Interface homme-machine et langage Java

Henri Garreta, Faculté des Sciences (Luminy) Cyril Pain-Barre, IUT d'Aix-Marseille (Aix)

http://henri.garreta.perso.luminy.univmed.fr/IHM_Java

Cours 1. Java pour les programmeurs C++

© H. Garreta, 2013

HM et Java

1 / 13

Java pour les programmeurs C++

Structure générale d'un fichier source

En Java, il n'y a pas

- des directives pour le préprocesseur (lignes commençant par #)
- o notamment, il n'y a pas besoin de #include
- des fichiers de spécification (« .h ») et des fichiers d'implémentation (« .cpp ») séparés
- des variables globales
- des fonctions autonomes

```
Fichier Liste.java

package mesOutils.mesListes;

import java.outil.Vector;
import java.aut.*;

class Maillon {

    déclaration des membres de la classe Maillon
}

public class Liste {

    déclaration des membres de la classe Liste
}
```

- recommandation : une seule classe par fichier
- obligation : une seule classe « public » par fichier

Java pour les programmeurs C-

Concepts des langages orientés objets

- un programme est un ensemble d'objets
- un objet est déterminé par un état et un comportement
- état : défini par les valeurs d'un ensemble de *données membres*, *champs* ou *variables d'instance*
- comportement : défini par un ensemble de *fonctions membres* ou *méthodes d'instance*
- une *classe* est la description d'une sorte d'*objets* de même type, les *instances* de la classe
- il y a aussi des données et des fonction non attachées aux objets :
 - les données membres static ou variables de classe
 - les fonctions membres static ou *méthodes de classe*

© H. Garreta, 2013 IHM et Java 2 / 1

Package

• subdivision de l'espace des noms, à la manière des namespace

```
Fichier Liste.java

package mesOutils.mesListes;

public class Liste {
...
}
```

- le nom complet de cette classe est mesOutils.mesListes.Liste
- l'instruction package doit être la première du fichier
- si absente, on est dans le « paquet sans nom » (unnamed package)
- les classes « public » sont visibles (utilisables) partout
- les classes non public sont visibles uniquement dans les autres classes du même package

© II Court 2012

Import

définition d'une classe dans un package :

```
Fichier Liste.java
  package mesOutils.mesListes;
  public class Liste {
```

• exemple d'utilisation de cette classe dans un autre package :

mesOutils.mesListes.Liste L = new mesOutils.mesListes.Liste();

• import permet d'utiliser le *nom court* des classes : import mesOutils.mesListes.Liste;

```
Liste L = new Liste();
```

• forme « collective » :

import mesOutils.mesListes.*;

cela désigne toutes les classes du paquet mesOutils.mesListes

import n'est pas « #include »

Java, le langage

- Sauf indication contraire, Java = C: même syntaxe, mêmes concepts de base
 - + la surcharge des noms des fonctions (comme en C++).
- Types primitifs (simples):

byte short int long float double char boolean

tous les entiers sont « avec signe »

Les types primitifs obéissent à la sémantique de l'accès par valeur

dans « x == 0 », x signifie « la valeur de la chose nommée x »

- Types objets :
 - tableaux
 - chaînes de caractères (String)
 - toutes les autres classes

Les objets obéissent à la sémantique de l'accès par référence

Compiler et exécuter une classe

- Les exécutables Java sont de plusieurs sortes :
 - les applications s'exécutent dans une machine virtuelle Java (qui, elle-même, s'exécute sur un système d'exploitation)
 - les applets s'exécutent dans les navigateurs web
 - les servlets s'exécutent dans les serveurs d'applications
- les applications Java sont compilées et interprétées
- l'application la plus simple (en rouge, les éléments invariables)

```
Fichier Bonjour.java
   public class Bonjour {
       public static void main(String[] args) {
           System.out.println("Bonsoir!");
```

- o compilation :
 - ▶ javac Bonjour.java
- exécution : ▶ java Bonjour Bonsoir!

Références

```
• référence = « pointeur » géré de manière interne par Java
```

```
p contiendra une référence sur un objet Point
Point p;
                 pour commencer, p vaut null
```

• instantiation (création) d'un objet :

```
p = new Point(50, 50);
```

on peut aussi associer déclaration et instantiation :

```
Point q = new Point(100, 200);
```

 désignation des membres (sous réserve que la variable x et la fonction distance soient membres de la classe Point et qu'on ait le droit d'y accéder)

```
p.x
   p.distance(q)
(en C++ cela s'écrirait p->x et p->distance(q))
```

• les références sont modifiables. Par la suite, on pourra écrire :

```
p = new Point(300, 0);
                               la précédente valeur de p est « oubliee »
```

ava pour les programmeurs C++

Tableaux

```
Déclaration :
                                                  (ici tab vaut null)
    UnType tab[];
                      ои
                            UnType[] tab;
Création du tableau
    tab = new UnType[nombre d'éléments];
Accès au i<sup>eme</sup> élément :
    tab[i]
Nombre d'éléments du tableau :
    tab.length
Exemple
   public class Echo {
       public static void main(String args[]) {
           for (int i = 0; i < args.length; i++)
               System.out.println(i + ": " + args[i]);
   }
                               ▶ javac Echo.java
                               ▶ java Echo poids 200+30 'au revoir'
                               0: poids
                               1: 200+30
                               2: au revoir
```

© H. Garreta, 2013

IHM et Java

Java pour les programmeurs C++

Classes, variables et méthodes

En Java il n'y a pas :

- des méthodes définies en dehors de toute classe
- des méthodes qualifiées inline
 le compilateur se charge de détecter les méthodes qui le méritent
- des classes ou méthodes amies (friend)
 mais être du même paquet est une forme d'amitié
- d'opérateur "::" pour la résolution de portée; au lieu de UneClasse::unMembre on écrit
 UneClasse.unMembre

H. Garreta, 2013 IHM et Java 11 /

Java pour les programmeurs

Chaînes de caractères

• Les objets String représentent des chaînes de caractères immuables.

```
Construction simple et commode:
    String s = "Bonjour";
Concaténation:
    "abc" + "defg" → "abcdefg"
Propriété remarquable: si exp₁ est de type String
    exp₁ + exp₂ ⇔ exp₁ + conversionEnString(exp₂)
Exemple:
    System.out.println("x = " + x);
Deux cas: si exp₁ est une chaîne et exp₂ d'un type primitif
    exp₁ + exp₂ ⇔ exp₁ + String.valueOf(exp₁)
Si exp₁ est une chaîne et exp₂ un objet
    exp₁ + exp₂ ⇔ exp₁ + exp₂.toString()
(tous les objets possèdent la méthode String toString())
```

H. Garreta, 2013 IHM et Java 10 / 13

le membre est accessible dans

Droits d'accès aux membres

Quatre niveaux:

© H. Garreta, 2013

Java pour les programmeurs C++

Variables et constantes d'instance et de classe

```
public class Truc {
                                                // variable d'instance
       int valeur;
                                                // constante d'instance
       final int rang;
                                                // variable de classe
       static int nombre;
                                                // constante de classe
       static final int VALEUR_MAX = 1000;
       public Truc(int val) {
                                                // constructeur de la classe
           assert val < VALEUR_MAX;</pre>
           valeur = val;
           rang = nombre;
           nombre++;
       }
   }
Exemple de construction d'un objet :
   Truc p = new Truc(125);
```

© H. Garreta, 2013 IHM et Java 13 / 13