :
 - Contient **uniquement** des variables de **type primitif** (int, double, char, etc.)
 - Ne peut **pas contenir d'objets** !
 - 1 pile par programme/thread !

Résumé

- 2 types de zones mémoire :
 - **Tas** : partagée
 - **Pile** : stocke des contextes locaux créés pour chaque méthode; unique par un thread/programme
- Seules des **données de type primitifs** et des **références à des objets** peuvent être stockées dans la **Pile**
- Tous les **objets** sont **forcément créés** dans le **Tas** et donc **partagés**.
- Java propose un mécanisme de gestion de la mémoire (*Garbage Collection*) → lorsqu'un **objet n'est plus accessible** par personne il est **détruit** !