

# Java

## Classes & Objets

University of Geneva  
www.miralab.ch

## Notions : objet, class, héritage

2

### Classe

C'est un type d'objet

Définition d'un objet en terme de

Contenus (variables)

Méthodes à exécuter pour accéder / modifier son contenu

### Objet

Instance d'une classe

Créé par un **constructeur**

une méthode spéciale fournie par la classe

### Héritage

Relation entre les classes,  
donc entre les objets

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

3

## Exemple : compte bancaire

Classe Compte	Objet compte1
Variable numéro	numéro = #37445567
Variable solde	solde = 10 233.55
Méthode ChangerNumero()	ChangerNumero()
Méthode CalculerSolde()	CalculerSolde()

Objet compte2
numéro = #374477
solde = 340 233.55
ChangerNumero()
CalculerSolde()

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

4

## Classe

Définit un type d'objet en termes de

Variables = stockage d'état

Méthodes = opérations changeant l'état de l'objet

Différents types de variable

Variable de classe

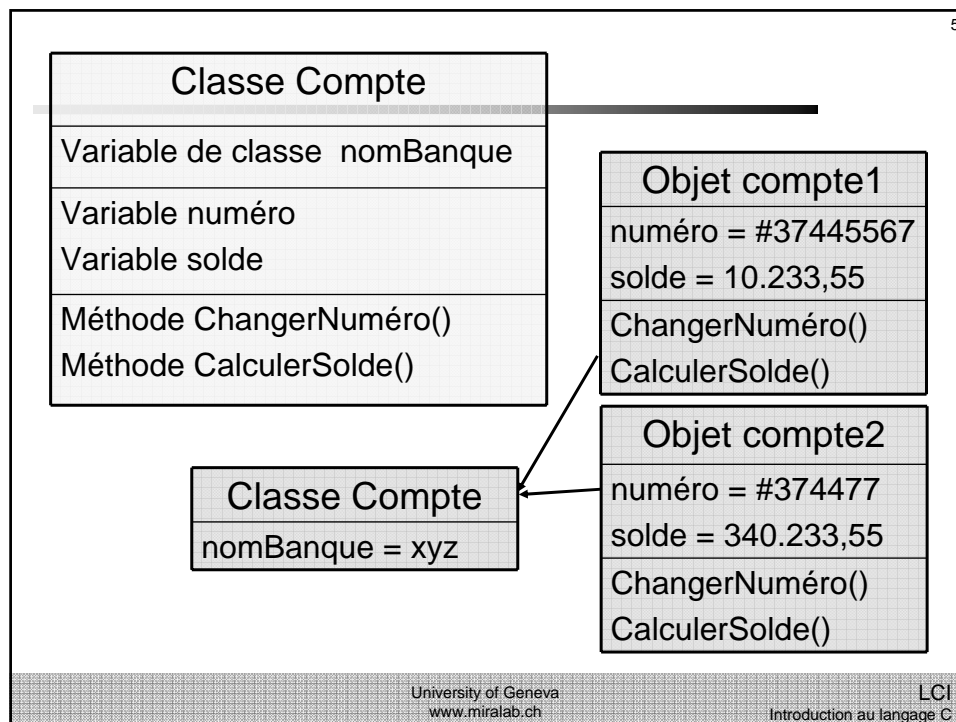
Commun à tous les objets

Variable d'instance

Lié à une instance d'objet

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C



## Variable d'instance

Utilise ces mots clefs comme qualificatifs  
(*modifier*)

*public* indique un accès libre à la variable

*private* indique un accès exclusif par les  
méthodes de la classe

*protected* indique un accès réservé aux  
méthodes de la classe et de ses descendants

Sans mot clef accessible seulement par les  
autres classes du même *package*

## Variable d'instance, exemple

```

class Compte
{
    public int numéro;
    private float solde;
    static public int nb2comptes = 0;
    { // bloc d'initialisation
        numéro = 0;
        solde = 0.0f;
        nb2compte++;
    }
}

```

Diagram annotations:

- Two arrows point from the text "Variable d'instance" to the declarations of `numéro` and `solde`.
- One arrow points from the text "Variable de classe" to the declaration of `nb2comptes`.

## Objet

### Instance d'une classe

Contient les valeurs associées à un objet particulier

pour chaque variables d'instance de la classe qui le définit

⇔ Un « exemplaire » de la classe avec ces variables d'instance

## Différence classe vs objet

### Classe

Définit le modèle de l'objet, de ses propriétés et attributs généraux

Ex: classe vélo

couleur, nombre de vitesse...

### Objet

Incarné une **instance** avec des paramètres propres

Ex: objet vélo

couleur = bleu, vitesse = 10, ...

## Analogie type / classe

### Type/variable

Déclaration

**int** a;

Initialisation

a = 0;

Opérations

**int** b = a\*2;

### Classe/objet

Déclaration

**String** s;

Initialisation

s = new **String**("test");

Opérations

**String** t = s + "2";

## Constantes

11

Déclaration, puis première initialisation

```
final int numéroBanque;
numéroBanque = 2345; //ok
numéroBanque = 2345; //erreur
```

Déclaration avec initialisation

```
final int numéroBanque = 2345;
numéroBanque = 2345; //erreur
```

## Exemple de constantes : Float

12

**Champs**

```
public final static float POSITIVE_INFINITY
public final static float NEGATIVE_INFINITY
public final static float NaN
public final static double MAX_VALUE
    1.40129846432481707e-45f
public final static double MIN_VALUE
    3.40282346638528860e+38f
```

## Variable de classe

13

Utilise le mot clef *static*

```
class Compte
{
    static int nb2comptes = 0;
    static String nomBanque = "MiraB";
    static final int numéroBanque;
    static
    { //bloc d'initialisation static
        numéroBanque = 0;
    }
}
```

## Accès à une variable d'un objet

14

Se fait en précisant le nom de l'objet suivit de "." et du nom de la variable

```
Compte client1 = new Compte();
if(client1.numéro == 1)
    client1.solde += 10000.0f;
```

Rappel :

L'accès est contrôlé selon

le type de variable *private*, *public*, ...  
et la classe de la méthode appelante

15

```

class Compte {
    public int numéro;
    private float solde;
    public float calculerSolde()
    {    return this.solde; }
}

class AutreCompte {
    private Compte c = new Compte();
    public float autreSolde()
    {    if(c.numéro > 0) //ok
        return c.solde; //erreur
        return 1.0f; }
}

```

16

## Structure d'un programme Java

### 3 sections optionnelles

Déclaration de *package*

Import de packages ou de classes: *import*

Déclarations des classes et des interfaces du fichier



## Package

17

Définit un groupe dans lequel les classes doivent être rangées/associées

Cette directive doit être placée au début du fichier

Le nom du package correspond à un répertoire dans lequel les classes doivent être placées

```
package altnet;
```

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Exemple

18

### Code

```
package altnet;
class AltClasse1
{
}
```

### Répertoire/fichier

```
altnet/
altnet/AltClasse1.class
```

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

## Import

19

Permet de

charger des classes d'autres packages  
en vue d'utiliser leurs fonctionnalités

On peut spécifier

le nom du package qui contient la ou les  
classes

```
import cePackage.*;
```

le nom explicite d'une classe

```
import package.autreClasse;
```

```
import cetteClasse;
```

## Déclaration de classes

20

Syntaxe : *qualificatif* **class** *nomClasse* { }

```
public class Classe2
{ // Corps de Classe2
}
```

Les qualificatifs sont

*public* indique la classe exportée par le fichier .java  
une seule classe de ce type par fichier  
par convention prend le même nom que le fichier .java

*final* indique que la classe ne peut pas être dérivée

*abstract* obligatoire pour les classes contenant des  
méthodes *abstract*

On ne peut pas créer d'instance d'objet de ces classes

21

## Déclaration de classes

Le corps d'une classe peut contenir

- Des déclarations de variables
- Des déclarations de méthodes
- Des blocs d'initialisations
- Des classes internes
  - Classes déclarées à l'intérieur d'une classe

22

## Classes internes

Qualificatifs :

*public, private, protected*

Sens identique aux variables d'instance

*final, abstract*

Sens identique aux classes externes

*static*

Indique que la classe est indépendante des instances d'objet

23

## Création de classe interne static

```

class Externe
{
    static class Interne {}
}
class Autre
{
    public Externe.Interne ei;
    { ei= new Externe.Interne(); }
}

```

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

24

## Création de classe interne non static

```

class Externe
{ public class Interne {}
}
final class Autre
{
    public Externe e = new Externe();
    public Externe.Interne ei;
    { ei= new Externe.Interne(); //erreur
      ei= e.new Interne(); //ok
    }
}

```

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

25

## Classe interne non static

Dépend de l'instance de la classe externe  
 2 objets d'une même classe externe peuvent  
 stocker des instance internes différentes

Conserve une référence vers cette classe  
 Permet d'appeler les méthodes de la classe  
 externe

26

## Classes internes imbriquées

Une classe interne peut à son tour  
 contenir des classes internes

```
class Externe
{ public class Interne1
  {
    private class Interne2 {}
  }
}
```

## Compilation

27

Chaque classe d'un fichier .java génère un fichier .class lors de la compilation

Autre.class

Externe.class

Pour 1 classe interne le nom du fichier =  
concaténation du nom classe externe et interne  
séparé par le symbole \$

Externe\$Interne.class

Externe\$Interne1\$Interne2.class

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

28

```
/* Début du fichier MaClasse.java */
/* 1/ éventuelle déclaration de package */
package MonPackage;
/* 2/ 0 ou plusieurs import */

// Importe classe sans package
import MaClasse;
// Importe classe d'un package
import MonPackage.MaClasse;
// Importe toutes les classes d'un package
import MonPackage.*;
```

University of Geneva  
www.miralab.ch

LCI  
Introduction au langage C

29

```

/* 3. Déclarations des classes */

public class MaClasse
// 1 seule classe déclarée public
// par convention porte le même nom que le fichier
{ // Corps de MaClasse
}

class MaClasse2
{ // Corps de MaClasse2
}

/* Fin du fichier MaClasse.java */

```

30

## Résumé

### Notions

d'objet  
de classe  
d'héritage

### Structure d'un programme

### Classes externes

### Variable

de classe  
d'instance

### Classes internes