Une introduction à la conception objet avec UML et Java

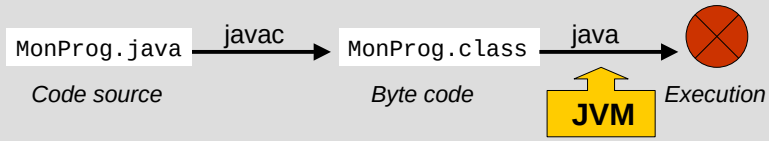
Licence professionnelle

UV G5a

Pascal Divoux ; Cedric Wemmert

Caractéristiques

- Java est portable
  - ➔Java est interprété, il produit du code (byte code) indépendant de la plate-forme et interprétable par une machine virtuelle : la Java Virtual Machine (JVM).
  - ➔Java introduit un niveau intermédiaire qui établit une indépendance par rapport à la machine physique. La JVM est présente sur Solaris, Win32, Mac, linux ...et dans les navigateurs récents
  - ➔Un programme Java est ainsi exécutable dans n'importe quel environnement disposant d'une JVM.



Syntaxe

➢Syntaxe

- Les commentaires

➔Il existe 3 manières de créer des commentaires :

```
♦    /* un commentaire sur
       plusieurs lignes */
♦ // un commentaire simple en fin de ligne

♦  /**un commentaire pour la javadoc */
```


Syntaxe

- Les identificateurs
  - ➔ Suite de caractères alphanumériques, le premier étant un alphabétique
  - ➔ Distinction minuscules/majuscules
    - ♦ `somme est différent de Somme`
  - ➔Conventions de nommage :
    - ♦ `variables et méthodes commencent par une minuscule`

      `ex : int x, float total;`
    - ♦ `Les noms de classes commencent par une majuscule`

      `class Puissance`
    - ♦ `Les majuscules intérieures sont utilisées pour séparer les mots, évitez les - et les _`

      `ex : totalDesSalairesPayes`
  - ➔ Mots réservés :

Attention, les mots réservés du langage java ne peuvent être utilisés comme identificateurs (if, long, class, char....)

Syntaxe

Les types de base

- ➔ les booléens (true ou false) : `boolean trouvé;`
- ➔ les caractères : `char c;`
  - *codés sur 16 bits (unicode)*
- ➔ les entiers : ils sont signés
  - `byte : 8 bits (de -128 à 127)`
  - `short : 16 bits (de -32768 à 32767)`
  - `int : 32 bits (de -(2 puiss 31) à (2 puiss 31)-1)`
  - `long :64 bits (de -(2 puiss 63) à (2 puiss 63)-1)`
- ➔ les réels :
  - `float : 32 bits [1.402e-45, 3.402e+38]`
  - `double : 64 bits [4.94e-324, 1.797e+308]`

Syntaxe

Le type « chaîne de caractères »

- ➔ String : `String ch;`
  - *C 'est une classe du langage java (majuscule initiale)*

Affectation :

- *booléens*
  - `trouve = false; // ou true`
- *constantes caractères entre simple quote*
  - `char c = 'a' ;`
- *entiers*
  - `int i = 12;`
  - `float f = 13.5 ;`
- *constantes chaînes entre double quotes :*
  - `String ch= ''bonjour ''`

Syntaxe

Déclaration et portée d'une variable

- ➔ Une variable doit être déclarée et initialisée avant son utilisation
  - `int n = 0;`
  - `String nom=''toto'';`

  Exception : les attributs de classe ont une valeur par défaut (zero, caractère et chaîne vides, false)
- ➔ La portée d 'une variable est limitée à son bloc d 'appartenance et à ses sous-blocs

  Un bloc est un groupe d 'instructions délimitées par des accolades

  `{.....}`
- ➔ Si la variable est redéclarée dans un sous-bloc c 'est cette nouvelle déclaration qui s 'applique dans le sous-bloc (et ses sous-sous-blocs) : masquage

Syntaxe

Les opérateurs

- ➔ arithmétiques

  `+ - * / % pow(x,y) et autres fonctions`
- ➔ affectation

  `= +=`
- ➔ comparaison

  `== != < > <= >=`
- ➔ logiques

  `&& (et) || (ou) ! (not)`
- ➔ de chaîne

  `+ (concaténation)`

Attention à la confusion entre = et ==

et entre somme et concaténation (surcharge d 'opérateur) si l 'un des 2 opérandes est un String : concaténation

Structures de contrôle

conditionnelle

```
if (expression booléenne){bloc1} [else {bloc2}]
```

la partie else est optionnelle

aiguillage

```
switch (expression) {
  case expr1 : {bloc1}
  break;
  case expr2 : {bloc2}
  default : {bloc3}
}
```

Le « break » entraîne la sortie directe du switch sinon évaluation du cas suivant

« default » est exécuté si aucun cas n'a été trouvé

Itération for

- `for (initialisation ; test-while ; incrément) {bloc}`
- `ex : for (int i=0 ; i<10 ; i++) {System.out.println(i);}`

Un « break » entraîne la sortie directe de la boucle

Itération while

- `while (expression-bool) {bloc}`

L 'expression est évaluée avant l 'entrée dans la boucle

- `do {bloc} while (expression-bool);`

Le bloc est exécuté au moins une fois, l 'expression est évaluée en sortie de boucle

Entrées/Sorties

Affichage sur la sortie standard (console)

- `System.out.print(chaineDeCaracteres);`
- `System.out.println(chaineDeCaracteres);`

La 2eme version ajoute un retour à la ligne

- Ex :

```
System.out.print(‘‘ Bonjour ’’);
System.out.println(leNom+ ‘‘  ’’ +lePrenom);
```


Lecture sur l'entrée standard (clavier)

Les opérations de lecture ne font pas partie du langage java, deux solutions :

Pour des entrées/sorties sommaires on utilisera la classe « Console » du package iutsud développé à l 'IUT

➔ opération d 'import (cf diapo suivante)

```
import iutsud.Console;
```

➔ readBoolean(), readInt(), ...readLine()

```
String texte=Console.readLine();
```

NB : Une seule valeur par ligne

Pour des entrées/sorties plus élaborées on utilisera les classes Swing (vues plus tard)

```
import javax.swing.*
```

Structure d 'un programme

- En java, tout est dans une classe (pas de fonctions à part)
- Le fichier source « `Puissance1.java` » a le même nom que la classe principale « `Class Puissance1` »
- La première instruction précise les classes à importer (cf suite) ; java.lang est importé d 'office
- Déclaration de la classe :
  - `class Puissance1 {...}`
- Déclaration des variables et des méthodes de la classe
  - `public static int puissance(int a, int b) {....}`
- Définition de la méthode principale
  - `public static void main (String[] args) {...}`

Présentation : Indentation des programmes (accolades)

Packages

- Un package est un ensemble de classes sémantiquement proches (swing, iutsud, java.util,...)
- La première instruction de chaque fichier de classe précise le package de la classe (sinon package « anonymous »)
  - `package monPack;`
  - `class MaClass {...`
- Tous les .class du package doivent être dans un répertoire qui a le même nom que le package
- Compiler
  - `javac -d . *.java`

crée automatiquement le répertoire « monPack »

- Utiliser
  - `java monPack.MaClasseMain`

Le répertoire contenant le package doit figurer dans le Classpath

- Pour utiliser une classe dans un programme, il faut donner son chemin d 'accès

➔ depuis la racine de l 'arborescence java

```
java.util.Date aujourdhui;
javax.swing.event.ChangeEvent.getSource();
```

➔depuis java.lang (package standard de java)

```
String s;
```

- La directive « import » permet d 'utiliser une ou plusieurs classes sans avoir à redonner leur chemin complet lors de leur utilisation

```
import java.util.Date;
import java.util.*;
Date aujourdhui = new Date();
import iutsud.Console;
Console.readInt();
```

NB : le package importé doit être situé dans un répertoire figurant dans le CLASSPATH