

PROGRAMMATION JAVA

**Licence d'Université Spécialisée
Ingénierie Logicielle et Systèmes d'Information (ILSI)**

Pr. Nawal AIT AALI

E-mail: nawal.aitaali@gmail.com

Présentation

- Cours
- Travaux Dirigées
- Travaux Pratiques

Plan

- Introduction générale
- La programmation Orientée Objet et Java
- La modélisation par UML
- Les éléments de la programmation Java
- L'Héritage
- Les classes abstraites et les interfaces
- La gestion des flux
- La programmation graphiques (SWING)

Introduction générale

- Le langage Java est un langage capable de s'exécuter sur n'importe quelle plate-forme car c'est d'une part un langage **compilé** et d'autre part un langage **interprété**.
- Le code source Java est transformé en de simples instructions binaires. (Byte Code= Instructions générées par le compilateur qu'un ordinateur abstrait peut exécuter).

Introduction générale

Utilisation du langage JAVA:

- Applications de bureau: création des applications de bureau multiplateformes qui peuvent s'exécuter sur différents systèmes d'exploitation.
- Applications Mobiles: Création et développement des applications mobiles Android.
- Applications web: création et développement des applications web coté serveur.

Introduction générale

Utilisation du langage JAVA:

- Applications d'entreprises: création et développement des applications d'entreprise (applications de gestion des ressources humaines...)
- Jeux: création et développement des Jeux.

Introduction générale

Caractéristiques et avantages

- Simple:
- Le code source en JAVA est simple car JAVA permet la définition des classes et de leurs méthodes (pas besoin de #define, typedef, etc).

Introduction générale

Caractéristiques et avantages

➤ Robuste:

- Les compilateurs très stricts car toutes les valeurs doivent être initialisées.
- Les erreurs à l'exécution sont vérifiées.
- La gestion d'exceptions (la gestion et le traitement des erreurs).
- La gestion automatique de la mémoire.

Introduction générale

Caractéristiques et avantages

➤ Portable:

- Le programme JAVA est indépendant de la plateforme.
- Le compilateur Java génère du **byte-code**. (Le byte-code constitue les instructions pour la machine virtuelle JVM)
- La Java Virtual Machine existe par défaut sur de nombreuses plateformes (Unix, Windows, Mac...) et peut être aisément installée.

Introduction générale

Caractéristiques et avantages

➤ Orienté Objet:

- Java est un langage de programmation orienté objet,
- Java encourage la structuration du code en objets réutilisables.

Programmation Orienté Objet

Programmation applicative

- Fondée sur l'évaluation d'expressions qui ne dépendent que de la valeur des arguments, et non de l'état de la mémoire.
- Elle s'agit d'une programmation fonctionnelle.
- Proche des notations mathématiques: elle Accepte des arguments et produit un résultat.

Programmation impérative

- Fondé sur l'exécution d'instructions qui modifient l'état de la mémoire.
- Utilise beaucoup les itérations et autres structures de contrôle.

Programmation Orienté Objet

Programmation orientée objet

- C'est un style de programmation où l'on considère que des composants autonomes (les **objets**) disposent de ressources et de moyens d'interactions entre eux.
- Ces objets représentent des données qui sont modélisées par des **classes** qui définissent des types.

Programmation Orienté Objet

Programmation orientée objet

- Un objet est défini par :

- Un état
- Un comportement
- Une identité

Personne
Variable: Age=32
Methode: getAge()

Identité

État

Comportement

- État : représenté par des attributs(variable) qui stockent des valeurs.
- Comportement : défini par des méthodes qui modifient un état.
- Identité : permet de distinguer un objet d'un autre objet.

Programmation Orienté Objet

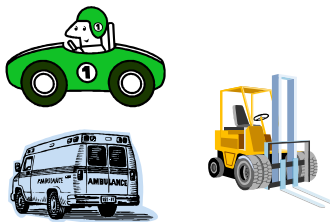
Programmation orientée objet

➤ Notion de classe:

Des objets similaires peuvent être informatiquement décrits par une même abstraction : une classe

- même structure de données et méthodes de traitement
- valeurs différentes pour chaque objet

Classe Voiture
-modèle, couleur



Classe Personne
-nom, prénom, statut



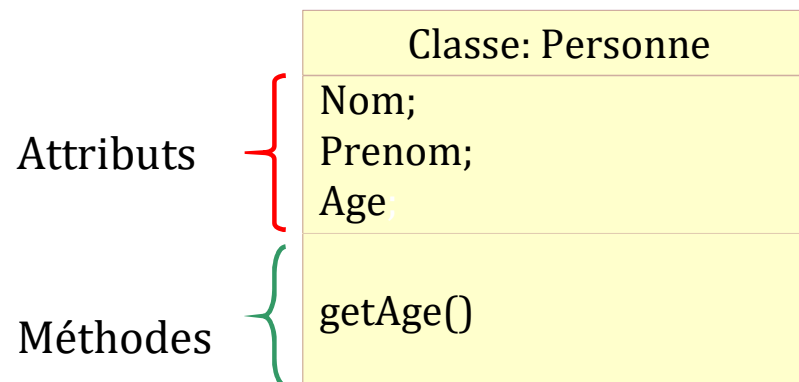
Classe Lecteur
-nom, prénom



Programmation Orienté Objet

Programmation orientée objet

- Notion de classe:
 - Une classe est composée de plusieurs membres dont chacun est soit :
 - un attribut : variable typée
 - une méthode (ou opération) : ensemble d'instructions de traitement.

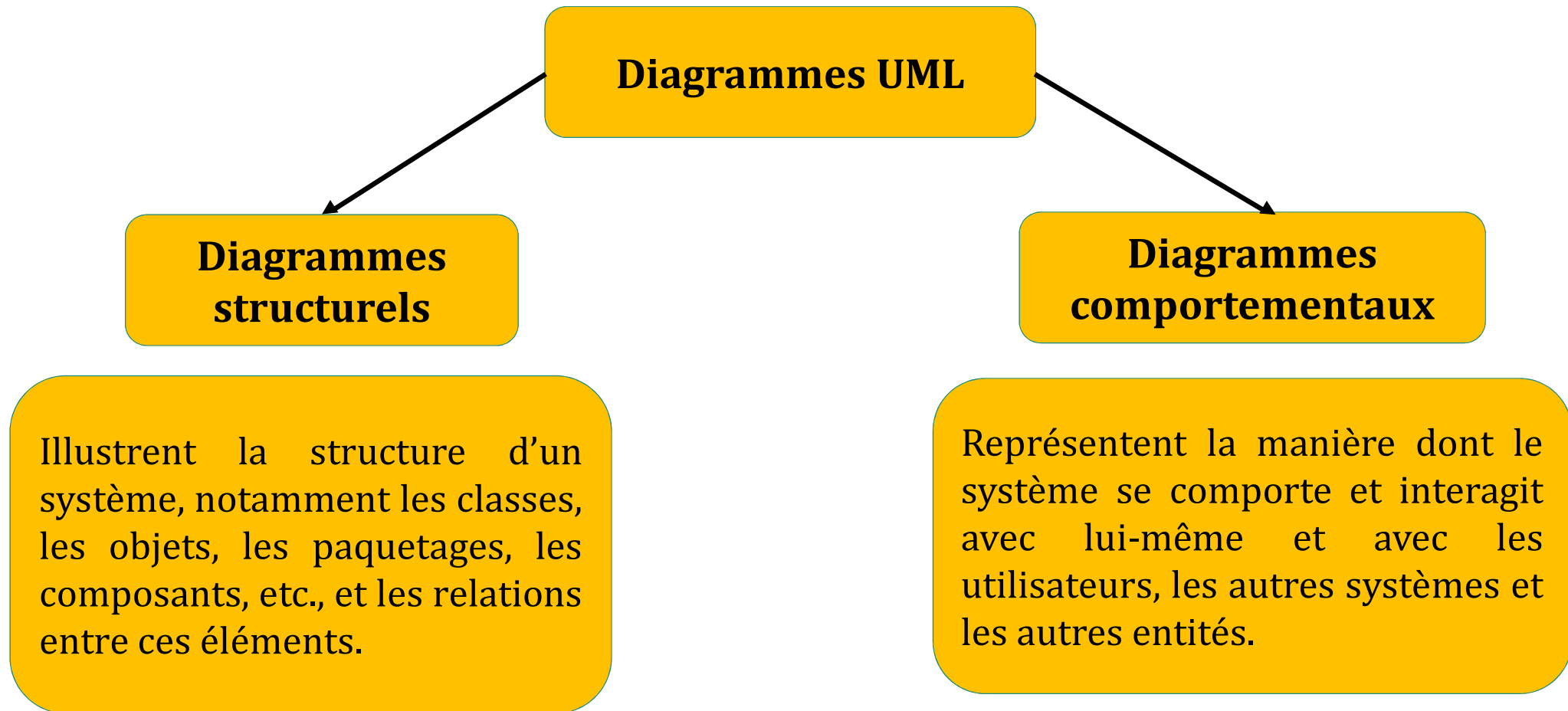


Modélisation par UML

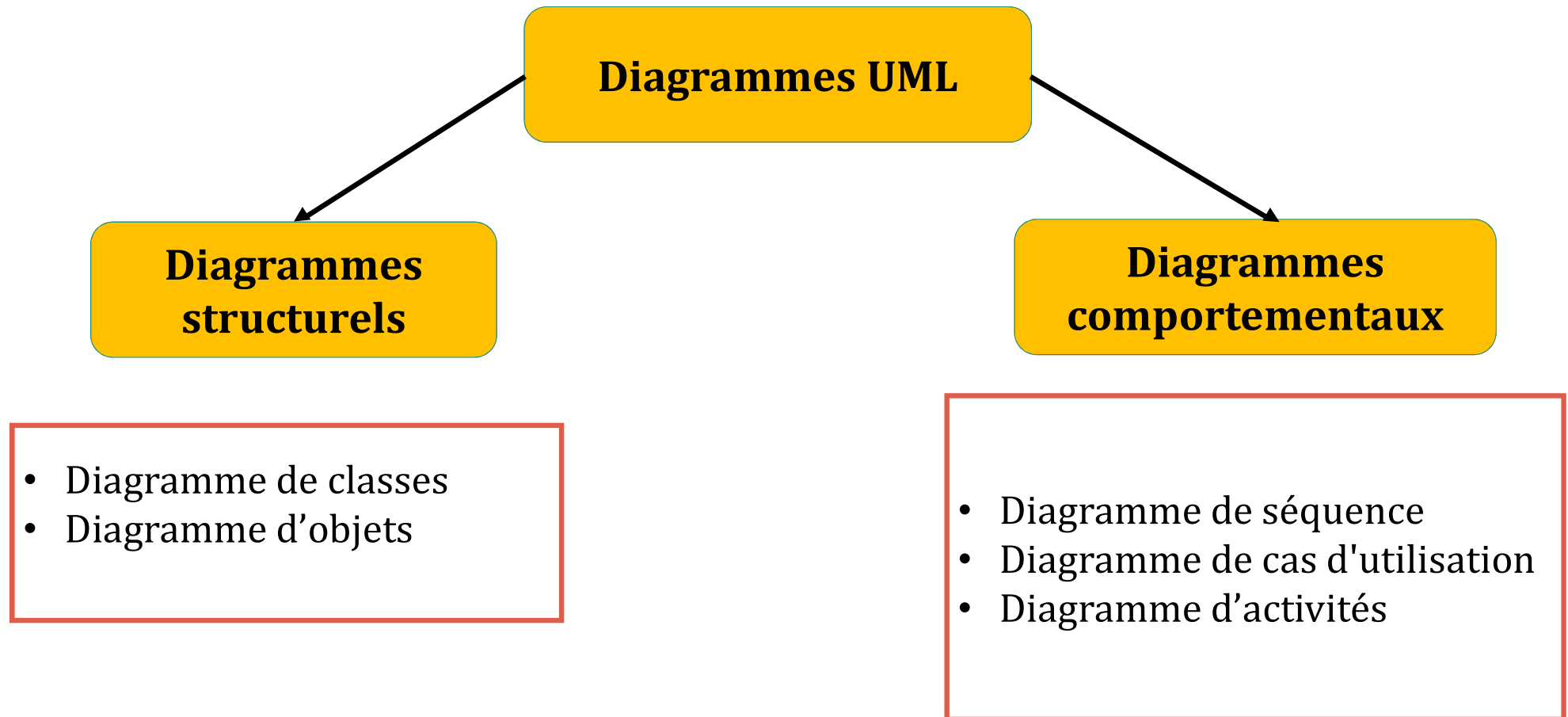
Définition

- UML: (Unified Modeling Language) est un langage standard de modélisation orienté-objet.
- C'est un moyen pour représenter visuellement l'architecture, la conception et la mise en œuvre des systèmes complexes.
- UML contient différents diagrammes utilisés pour décrire un problème informatique.

Modélisation par UML



Modélisation par UML



Modélisation par UML

Diagramme de classes

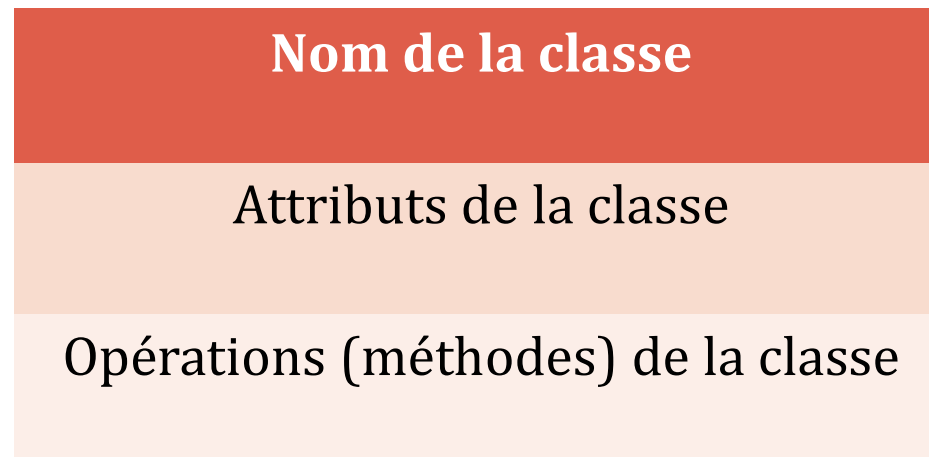
➤ *Avantages:*

- Illustrer des modèles de données pour des systèmes d'information, quel que soit leur degré de complexité.
- Mieux comprendre l'aperçu général des schémas d'une application.
- Créer des schémas détaillés qui mettent l'accent sur le code spécifique qui doit être programmé et mis en œuvre

Modélisation par UML

Diagramme de classes

- *Composants de base d'un diagramme de classes*

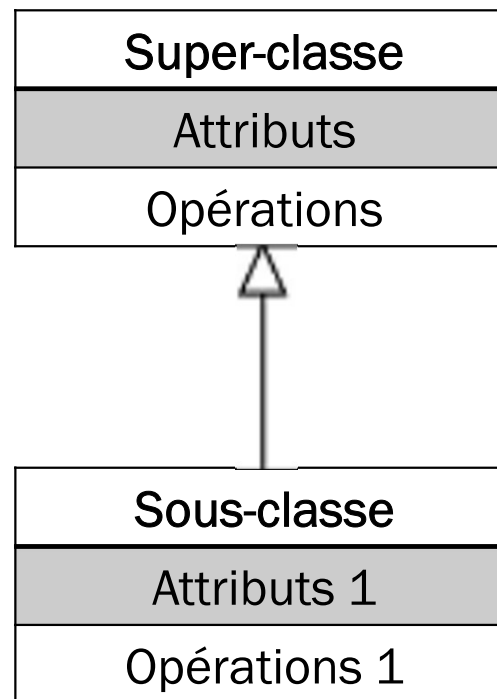


Modélisation par UML

Diagramme de classes

➤ *Interaction entre les classes*

- **Héritage:** Il s'agit du processus par lequel un enfant ou une sous-classe adopte la fonctionnalité d'un parent ou d'une super-classe.

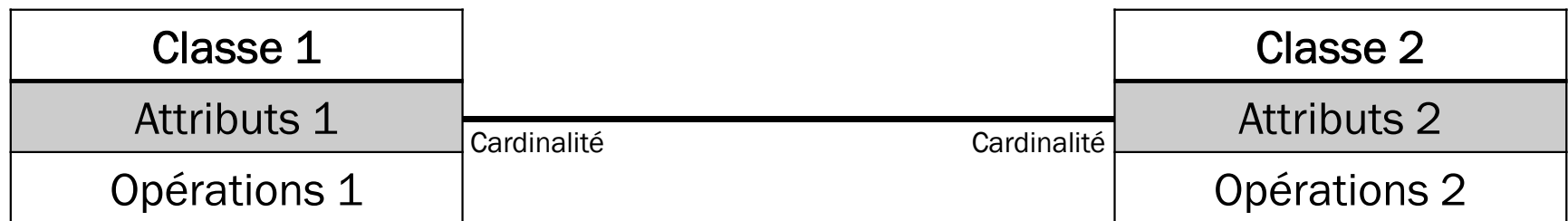


Modélisation par UML

Diagramme de classes

➤ *Interaction entre les classes*

- Association bidirectionnelle: relation par défaut entre deux classes. Chacune des deux classes a conscience de l'existence de l'autre et de sa relation avec elle. Cette association est représentée par une ligne droite entre deux classes.



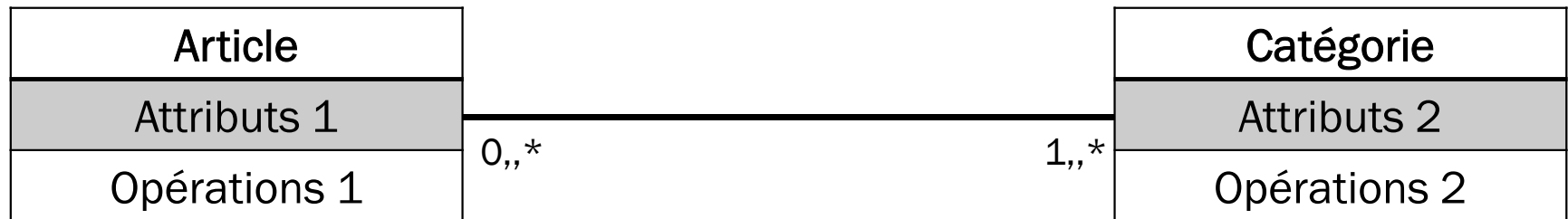
Modélisation par UML

Diagramme de classes

➤ *Interaction entre les classes*

- Association bidirectionnelle:

Exemple

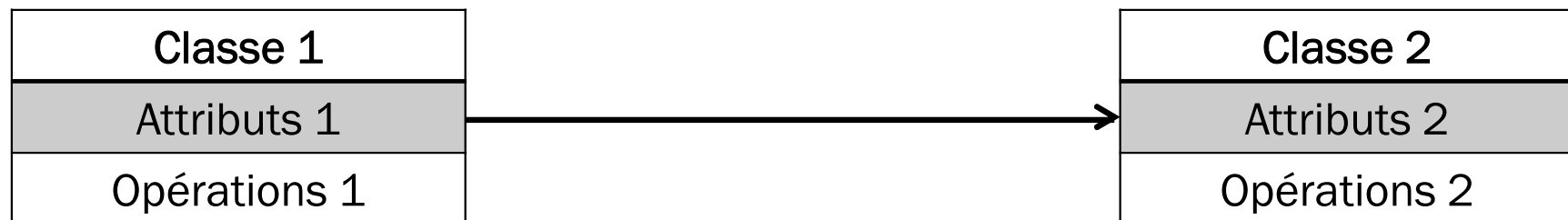


Modélisation par UML

Diagramme de classes

➤ *Interaction entre les classes*

- Association unidirectionnelle: Une classe a conscience de l'existence de l'autre et interagit avec elle. Une association unidirectionnelle est représentée par une ligne de connexion droite avec une pointe de flèche ouverte allant de la classe « sachante » vers la classe « connue ».



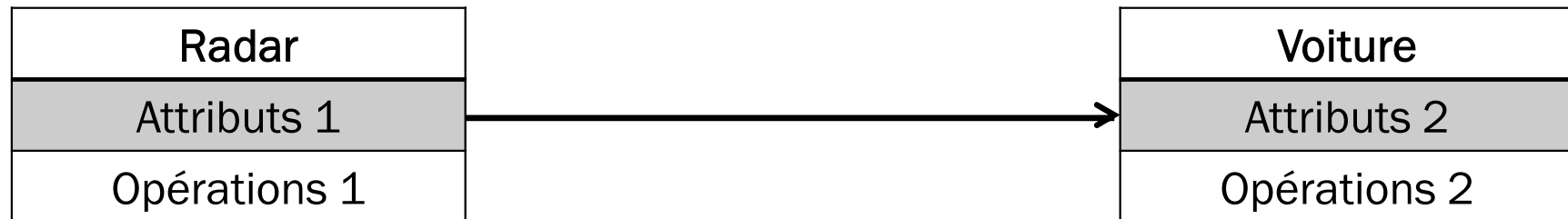
Modélisation par UML

Diagramme de classes

➤ *Interaction entre les classes*

- Association unidirectionnelle:

Exemple



Modélisation par UML

Diagramme de classes

➤ *Exercice:*

Elaborez les diagrammes de classe correspondants en choisissant le type de relation approprié pour les phrases suivantes:

- Un répertoire contient plusieurs fichiers
- Une pièce contient des murs
- Les modems et claviers sont des périphériques d'entrée / sortie
- Une transaction boursière est un achat ou une vente
- Un compte bancaire peut appartenir à une personne physique ou morale

Modélisation par UML

Diagramme de classes

➤ *Exercice:*

Elaborez les diagrammes de classe correspondants en choisissant le type de relation approprié pour les phrases suivantes:

- Tout écrivain a écrit au moins une œuvre
- Les personnes peuvent être associées à des universités en tant qu'étudiants aussi bien qu'en tant que professeurs et administrateurs.
- Les cinémas sont composés de plusieurs salles. Les films sont projetés dans des salles.

Les éléments de la programmation Java

➤ Classe en Java

Le code source d'un programme Java est contenu dans plusieurs fichiers d'extension **.java**

- Le **nom du fichier** doit être le même que celui de **la classe définie**;
- Par convention, le nom d'une classe commence toujours par une majuscule.
- L'ensemble des classes définies sont regroupées dans un paquet (Package).

Le code source d'une classe commence par le mot-clé class suivi de son contenu :

```
class <nom de la classe> {  
    <contenu de la classe>  
}
```

Les éléments de la programmation Java

➤ Méthodes en Java

Une méthode est composée de sa signature et d'un traitement associé.

La signature d'une méthode contient notamment :

- un nom de méthode;
- un type de données de retour;
- des types de données pour ses arguments;

Par convention le nom des méthodes et attributs commencent toujours par **une minuscule.**

```
<type de retour> <nom de méthode> ( <arguments> * ) {  
    <traitement associé>  
}
```

Les éléments de la programmation Java

➤ **Exemple:**

```
class Livre {  
    String titre;  
  
    void setTitre(String t) {  
        titre = t;  
    }  
  
    String getTitre() {  
        return titre;  
    }  
}
```

Les éléments de la programmation Java

➤ Préparer un code source en JAVA

Commentaires en JAVA (trois façons)

- Tout texte entre « **//** » et la fin de la ligne
// Commentaires sur une seule ligne
- Tout texte entre « **/*** » et « ***/** »
/* Commentaires
sur un nombre important voire très important
de lignes ***/**

Les éléments de la programmation Java

➤ Préparer un code source en JAVA

Identificateurs en JAVA

Un identificateur (identifier) permet de désigner une classe, une méthode, une variable ...

Interdire d'utiliser les mots-clés (ex: abstract, new, return, try, ...)

Un identificateur commence par :

- Une lettre
- Un « \$ »
- Un « _ » (underscore)

Il ne commencent pas par :

- Un chiffre
- Un signe de ponctuation autre que « \$ » ou « _ »

Les éléments de la programmation Java

Les mots clés de JAVA		
abstract	finally	public
boolean	float	return
break	for	short
byte	goto	static
case	if	strictfp
catch	implements	super
char	import	switch
class	instanceof	synchronized
const	int	this
continue	interface	throw
default	long	throws
do	native	transient
double	new	try
else	package	void
extends	private	volatile
final	protected	while

Les éléments de la programmation Java

➤ Préparer un code source en JAVA

Types de données

Les entiers

byte	8 bits	-2^7 à 2^7-1
short	16 bits	-2^{15} à $2^{15}-1$
int	32 bits	-2^{31} à $2^{31}-1$
long	64 bits	-2^{63} à $2^{63}-1$

Les flottants

float	32 bits	$1.4e^{-45}$ à $1.4e^{38}$
double	64 bits	$4.9e^{-324}$ à $1.8e^{308}$

Logique

Les caractères

char	16 bits	0 à 65535 (2^{16})
------	---------	------------------------

boolean	2 valeur	True ou false
---------	----------	---------------

Les éléments de la programmation Java

➤ **Préparer un code source en JAVA**

Déclaration et initialisation des types de données

Int	int x=12;
short	short x= 32;
double	double x=23.2323;
Float	float x= 23.233F;
char	char c='a';
boolean	boolean b=true;

Les éléments de la programmation Java

➤ Préparer un code source en JAVA

Assignation de types de données

- `int u = 3;` Déclaration et initialisation d'un entier
- `int t;` Déclaration d'un entier t (t est l'identificateur)
- `t=7;` Initialisation de t à la valeur 7
- `u=t;` Assignation (affectation) de la valeur de t à u
- `m=9;` Erreur déclaration préalable nécessaire
- `char c;` Déclaration
- `c='a';` Initialisation

Les éléments de la programmation Java

➤ Préparer un code source en JAVA

Arithmétique et opérations

Pour comparer deux valeurs

Opérateur	Exemple	Renvoie TRUE si
>	$v1 > v2$	$v1$ plus grand que $v2$
>=	$v1 \geq v2$	Plus grand ou égal
<	$v1 < v2$	Plus petit que
<=	$v1 \leq v2$	Plus petit ou égal à
==	$v1 == v2$	égal
!=	$v1 \neq v2$	différent

Les éléments de la programmation Java

➤ Préparer un code source en JAVA

Opérateurs Logiques et opérations

Opérateur	Usage	Renvoie TRUE si
<code>&&</code> <code>&</code>	<code>expr1 && expr2</code> <code>expr1 & expr2</code>	<code>expr1</code> et <code>expr2</code> sont vraies Idem(opérateur bits à bits) mais évalue toujours les 2 expressions
<code> </code> <code> </code>	<code>expr1 expr2</code> <code>expr1 expr2</code>	<code>Expr1</code> ou <code>expr2</code> , ou les deux sont vraies Idem(opérateur bits à bits) mais évalue toujours les 2 expressions
<code>!</code>	<code>! expr1</code>	<code>expr1</code> est fausse
<code>!=</code>	<code>expr1 != expr2</code>	si <code>expr1</code> est différent de <code>expr2</code>

Les éléments de la programmation Java

➤ Préparer un code source en JAVA

Arithmétique et opérations

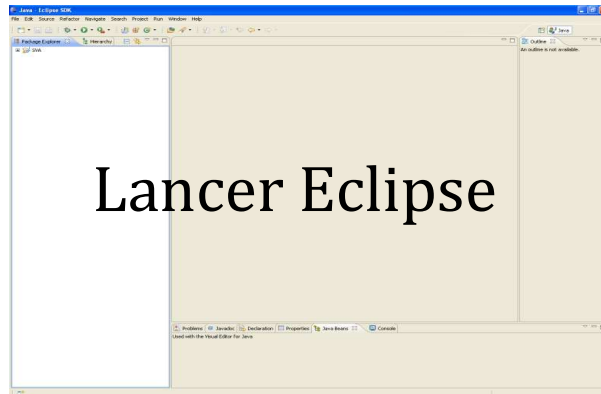
L'opérateur de base est '='

Il existe des opérateurs d'assignation qui réalisent à la fois une opération arithmétique, logique, ou bit à bit et l'assignation proprement dite

Opérateur	Exemple	Équivalent à
+=	expr1 += expr2	expr1 = expr1 + expr2
-=	expr1 -= expr2	expr1 = expr1 – expr2
*=	expr1 *= expr2	expr1 = expr1 * expr2
/=	expr1 /= expr2	expr1 = expr1 / expr2
%=	expr1 %= expr2	expr1 = expr1 % expr2

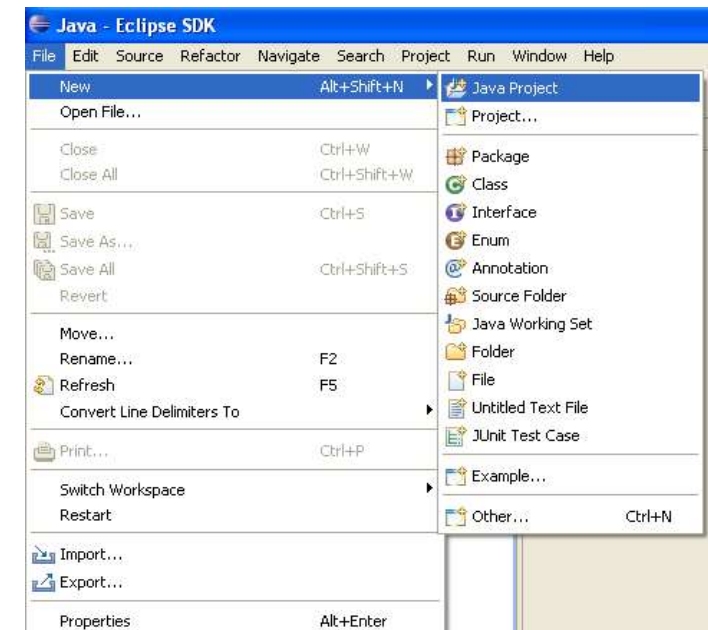
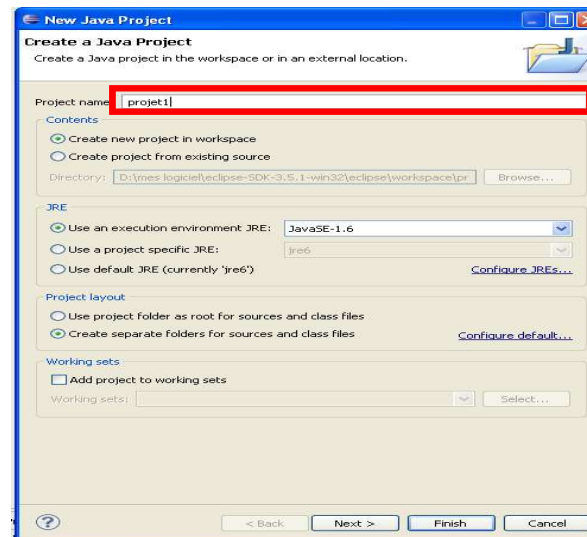
Les éléments de la programmation Java

➤ Etapes d'exécution de code JAVA



Nom du projet

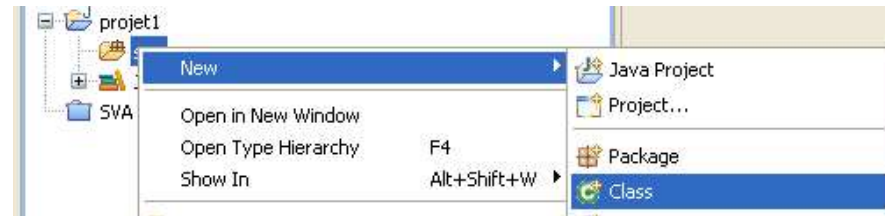
Créer
Nouveau
projet



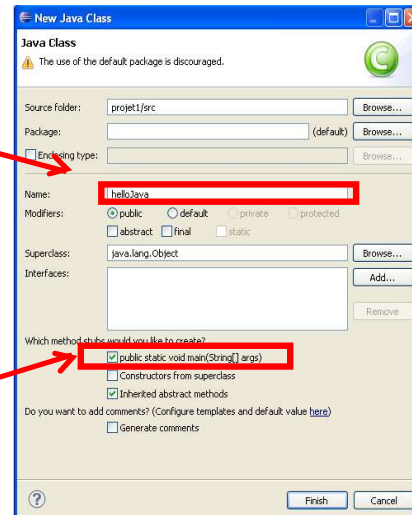
Les éléments de la programmation Java

➤ Etapes d'exécution de code JAVA

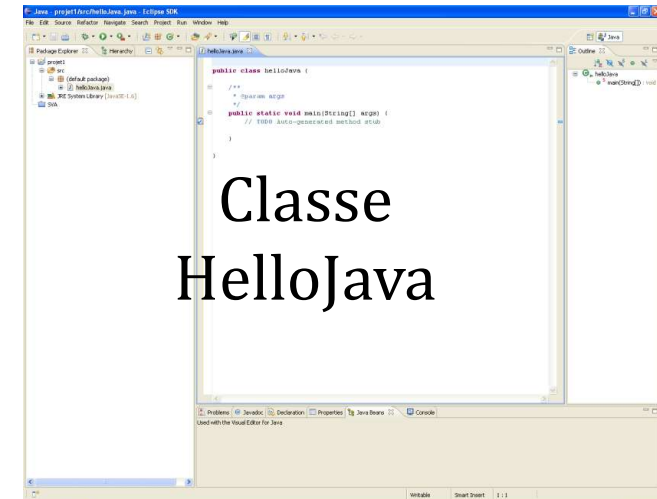
Créer Nouvelle
classe



Nom de la classe

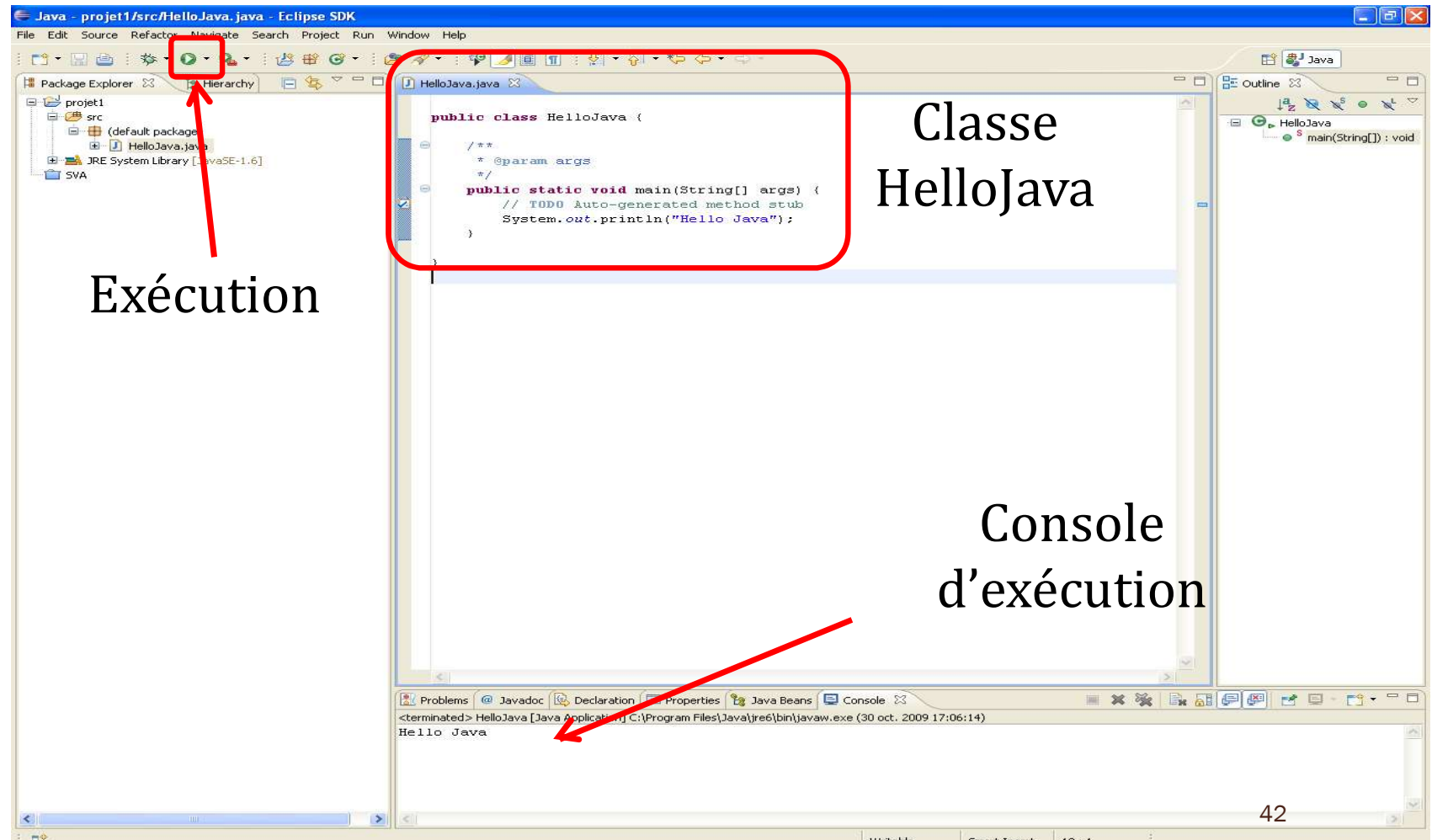


Activé cette option
pour la génération
de la fonction main



Les éléments de la programmation Java

➤ Etapes d'exécution de code JAVA



Les éléments de la programmation Java

➤ Etapes d'exécution de code JAVA

Exemple:

```
package prj;

public class Hellojava{
    public static void main (String args[]) {
        System.out.println("Bonjour tout le monde !");
    }
}
```

Les éléments de la programmation Java

➤ Instanciation des classes/ Création des objets

Syntaxe en JAVA

```
<nom de la classe> <nom d'objet> = new <nom de la classe> ();
```

- L'opérateur **new** provoque une allocation mémoire.
- Un processus appelé ramasse-miettes (Garbage Collector), s'occupe de libérer la mémoire.

Exemple 1:

```
Voiture mavoiture = new Voiture();
```

Les éléments de la programmation Java

➤ Instanciation des classes/ Création des objets

Exemple 2:

```
package Pratique;
```

```
public class Voiture {  
    public int vitesse;  
    public int modèle;  
    public String couleur;
```

```
    public void setvitesse(int V) {  
        vitesse = V;  
    }  
}
```

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub
```

```
    Voiture mavoiture= new Voiture();
```

```
    mavoiture.vitesse= 2;  
    mavoiture.modèle= 2018;  
    mavoiture.couleur= "noir";  
    mavoiture.setvitesse(3);  
    System.out.println(mavoiture.modèle);  
    System.out.println(mavoiture.vitesse);
```

```
}
```

```
}
```

Les éléments de la programmation Java

TP n°1

- Créer un programme en JAVA qui permet de créer une classe Personne ayant les attributs suivants: nom, prénom et âge.
- Ajouter une méthode qui permet de modifier l'âge de la personne
- Instancier la classe Personne avec 2 objets distincts.
- Modifier l'âge saisi pour les deux objets.
- Afficher les résultats de l'instanciation.