



# Le langage Java

Introduction

Vocabulaire, types de programmes, packages

# Programme détaillé ou sommaire

Vocabulaire

Types et structures de programmes

Les packages

# Chapitre 1

## Java – Vocabulaire

Vocabulaire

# JRE

**JRE** = Java **Runtime** Environment

A télécharger et à installer sur votre machine/PC.

Minimum requis pour **exécuter** un programme java.

Contient notamment **la commande** permettant d'exécuter un programme Java.

**java monProgramme**

# JVM

**JVM** = **J**ava **V**irtual **M**achine

La commande java, qui permet d'exécuter votre application, lance une JVM.

La JVM désigne l'ensemble des processus qui tournent en arrière plan pour gérer l'exécution de votre application:

- interactions avec le système d'exploitation
- gestion de la mémoire
- optimisation du code à la volée.

# JDK

Java Development Kit = **JRE** + outils de développement

## Outils de développement

- Compilateur

- Debugueur

- Générateur de documentation

## Bibliothèques de classes

## Documentation

- Des outils

- Des classes

# Chapitre 2

## Structure d'un programme

Structure d'un programme en Java

# Structure des programmes

## ❑ Code java

- Physiquement : des **fichiers** `.java` (texte) organisés en **répertoires**
- Logiquement : des **classes** organisées en **packages**

❑ On peut écrire du code Java avec un simple éditeur de texte (vi ou Notepad).



# Exemple de programme

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

*NB 1 : les majuscules sont importantes comme la plupart des signes*

*NB 2 : le code sera plus amplement décrit plus tard*

# Chapitre 3

## Les packages

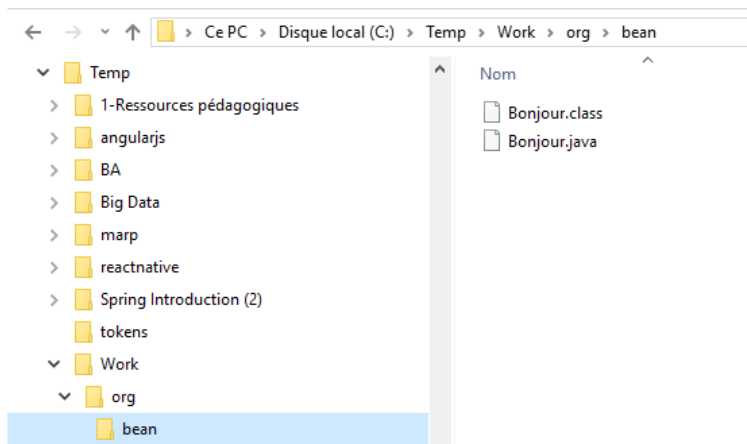
Classification

Rôle

Utilisation

# Packages – qu'est ce qu'un package

- ❑ Un package est un élément d'organisation, de rangement des classes.
- ❑ Un package correspond à un répertoire physique



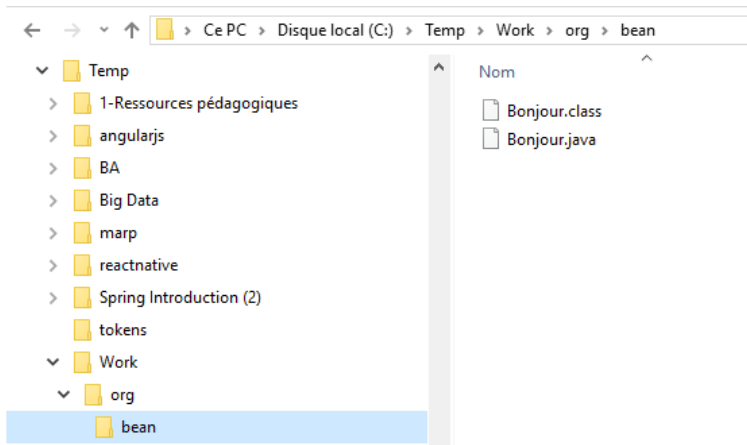
```
package org.bean;

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

- ❑ Ci-dessus le package **org.bean** correspond à un répertoire physique **/org/bean**

# Packages – Racine d'un projet

- ❑ La racine du projet est le **répertoire** qui contient le **package racine**.
- ❑ La commande java (run) est lancée depuis ce répertoire
  - On se positionne dans le répertoire **work**
  - On exécute la commande : **java org.bean.Bonjour**



```
package org.bean;

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

# Packages - rôle

- ❑ Permet de ranger les classes par
  - Responsabilité
  - Fonctionnalité
  - Etc.
  
- ❑ Chaque société/projet a ses normes de découpage en packages

# Chapitre 6

# Compilation

Variables d'environnement

Commande javac

Classpath

Options de compilation

# Commande javac

- ❑ Produit les fichiers compilés (**.class**) à partir des sources (**.java**)

- ❑ Ligne de commande


**javac Fichier.java**

- ❑ Possibilité de compiler tous les fichiers du répertoire :

**javac \*.java**

# Packages - utilisation

**C:/Temp/work/org/bean/Bonjour.java**



```
package org.bean;

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

Pour compiler Bonjour.java, il y a **2 possibilités**:

- 1) se placer dans le répertoire **C:/Temp/work/org/bean** et taper **javac Bonjour.java**.
- 2) se placer dans le répertoire **C:/Temp/work** et taper **javac org/bean/Bonjour.java**



# Chapitre 8

# Documentation

Documentation du code

Générateur de documentation

Documentation Officielle

# Documentation du code

## Commentaires sur plusieurs lignes

Encadrés par `/*` et `*/`

Pour les détails techniques

## Commentaires sur une seule ligne

Commence par `//`

Pour les détails au milieu du code

## Commentaires javadoc

Encadrés par `/**` et `*/`

Pour la documentation technique

Syntaxe normalisée

Automatiquement repris par les outils de production de documentation

# Générateur de documentation

Reprend les commentaires encadrés par `/**` et `*/`

Ligne de commande

```
javadoc Fichier.java
```

Nombreuses options (voir `-help`)

Documentation au format HTML

Possibilité d'inclure du code HTML

Possibilité d'inclure des tags dans les commentaires (`@tag`)

```
@author jdurand
```

```
@see com.java.bankonet.Client
```

```
@version 2.3, 03/06/2015
```

Documentation API JDK elle-même générée par javadoc

# Documentation sur Java 11

Site principal : <https://docs.oracle.com/en/java/javase/11/>

API complète des classes de base

Outils standards du JDK (javac, java, jdb,...)

Exemples

Tutoriaux

Changements entre les versions

...

# Rappels des points clés

Soit la classe **Bonjour.java** située dans **C:/Temp/work/org/bean/**

La classe **Bonjour.java** doit posséder en première ligne: **package org.bean;**

## Pour compiler:

Se placer dans le répertoire **C:/Temp/work/**

Exécuter la commande suivante: **javac org/bean/Bonjour.java**

## Pour exécuter:

Se placer dans le répertoire **C:/Temp/work/**

Exécuter la commande suivante: **java org.bean.Bonjour**

# Atelier (TP)

OBJECTIFS : installer les outils, préparer son environnement de développement

DESCRIPTION :

TP n°1 – Dans ce TP vous allez installer le JDK 17.

TP n°2 – Dans ce TP vous allez installer l'IDE Spring Tool Suite 4.14