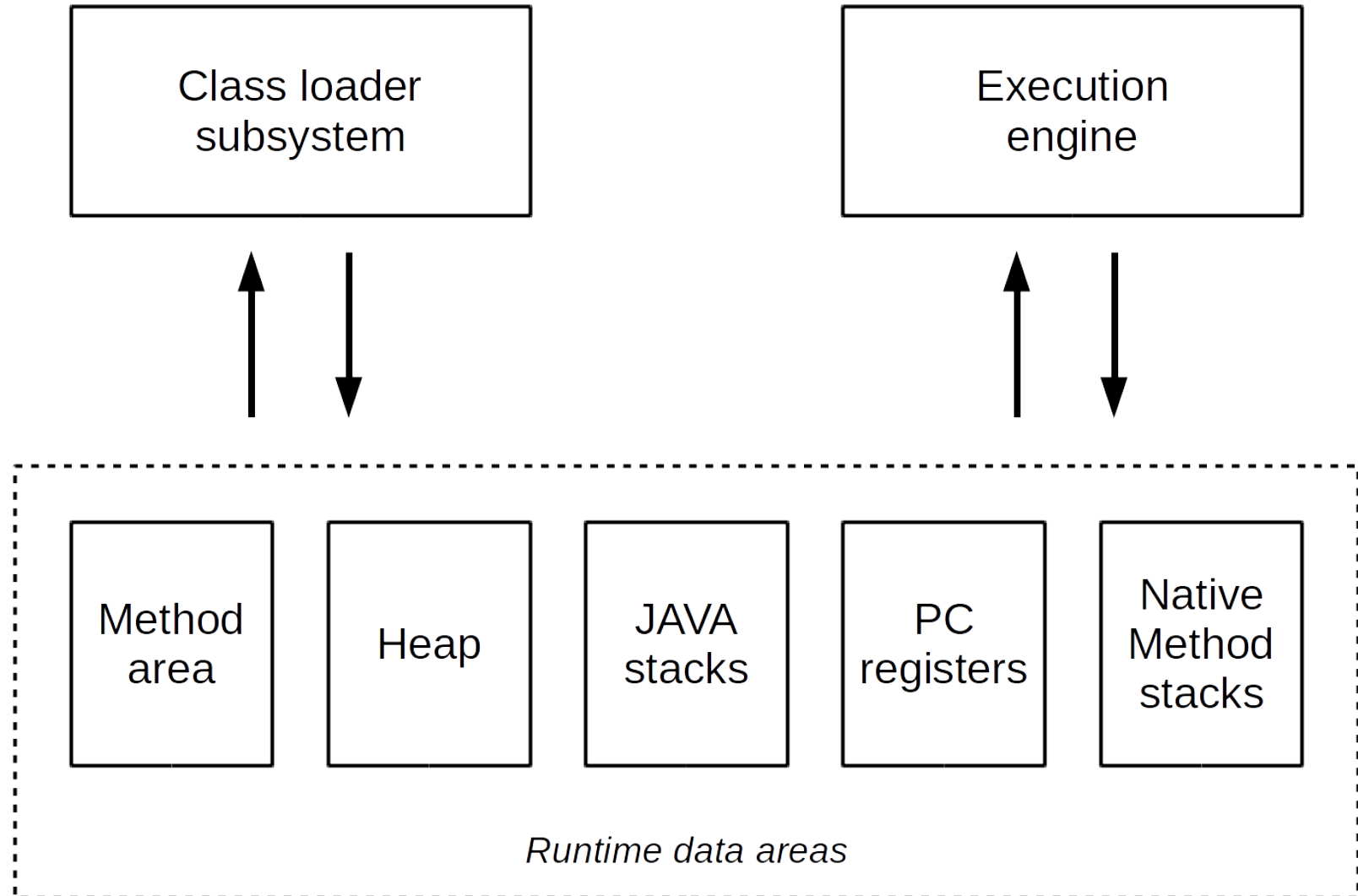


JVM, introspection, exécution

Leleux Laurent

2016 - 2017

JVM



JVM memory

- **Method Area** : classes + interfaces
- **Heap** : instances
- **JAVA stacks** : méthodes en cours d'exécution
- **PC registers** : registres de la JVM
- **Native method stacks** : méthodes natives en cours d'exécution

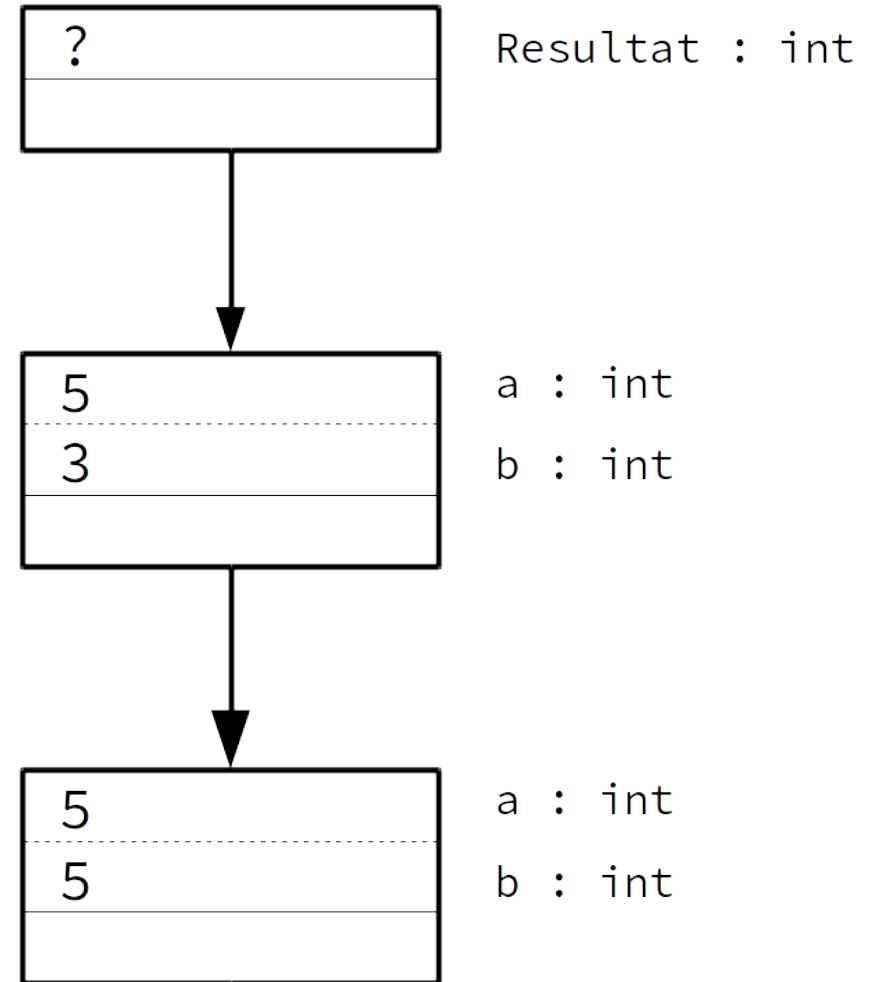
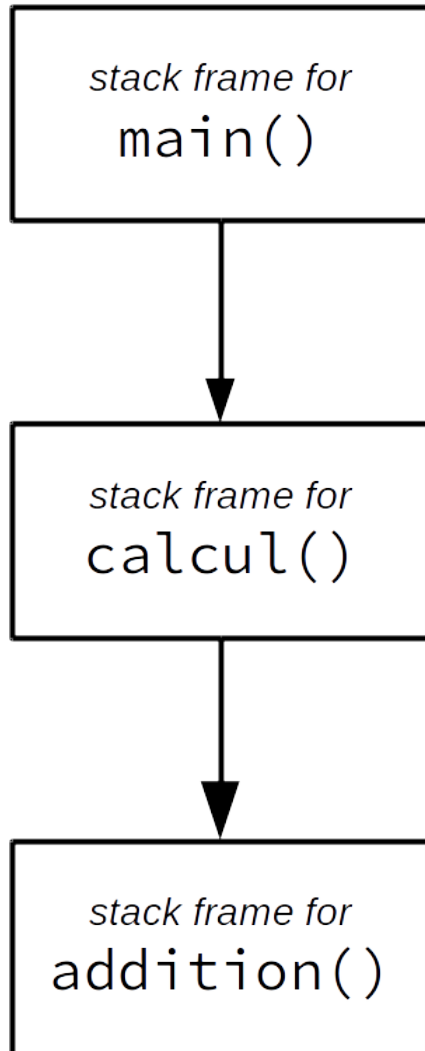
JAVA Stacks

- Découpée en « frames »
- Une par méthode
- Contient :
 - Variables locales / Paramètres (primitives ou références)
 - Zone de travail
- Détruite à la fin de la méthode

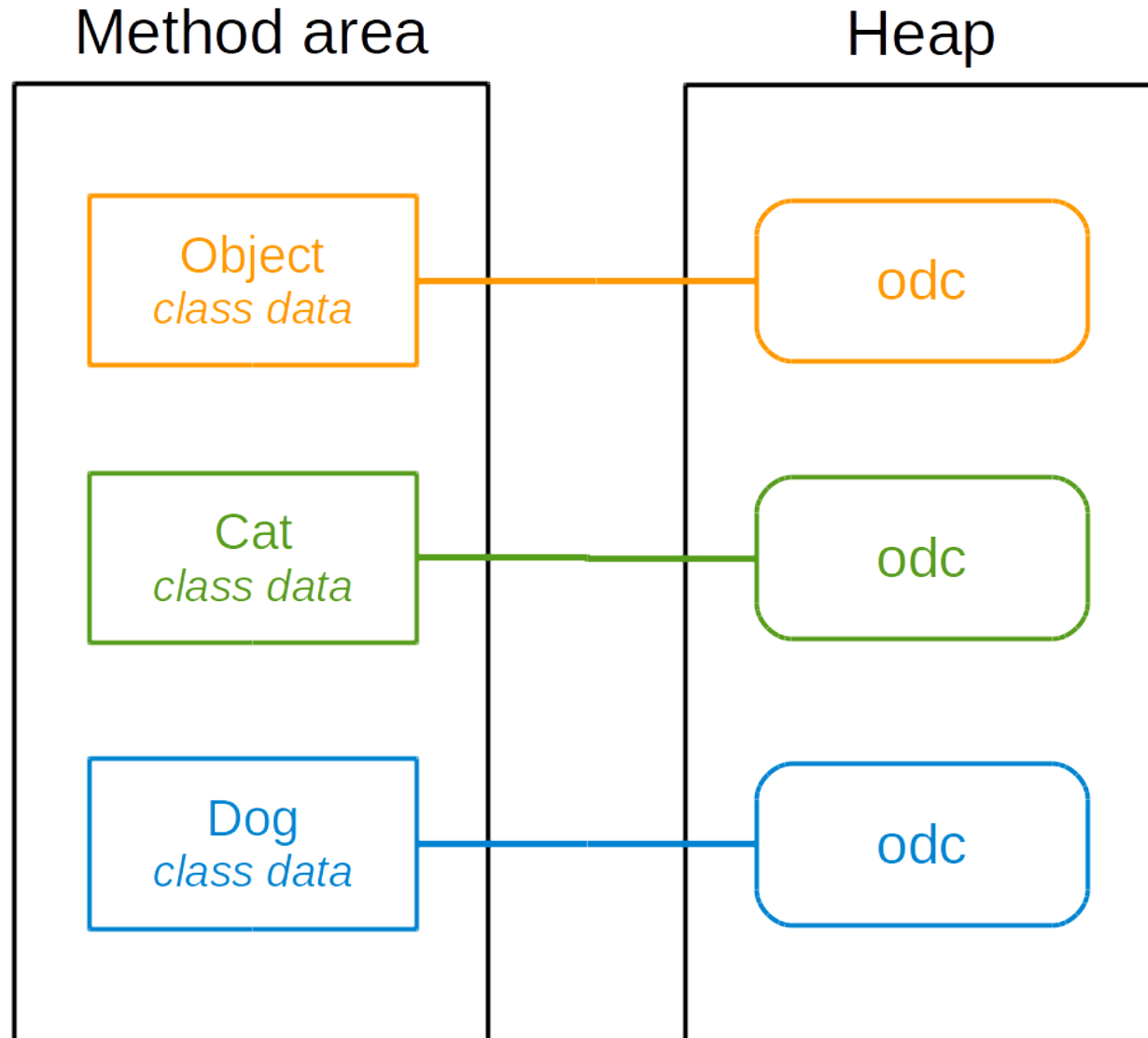
Example

```
class Calculatrice {  
  
    public static void main(String[] args) {  
        int resultat = calcul(5, 3);  
        System.out.println(resultat);  
    }  
  
    public static int addition(int a, int b) {  
        return a + b;  
    }  
  
    public static int soustraction(int a, int b) {  
        return a - b;  
    }  
  
    public static int calcul(int a, int b) {  
        return soustraction(addition(a, a), addition(b, b));  
    }  
  
}
```

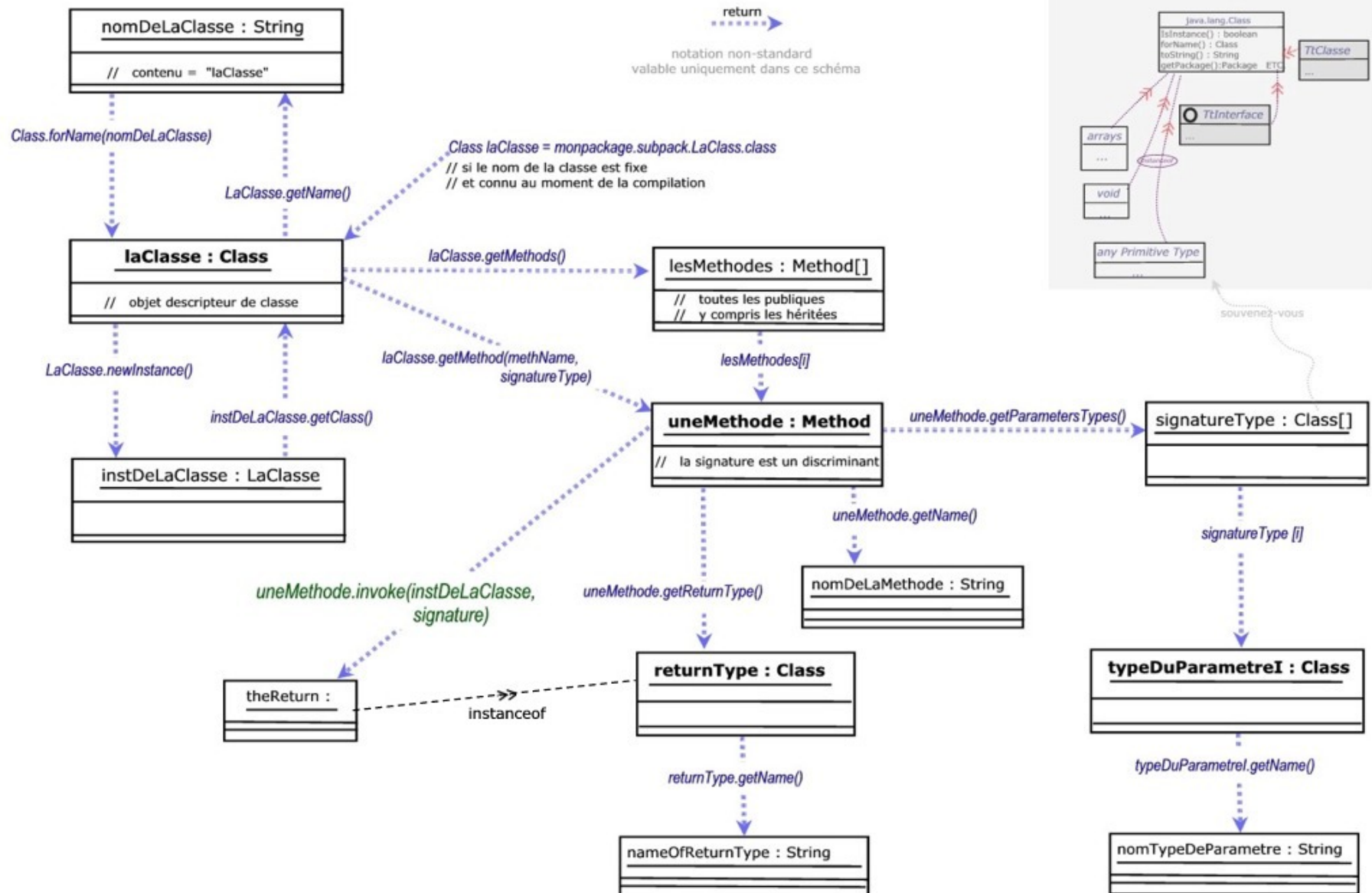
Example



ODC



Introspection



Exécution

- `java Zoo`
- Chargement de `Zoo.class`
 - Method area
 - Heap
 - Liaison des deux
 - `<clinit>`
- Exécution de `main()`
- `new Bear("Derek");`
 - Chargement de `Bear.class`
 - Allocation de mémoire dans le Heap
 - `<init>`
 - Constructeur
- [...]