**JAVA Standard**

* SE : Standard Edition
* LTS : Long Term Support
* Objet : une façon d’organiser des données
* Objet : structure de données (notion de groupe)
* Objet qui manipule des services (API)
* Pas de manipulation de pointeur : variables ou objet qui stockent des adresses mémoires
* Détection des erreurs lors de la compilation
* Typage fort
* Héritage simple (pas d’héritage multiple
* Java indépendant de l’architecture (programme exécutable partout contrairement à C++)
  + Fichiers sources 🡪 Compilation 🡪 ByteCode (+autres) 🡪 JVM 🡪  **Exécution**

(.class) javac (.class) (interprétation)

* Thread : tâche à faire
* Multi threading (parallélisme et synchronisation des ressources)
* API Java
  + Librairie de classes remplissant différentes fonctionnalités
  + Chaque classe regroupée en package
  + Toutes les API java sont documentées par un site HTML (javadoc)

**Syntaxe**

* Java est sensible à la casse
* Commentaire :
  + /\*commentaire\*/
  + //commentaire sur toute la ligne
  + /\*\*commentaire d’explication\*/ (se place avant une déclaration et visible dans javadoc
* Instructions
  + Se termine par « ; »
  + Pas d’instruction en dehors d’une méthode
* Bloc
  + { début
  + } fin
  + La définition d’une ou d’une méthode se fait dans un bloc
* Point d’entrée d’un exécutable java
  + Il faut absolument une classe contenant une méthode **main**
* Signature d’une méthode
  + La signature c’est le nombre, le nom et l’ordre des paramètres de la méthode
  + Les méthodes peuvent avoir le même nom en java
  + Il faut que chaque méthode de même nom aie une signature différente
* Identificateurs
  + Nom des classes, variables, objets, constants, …
  + Un identificateur doit commencer par une lettre, un \_ ou un $
* Règles de nommage des identificateurs
  + Les identificateurs s’écrivent en Camel Case
  + Classes et constructeurs : première lettre en majuscule
  + Tout autre type : première lettre en minuscule
  + Constantes : tous les caractères en majuscule et séparés par \_ (mots composés)
* Mots réservés :
  + Les types primitifs (boolean, byte, char, int, float, long, short, double,) :
  + Les décorateurs (class, package, import, new, null, this, …)
  + Les modificateurs (public, static, final, native, …)
  + Gérer les exceptions (Catch, finally, thrown, …)
* Les 8 types primitifs :
  + Caractère : char
  + Entiers : byte, short, int, long
  + Réels : float, double
    - Tout est objet sauf les 8 types de base en java

**Tableau**

* Déclaration : int tab[] ; String nom[] ;
* Création : tab= new int [20] ; nom= new String [100] ;
* Initialisation : tab [0]= 50 ; nom [0]= new String (‘’OK ‘’)

**Classe String**

* String s = ‘’aaa ‘’ ou String s = new String(‘’aaa’’)
* On peut concaténer deux objets de type String
* Attention
  + String str1 = ‘’abc ‘’ ; String str2 = ‘’abc ‘’ ;
  + Str1 == str2 🡺 false (car ici on compare deux adresses mémoires différentes)
  + Str1.equals(str2) 🡺 true (car in compare le contenu des deux adresses mémoires)

**JAVA orienté Objet**

* Nouvelle classe 🡺 nouveau type
* On ne crée pas d’objet, on crée une fabrique d’objet 🡺 un moule
* Une classe se compose de :
  + Caractéristiques : attributs
  + Comportements : méthodes
* Bonne pratique :
  + Une classe par fichier
  + Si jamais il y a plusieurs classes dans un fichier, le compilateur java crée plusieurs fichiers (.class)
* Pas d’instruction en dehors d’une méthode