

# Java

## Présentation

# Le langage Java

- Langage orienté objet
- Reprend en partie la syntaxe du C++
- Syntaxe et concepts de java repris lors de la création du langage C#
- Exécution basée sur une machine virtuelle

# Création du langage

- James Gosling
- Patrick Naughton
- Employés de Sun Microsystems
- Présentation officielle le 23 mai 1995 au SunWorld
- Oracle propriétaire de Java (rachat de Sun)























# Java : monde libre

- Depuis fin 2006, les outils java sont passés sous licence GPL : General Public License
  - liberté d'exécuter le logiciel, pour n'importe quel usage ;
  - liberté d'étudier le fonctionnement d'un programme et de l'adapter à ses besoins, ce qui passe par l'accès aux codes sources ;
  - liberté de redistribuer des copies ;
  - liberté de faire bénéficier à la communauté des versions modifiées.

# Vocabulaire Java

- **JAVA : Just Another Vague Acronym**
- **Plateformes JAVA**
  - **J2EE : Java 2 Platform Enterprise Edition**
  - **J2SE : Java 2 Platform Standard Edition**
  - **J2ME : Java 2 Platform Micro Edition**
    - Pour plateformes mobiles

# Un des langages les plus populaires

Language Rank	Types	Spectrum Ranking	Spectrum Ranking
1. Python	  	100.0	100.0
2. C++	  	98.4	99.7
3. C	  	98.2	99.4
4. Java	  	97.5	97.3
5. C#	  	89.8	88.7
6. PHP		85.4	88.7
7. R		83.3	86.0
8. JavaScript	 	82.8	81.9
9. Go	 	76.7	76.8
10. Assembly		74.5	76.0

Source : IEEE - 2018

# TOP 10 des langages demandés par les employeurs



Source : IEEE - 2018

# Exemple de code

```
public class Personne
{
    private String nom;
    private String prenom;
    private GregorianCalendar date_nais;

    public long calculAge()
    {
        long age;
        date_nais=new GregorianCalendar(1963,11,29);
        age=new GregorianCalendar().getTimeInMillis()-
date_nais.getTimeInMillis();
        age=age/1000/60/60/24/365;
        return age;
    }

    public void affichage()
    {
        System.out.println("nom : " + nom);
        System.out.println("prénom : " + prenom);
        System.out.println("âge : " + calculAge());
    }
}
```

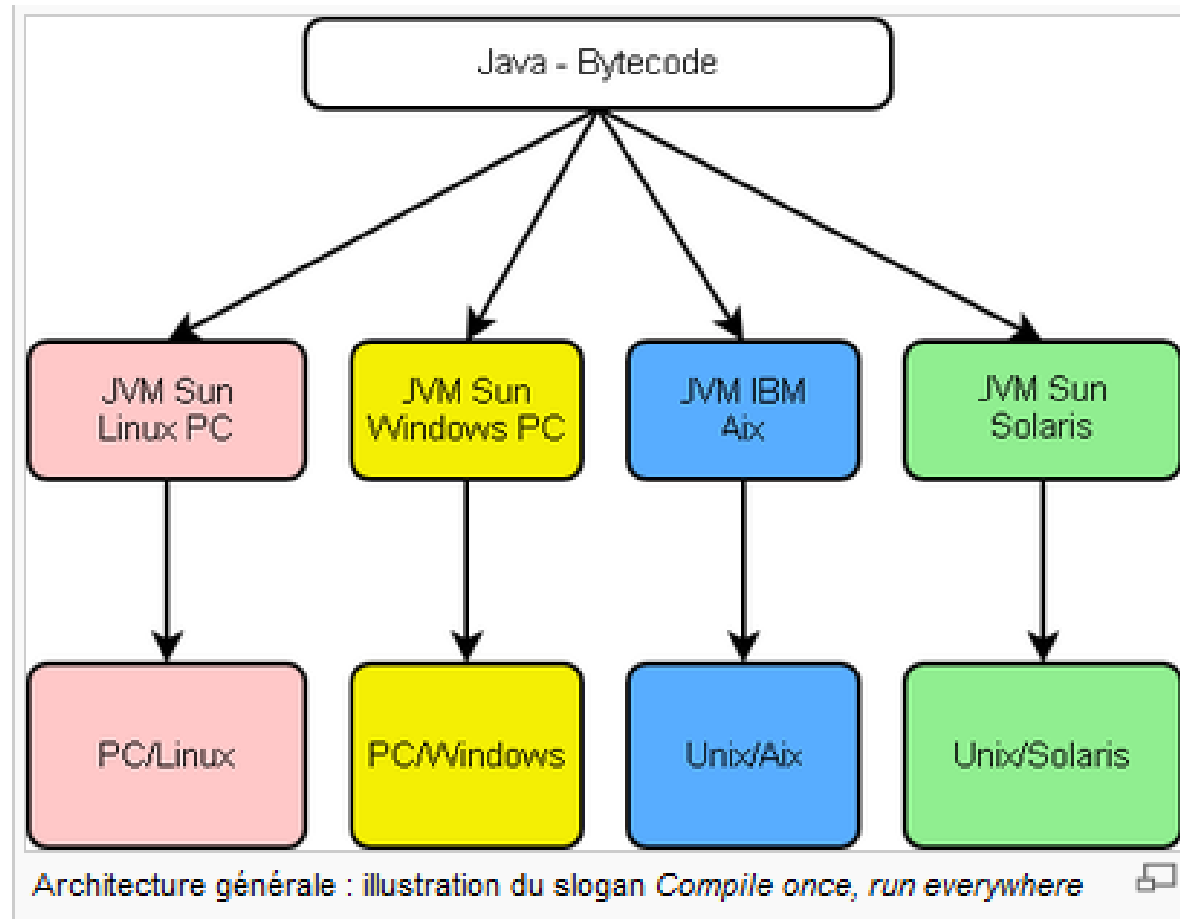


# Exécution des programmes

# La machine virtuelle

- Programmes java sont compilés dans un langage intermédiaire : le bytecode
- Le bytecode est exécuté par la machine virtuelle
- Permet une portabilité sur les différents OS

# La machine virtuelle



JVM : Java Virtual Machine

# La machine virtuelle

- Doit être installée sur les postes qui exécutent l'application
- La version installée doit correspondre à la version utilisée pour le développement

# Garbage collector

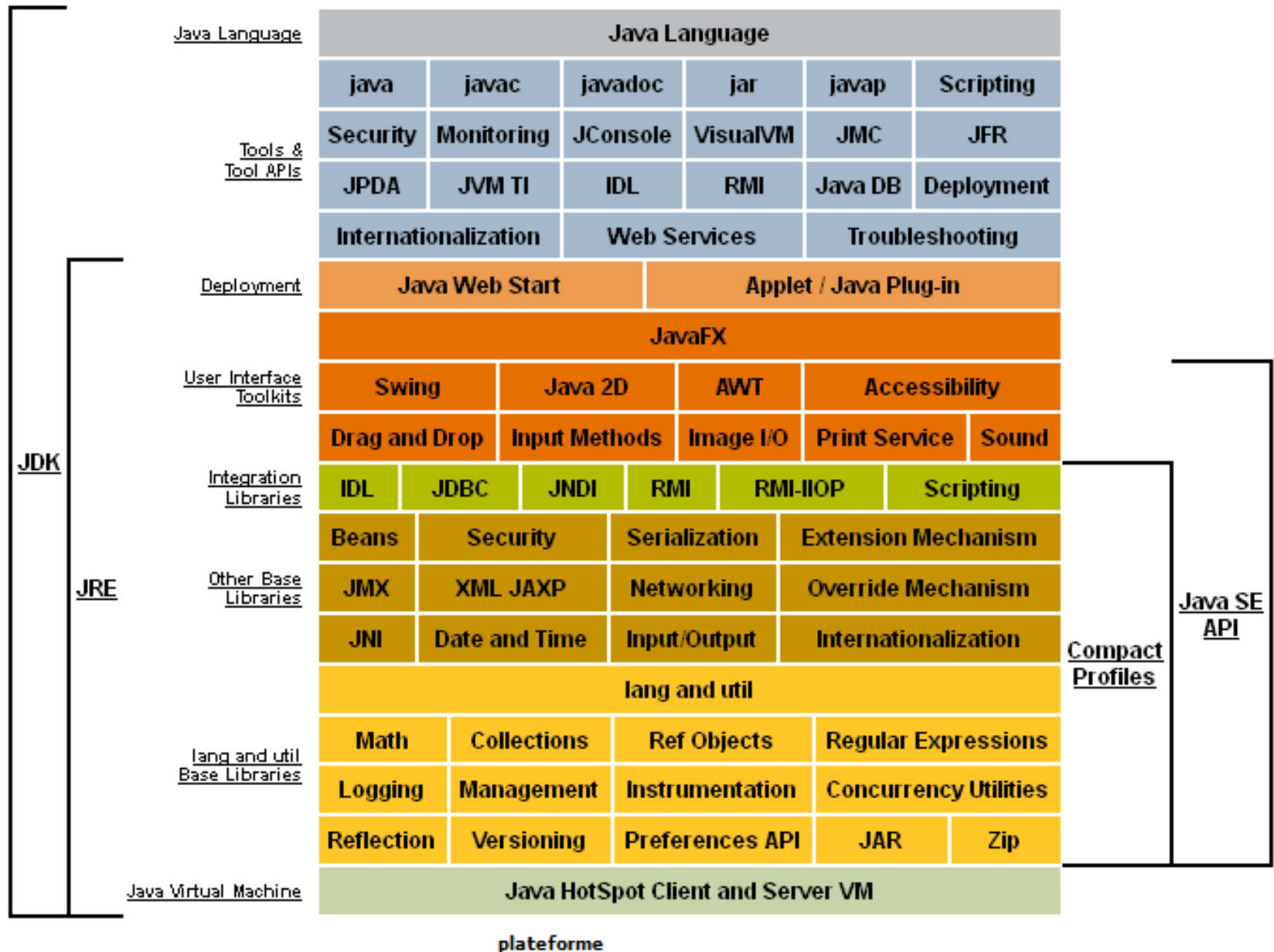
- Permet de gérer automatiquement la mémoire
- Un objet créé alloue de la mémoire pour stocker ses propriétés
  - Difficultés pour gérer la destruction des objets
- Libération de la mémoire gérée par le garbage collector

# Fonctionnement du garbage collector

- Les objets sont créés dans un memory heap (dans la machine virtuelle) avec une référence associée
- L'objet est supprimé lorsque toutes ses références ont été supprimées
- Evite les fuites mémoire, facilite la programmation

# Vocabulaire JAVA

- Packages d'une plateforme
  - **JDK : Java Development Kit**
    - Pour les développeurs : classes, documentation, compilateur...
    - Machine virtuelle
  - **JRE : Java Runtime Environment**
    - Pour exécuter les programmes java
    - Machine virtuelle





# Développement en Java

- Application console : avec fenêtres
- Applets : programmes incorporés à des pages web
- Applications pour appareil mobile...

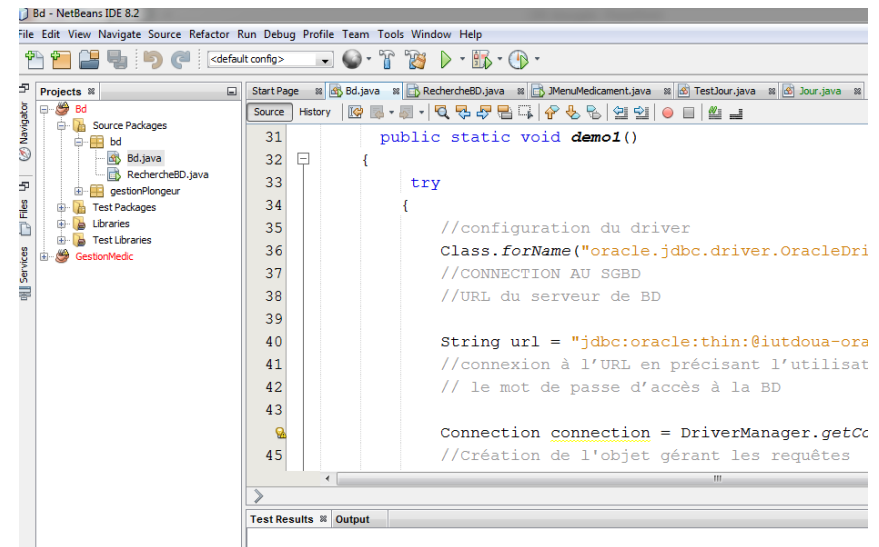
# Versions de java

- Actuellement, JAVA SE 11
- Version utilisée : JAVA SE 8
- Téléchargement de la JDK :  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Pour connaître la version java de son poste :
  - Aller dans l'invite de commande
  - Taper java -version

# Outil de développement



- Environnement de développement gratuit et open source
- <https://netbeans.org/>
- Nombreux plugin disponibles



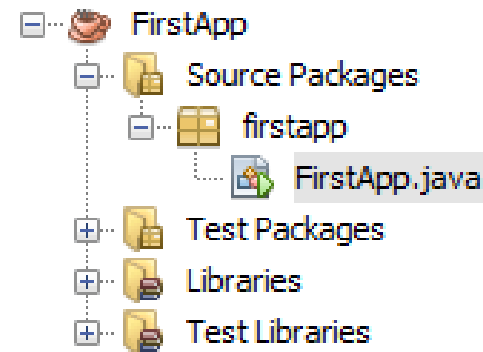
# Ecriture d'un programme

# Librairie Java

- Pour importer une librairie :
  - `import java.xxx.yyy;`
- Permet de pouvoir utiliser les classes définies dans la librairie

# Notion de package

- Organisation des programmes en package
- Création de classes à l'intérieur d'un package
- Les classes situées dans un même package peuvent accéder les unes aux autres



# Les packages

- Création : `package NomDuPackage ;`
  - En début de fichier
- Import : `import NomDuPackage.NomClasse;`

# Exécution d'un programme

- Point d'entrée de l'exécution : procédure main
  - `public static void main(String[] args) {}`
- Obligatoirement à l'intérieur d'une classe



# NETBEANS

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services

FirstApp

- Source Packages
  - firstapp
    - FirstApp.java
  - Test Packages
  - Libraries
  - Test Libraries
- JavaApplication2
  - Source Packages
    - javaapplication2
      - JavaApplication2.java
    - Libraries

Navigation projet

Start Page FirstApp.java JavaApplication2.java

Source History

```

10      * @param args the comma
11      * /
12      public static void main
13          // TODO code applica
14      }
15
16  }
17

```

Programme

main - Navigator

Members

JavaApplication2

- main(String[] args)

javaapplication2.JavaApplication2 main

Output

Debugger Console JavaApplication2 (run)

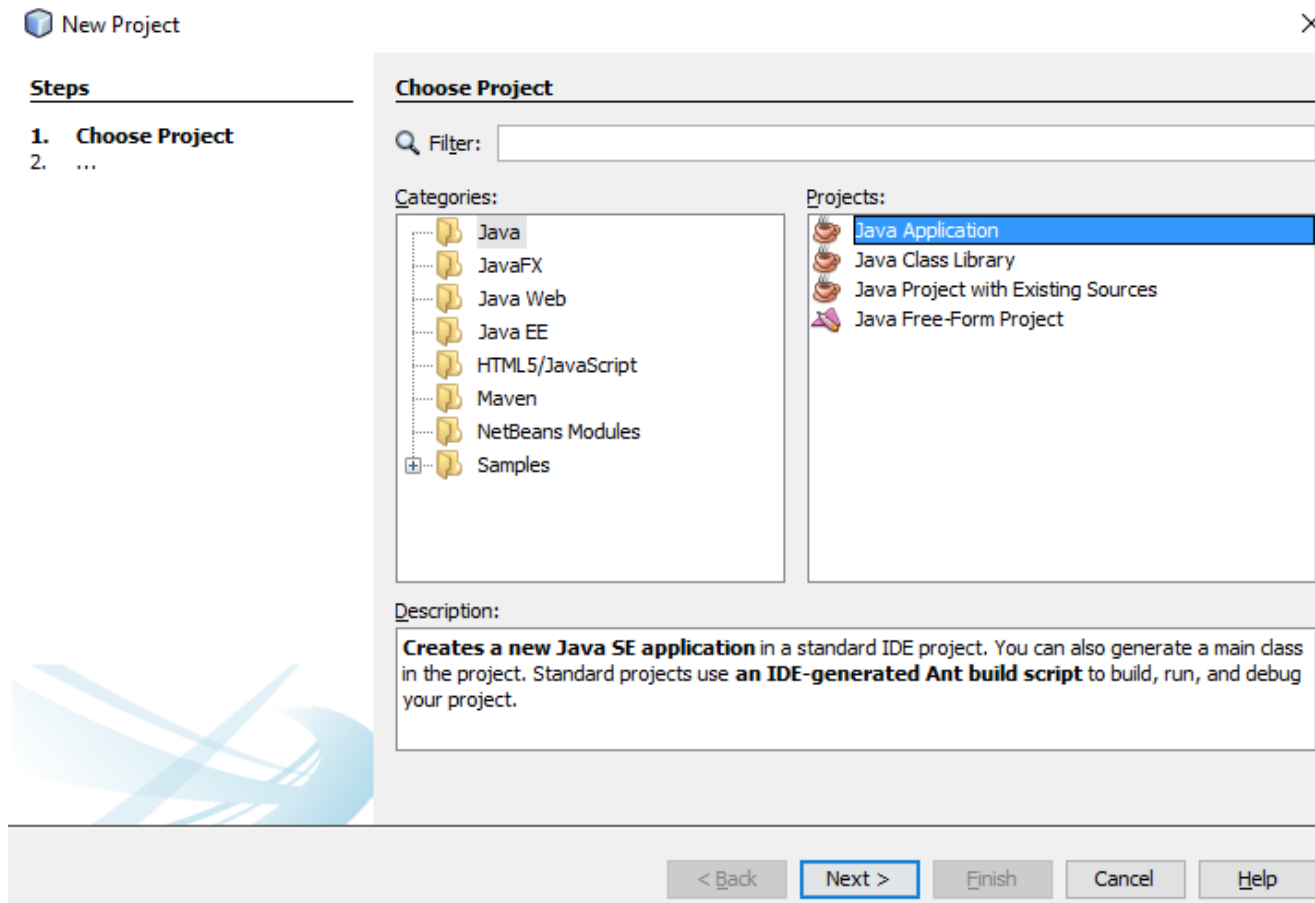
```

run:
BUILD SUCCESSFUL (total time: 0 seconds)

```

Exécution / Debug

# Création d'un projet



# Création d'un projet

New Java Application

Steps

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name:

Project Location:  Browse...

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:  Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

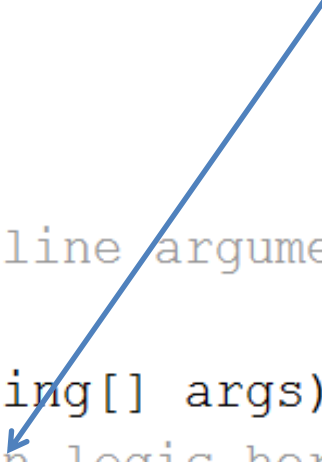
< Back Next > Finish Cancel Help

Nom du projet  
Nom de la première classe  
(contenant le main)

# Programme

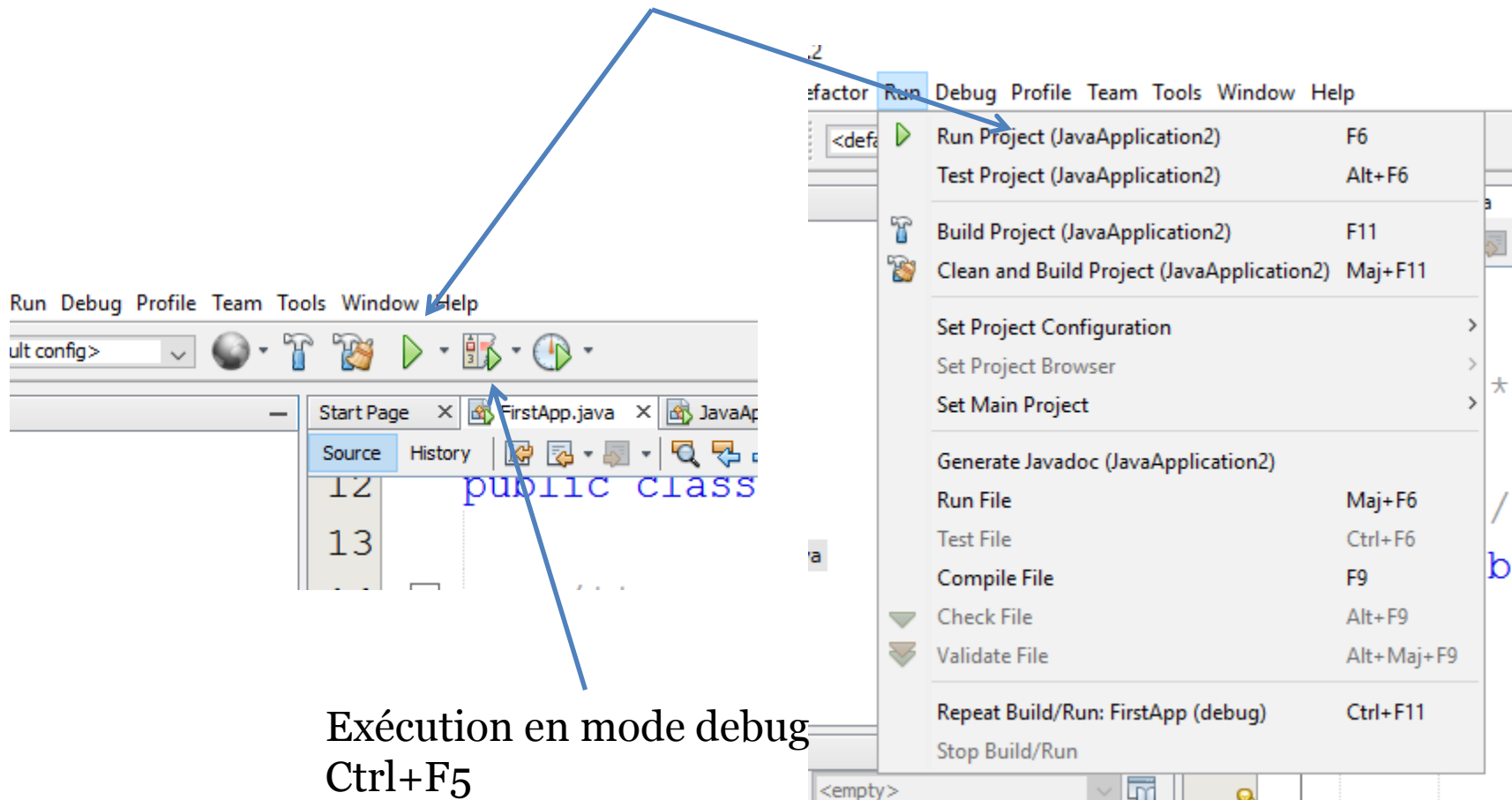
Emplacement du programme

```
package javaapplication2;  
public class JavaApplication2 {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
    }  
  
}
```



# Exécution du projet

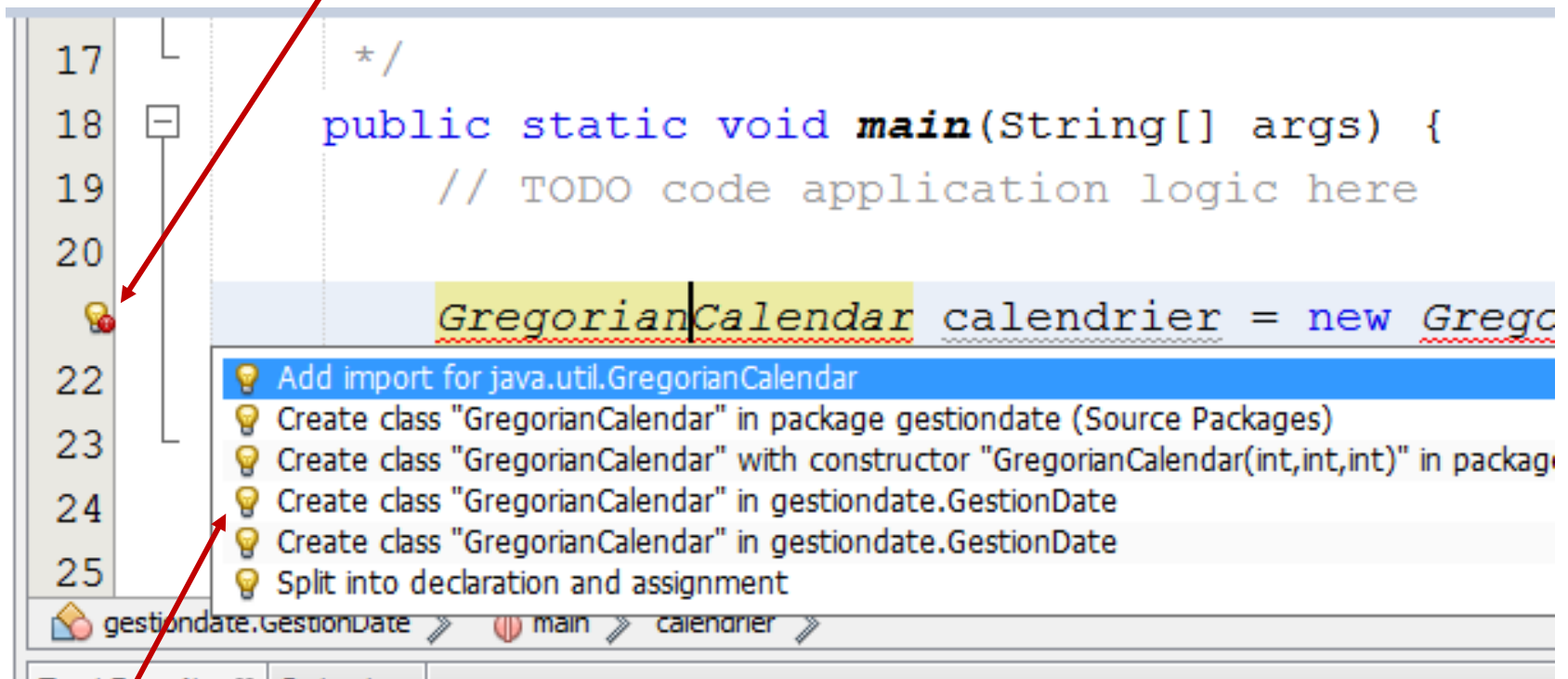
Exécution standard : F6



Exécution en mode debug  
Ctrl+F5

# Détection d'erreurs

Message d'erreur



Proposition de correction Alt+Entrée

# Test d'un programme



# Exercice saisie

- Dans le main, créer un tableau de 10 entiers et demander à l'utilisateur de saisir les différentes valeurs.
- Afficher le contenu du tableau

## Exercice fonction

- Créer une fonction concatTab qui prend en paramètre deux tableaux de cinq entiers. La fonction devra retourner un tableau de 10 entiers contenant les 5 entiers du premier tableau passé en paramètres, suivis des 5 entiers du deuxième tableau passé en paramètres.

## Exercice appel

- Dans le programme principal (main), créer 2 tableaux de 5 entiers contenant les chiffres de votre choix. Vous appellerez ensuite la fonction créée précédemment puis vous afficherez le contenu du tableau retourné.

## Autres fonctions

- A partir du tableau fusionné, on souhaite connaître la moyenne des entiers, le minimum et le maximum
- Créer la fonction moyenne qui retourne la moyenne des éléments d'un tableau
- Créer les fonction mini et maxi
- Appeler ces 3 fonctions dans le main

# Supplément

- Créer la fonction concatSansDoublon qui à partir de 2 tableaux passés en paramètres copie leurs valeurs en évitant les doublons et retourne le tableau sans doublons