



Une entreprise ordinaire à l'agilité extraordinaire

FORMATION JAVA SPRING ANGULAR

JAVA STANDARD EDITION 8

mohamed.el-babili@fms-ea.com

33+ 628 111 476

Version : 2.0

DMAJ : 13/01/23

Sommaire

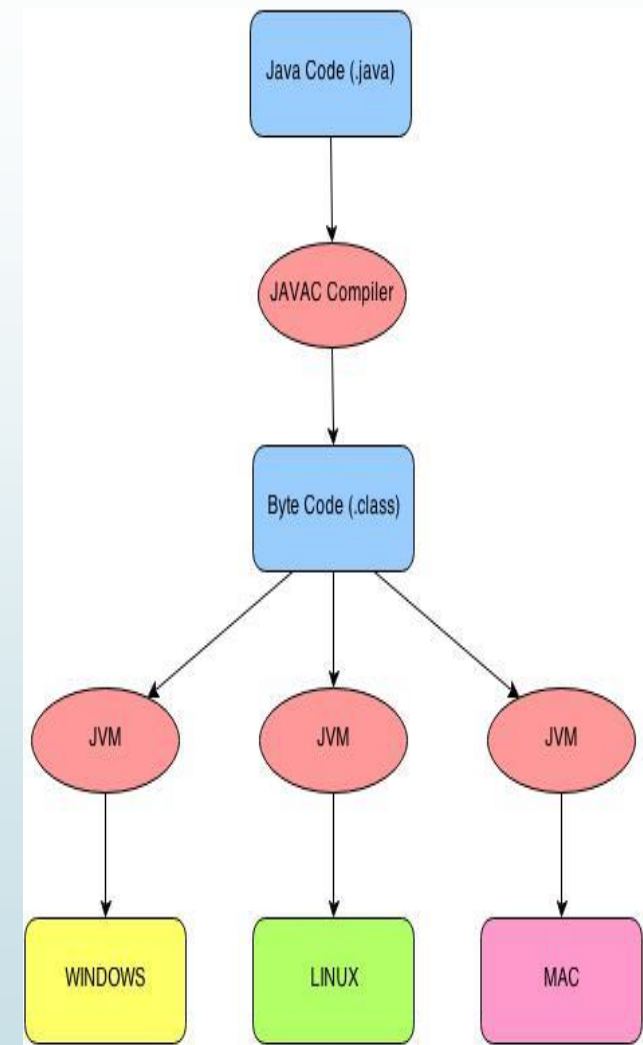
2

- Historique du langage
- Forces de Java
- Chiffres clés de la communauté Java
- Premier programme Java en ligne de commande
- Prise en main d'Eclipse
- Bonnes pratiques et convention d'écritures
- Les portes logiques
- Les types primitifs et les opérateurs de données
- Les types caractères et chaînes de caractères
- Manipulation des structures de contrôle et tableaux
- Méthodes statiques & notion de portée/package
- L'algorithmique
- Comment exporter/importer un projet ?
- Ressources
- Next Steps

Historique du langage

3

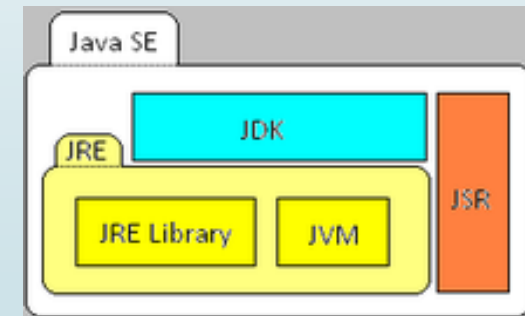
- Comme beaucoup de langages, Java est né d'un besoin. En 1991 Sun Microsystems cherchait à développer, des applications pour systèmes embarqués sur des appareils électroménager.
- D'abord appelé Oak(chêne) déjà utilisé, le nom changea en Java « sois disant » parce que la boisson préférée des développeurs était un arabica en provenance de l'île de Java.
- Fortement inspiré du C++, on considère que le père de ce langage est James Gosling.
- Ce langage ne connut pas un franc succès avant l'arrivée du web qui l'a fait littéralement décoller grâce à son atout multi plateforme.
- En 2009 Oracle rachète Sun.





Forces de Java

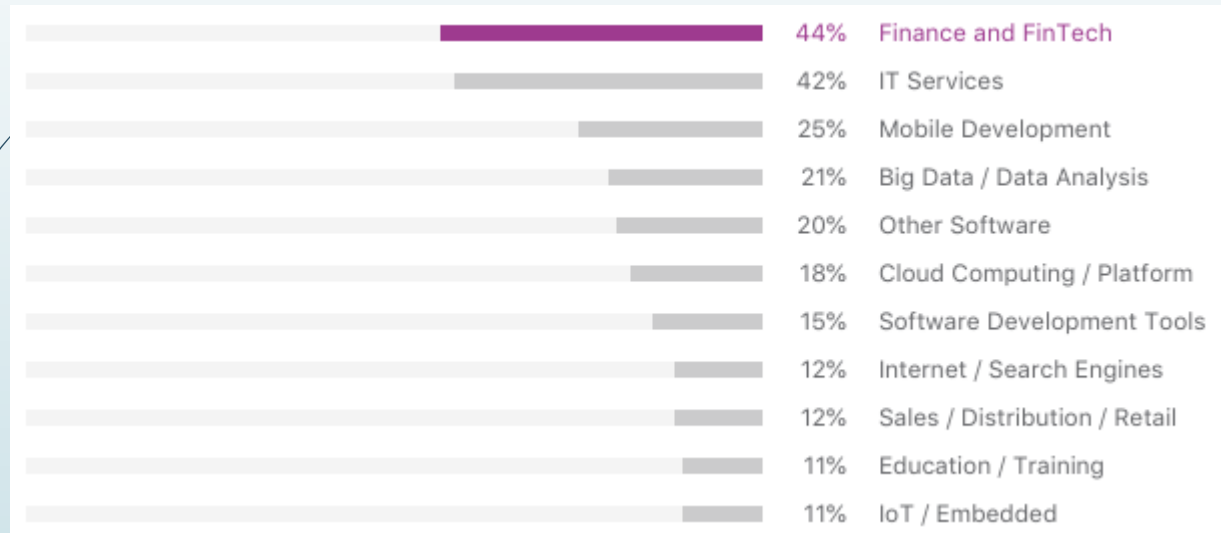
- Java est un véritable langage Objet
- Portabilité
- Api Java
- Garbage Collector
- Multi threading
- Robustesse (sécurité...)
- Nombreuses ressources ici
- C'est quoi la JVM/JRE/JDK/JSE/JEE ?
 - Oak(1996) ... J2SE(1998/2002) ... Java SE 5(2004) ... Java SE 8 LTS (2014) ... JSE 11(2018) ... JSE 17 (2021)
 - Nouveautés de Java 8 (JCP → JSR 337)
Expressions lambda, Stream, ForEach...



chiffres clés de la communauté Java



[Voir l'étude complète ici](#)



Premier programme Java en ligne de commande

6

- Pour que ça marche, il faut avoir installer quoi ?
- Saisir un programme simple « Hello.java » dans un éditeur
 - Ajouter un dossier sur le bureau par ex « TestJava »
- Compiler le programme avec la commande « javac Hello.java »
 - un fichier Hello.class a été généré
- Exécuter le programme avec la commande « java Hello »
 - un message s'affiche en ligne de commande : « Bonjour ! »
- Passez en argument « votre prénom » à la ligne de commande pour obtenir :
 - « bonjour tintin »

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

Nb : Sous windows, il est nécessaire d'ajouter le chemin(path) pour exécuter les [commandes](#)

+ pour passer des arguments en ligne de commande avec Eclipse, il faut aller dans Run config/argument

Prise en main d'Eclipse

7

- C'est quoi un IDE ? Quels sont les avantages ? Citez en ?
- Particularités d'Eclipse : Écrit en Java, ajout de plugin(c, c++, java)...
- Les autres IDE pour Java : NetBeans, IntelliJ IDEA, VSC...
- Arborescences : WorkSpaces / Perspectives / Projets / Classes / Vues
 - Ajouter le dossier « JavaSE-workspace »
- Refaire notre premier programme avec Eclipse
 - Ouvrir Eclipse sur notre workspace puis créer un nouveau projet Java
 - choisir un nom approprié « ProjetHello » par exemple
 - Cocher JRE/Use execution env JavaSE-1.8
 - dans l'explorateur de projets/clic droit sur le projet/new Class
 - saisir le nom « Hello » et cocher « public static void main... » (génération automatique du point d'entrée de votre application)
 - dans le main, ajouter : « System.out.println(« Bonjour »);
 - exécuter par clic droit sur la classe Hello.java/Run as java app

Bonnes pratiques et convention d'écritures

8

- Pour partir sur de bonnes bases, il est important de se mettre d'accord sur la manière de travailler :
- Règle N° 1 : une classe publique doit être dans le fichier du même nom sans l'extension, avec la première lettre en majuscule, on peut avoir des noms composés, ex : « CurrentAccount »
- Règle N° 2 : indentation de code
- Règle N° 3 : utilisation de l'anglais
- Règle N° 4 : ajouter des commentaires
- Règle N° 5 : choix des noms de variables

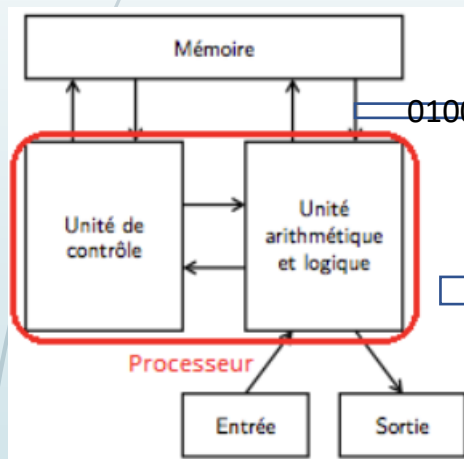
Les portes logiques

9

Le processeur est le cerveau de l'ordi :

- Il organise les échanges de données entre les différents composants (Hdd, Ram, carte graphique, périphériques...)
- Il fait les calculs qui font interagir l'ordi avec l'utilisateur et affiche le SE à l'écran.
- Exécute les instructions d'un programme placé dans la mémoire...

L'unité arithmétique et logique permet de répondre au besoin de représenter des préoccupations humaines telles que lorsque j'appuie sur le bouton de la machine à café, s'il y a de l'eau et du café, on passe à l'étape suivante.



010000010100001001000011

65	01000001	101	41	A
66	01000010	102	42	B
67	01000011	103	43	C

Les portes logiques servent à obtenir une sortie précise (ou chemin) en fonction des données en entrée et du filtre utilisé

La porte logique NOT (non)

A	S
0	1
1	0

La porte logique AND (et)

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

La porte logique OR (ou)

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

[J'en veux plus](#)

Les types primitifs et les opérateurs de données

10

Devinez les résultats ?

```
public class Test {
    public static void main(String[] args) {
        int counter = 5;
        String name = "Nina";
        double pi;
        int x = 10 , y = 5;

        pi = 3.1415;

        System.out.println( counter );
        System.out.println( name );
        System.out.println( pi );
        System.out.println( x + y);
        System.out.println( "x = " + x + " y = " + y );

        //-----après les variables, voici les opérateurs arithmétiques
        int firstValue = 11;
        int secondValue = 3;
        int div = firstValue / secondValue;
        int rest = firstValue % secondValue;

        int computedValue = div * secondValue + rest;
        System.out.println( "11 / 3 == " + div );
        System.out.println( "11 % 3 == " + rest + '\n');
        System.out.println( "computedValue == " + computedValue );

        /*----- opérateurs de
                    comparaisons-----*/
        boolean firstBoolean = true;
        boolean secondBoolean = false;

        if(firstBoolean == secondBoolean) System.out.println("les booléens sont identiques");
        else if(firstBoolean != secondBoolean) System.out.println("les booléens ne sont pas les mêmes");

        if(firstBoolean == true && secondBoolean == false) System.out.println("c'est cool");

        if(firstBoolean == true || secondBoolean == true) System.out.println("c'est super cool");

        //----- opérateur post ou préfixé
        int value = 10;
        System.out.println( ++value );
        System.out.println( --value );
        System.out.println( value++ );
        System.out.println( value-- );
    }
}
```

Les types caractères et chaînes de caractères

11

Comprendre l'encodage,
voir la table ascii

Devinez les résultats ?

```
//import java.lang.Character;
//import java.lang.String;
//import java.lang.Double;

public class Test {

    public static void main(String[] args) {

        char a = 'a';
        char b = 97;
        char c = '\u0061';
        char d = '\n';
        char e = 'e';

        System.out.print( a + "->" + b + "->" + '\t' + c + "-> \n" + d + "->" + e + "\n");
        System.out.println(Character.isLetter( a ));

        String firstName = "luke";
        String lastName = "skywalker";
        System.out.println(firstName.toLowerCase() + " " + lastName.toUpperCase());
        java.lang.String fullName = firstName;
        fullName += " " + lastName;
        System.out.println(fullName);

        String intrus = firstName;

        if(intrus.equals("luke")) System.out.println("les chaines sont égales ! ");
        else System.out.println("les chaines sont différentes !");

        System.out.println(firstName == "luke");
        System.out.println(firstName == new String("luke"));

        if(lastName.length() > 0) {
            System.out.println(lastName.substring(3));
        }

        // Types primitifs & conversion
        String strPi = "3.14";
        double value = java.lang.Double.parseDouble( strPi );
        value *= 2;
        String result = "pi * 2 = " + Double.toString(value);
        System.out.println(result);

    }
}
```

BIN	10010101
DEC	149
HEX	95

$$10010101 = (1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 149$$

Classe enveloppante
(wrapper classes)

Manipulation des structures de contrôle et tableaux...

12

Devinez les résultats ?

```
import java.util.ArrayList;
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        int [] myArray = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
        for(int i = 0 ; i < myArray.length ; i++)
            System.out.print(myArray[i] + " ");

        //System.out.println(myArray[10]);
        System.out.println();

        int [] array = new int[10];
        for(int i = 0 ; i < array.length ; i++)
            array[i] = i;

        String [] myStrings = { "monday", "tuesday", "happyday" };
        for(String str : myStrings) //équivalent du foreach
            System.out.println(str);

        ArrayList<String> coll = new ArrayList<>();
        coll.add( "lundi" );
        coll.add( "mardi" );
        System.out.println( coll.size() );
        for (String string : coll) {
            System.out.println( string );
        }

        /*long counter = 0;
        while(true) {
            System.out.println(" ----> " + counter++);
        }*/

        Scanner scan = new Scanner(System.in);
        int val = scan.nextInt(); //saisi au clavier de 5

        switch(val) {
            case 0 : System.out.println("ok0");
                    break;
            case 1 : System.out.println("ok1");
                    break;
            case 5 : System.out.println("ok5");
                    break;
            case 6 : System.out.println("ok6");
                    break;
            default: System.out.println("ok default");
        }
    }
}
```

Méthodes statiques & notion de Portée/package

13

JavaLdnr - ProjetStatic/src/fr/ldnr/demo/TestStaticMethod.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- ProjetoHello
 - JRE System Library [JavaSE-1.8]
 - src
 - (default package)
 - ControlArray.java
 - Hello.java
 - TypeCharString.java
 - TypeOperator.java
- ProjetStatic
 - JRE System Library [JavaSE-1.8]
 - src
 - fr.lndnr.demo
 - StaticMethod.java
 - TestStaticMethod.java
- StockSwingApp
 - JRE System Library [JavaSE-1.8]
 - src
 - fr.lndnr
 - App.java
 - fr.lndnr.bdd
 - fr.lndnr.entities
 - Category.java
 - Product.java
 - fr.lndnr.job
 - fr.lndnr.mvc
 - Controller.java
 - Model.java
 - View.java
- Referenced Libraries
- lib

StaticMethod.java

```
1 package fr.lndnr.demo;
2
3 import java.lang.Math;
4
5 public class StaticMethod {
6
7     private static double add(double a, double b) {
8         return a + b;
9     }
10    protected static double sub(double a, double b) {
11        return a - b;
12    }
13    public static double max(double a, double b) {
14        System.out.print("***");
15        return a > b ? a : b;
16    }
17    static double min(double a, double b) {
18        return a < b ? a : b;
19    }
20
21    public static void main(String[] args) {
22        double x = 5.5, y = 6;
23        System.out.println(add(x,y));
24        System.out.println(sub(x,y));
25        System.out.println(max(x,y));
26        System.out.println(min(x,y));
27        System.out.println(Math.max(x,y));
28    }
29
30 }
```

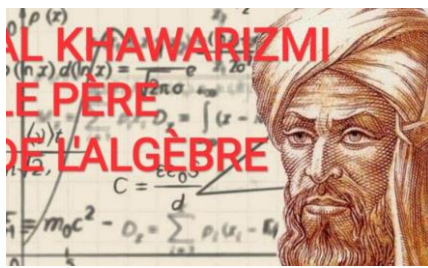
TestStaticMethod.java

```
1 package fr.lndnr.demo;
2
3 public class TestStaticMethod {
4
5     public static void main(String[] args) {
6         double x = 5.5, y = 6;
7         System.out.println(StaticMethod.add(x,y));
8         System.out.println(StaticMethod.sub(x,y));
9         System.out.println(StaticMethod.max(x,y));
10        System.out.println(StaticMethod.min(x,y));
11        System.out.println(Math.max(x,y));
12    }
13
14 }
15 }
```

Pourquoi c'est rouge ?

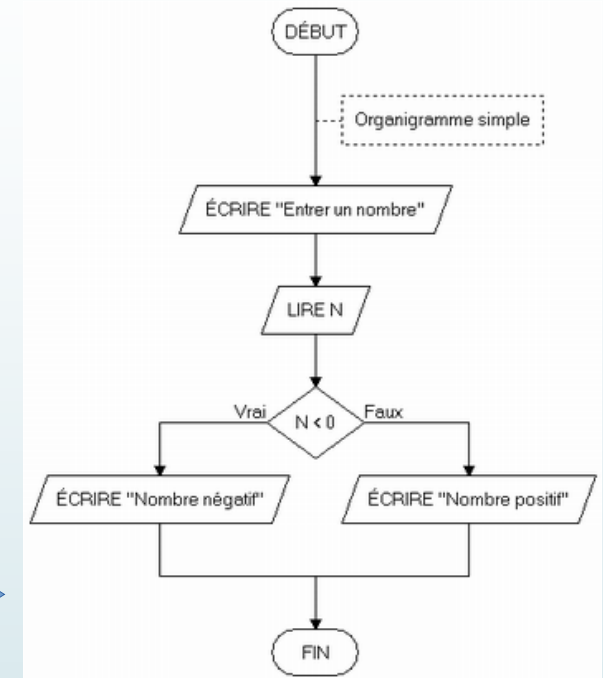
Devinez les résultats ?

Problems @ Javadoc Declaration Diagrams



L'Algorithmique

- **L'algorithmique** est l'étude et la production de règles et techniques qui sont impliquées dans la définition et la conception d'Algorithmes.
- Un **algorithme** est une suite finie clair d'opérations ou d'instructions permettant de répondre à un ou plusieurs besoins dans un langage informatique.
- **Exemple** : Au labo, on a besoin de savoir si une valeur, une fois saisie, est positive ou négative.



```

public static boolean posNeg(int n) {
    if(n >= 0) return true;
    else return false;
}

public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    int val = scan.nextInt();

    System.out.println(posNeg(val));

    //if(val >= 0) System.out.println("positif");
    //else System.out.println("negatif");

    scan.close();
}
  
```

```

Debut
|   Lire N
|   Si N > 0 alors retourner "nombre positif"
|   Sinon si N < 0 alors retourner "nombre négatif"
Fin
  
```

```

*****

Debut
|   Lire N
|   Si N >= 0 alors retourner "nombre positif"
|   Sinon retourner "nombre négatif"
Fin
  
```


Comment exporter/importer un ou plusieurs projets ?

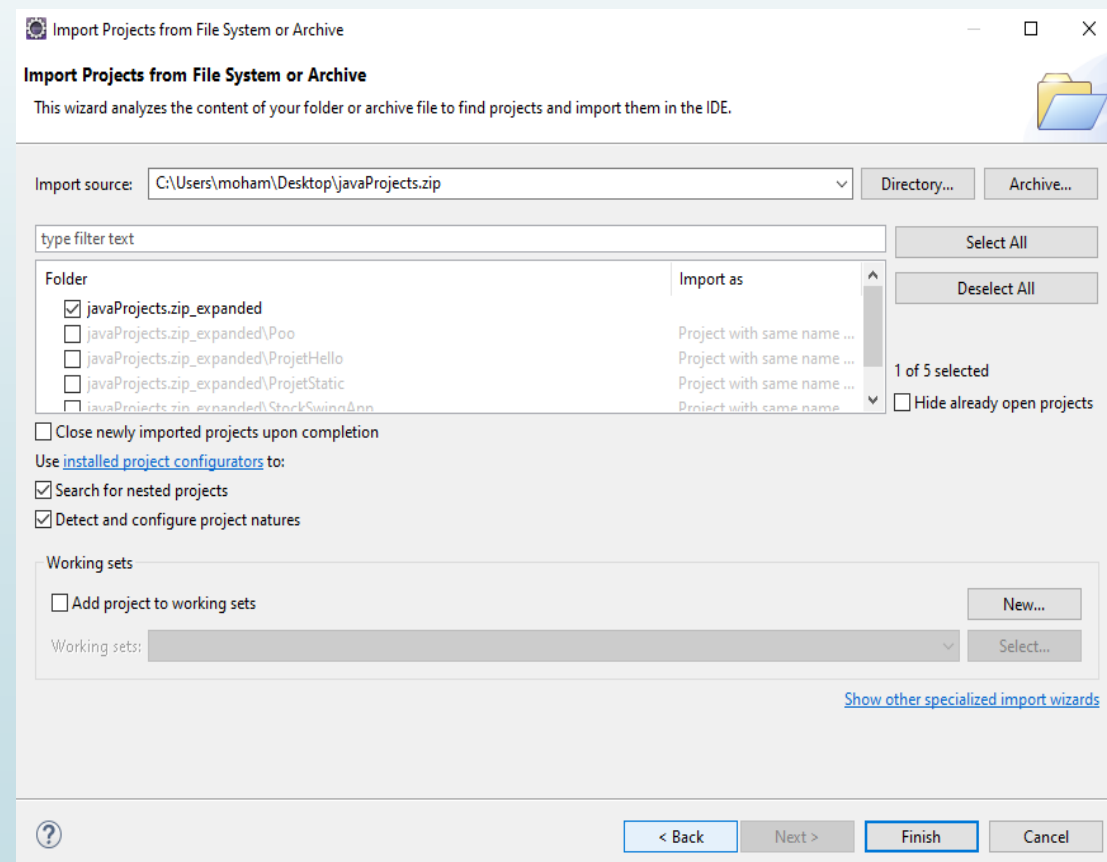
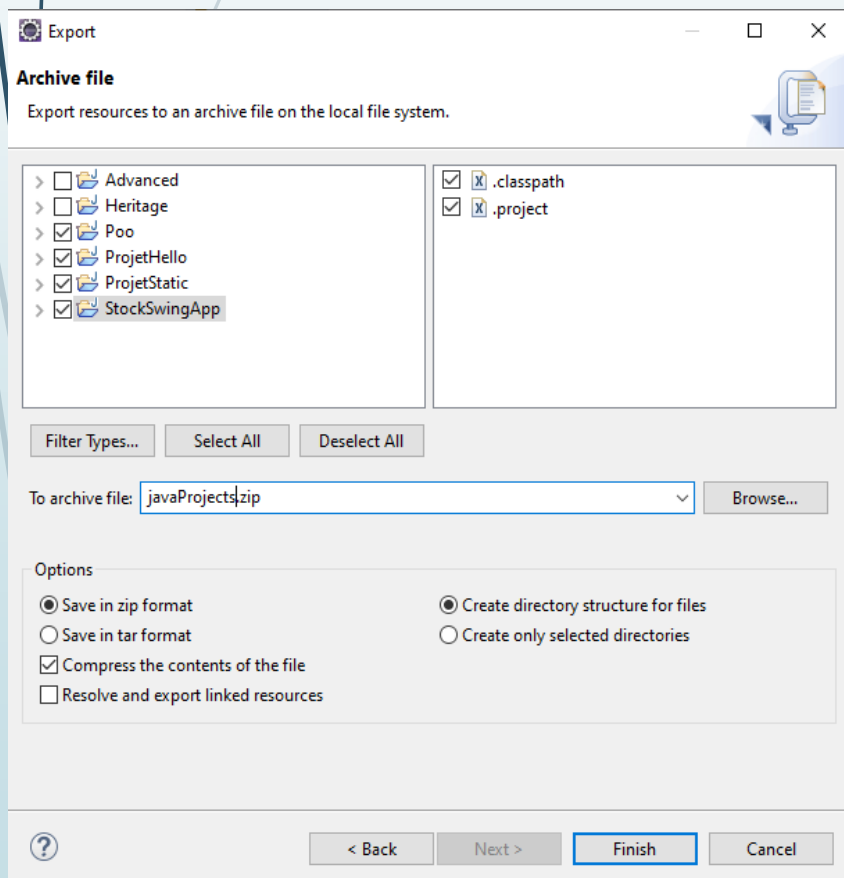
15

► Exporter ?

- File/Export/Archive File/...
- → select projects
- → nom du zip

Importer ?

- File/Import/Projects from Archive
- select archive
- nom du zip



Ressources

16

- <https://docs.oracle.com/javase/8/docs/api/>
- <https://www.tutorialspoint.com/java/index.htm>
 - <https://www.w3schools.com/java/>
 - <https://java.developpez.com/>
- <https://www.jmdoudoux.fr/java/dej/index.htm>
 - <https://koor.fr/Java/Index.wp>
 - <https://openclassrooms.com/>