

## Piscine java

Voici une série d'exercices progressifs en Java pour une semaine, avec une difficulté croissante chaque jour. Ces exercices permettent d'appréhender progressivement les bases du langage Java, en commençant par des concepts simples pour aller vers des sujets plus complexes.

### ### Jour 1 : Introduction aux bases

**\*\*Objectif\*\*** : Comprendre les bases de Java (types, variables, opérateurs, boucles, conditions).

#### 1. **\*\*Exercice 1 : Hello World\*\***

Créez un programme qui affiche "Hello World" dans la console.

#### 2. **\*\*Exercice 2 : Calculatrice basique\*\***

Demandez à l'utilisateur d'entrer deux nombres. Effectuez les opérations de base (addition, soustraction, multiplication, division) et affichez les résultats.

#### 3. **\*\*Exercice 3 : Parité\*\***

Demandez à l'utilisateur d'entrer un nombre, et affichez s'il est pair ou impair.

#### 4. **\*\*Exercice 4 : Boucle For\*\***

Écrivez un programme qui affiche tous les nombres de 1 à 10 à l'aide d'une boucle `for`.

---

### ### Jour 2 : Structures de contrôle et boucles

**\*\*Objectif\*\*** : Manipuler les structures de contrôle (if, switch) et les boucles (for, while, do-while).

#### 1. **\*\*Exercice 5 : Nombre maximum\*\***

Demandez à l'utilisateur de saisir trois nombres et affichez le plus grand.

#### 2. **\*\*Exercice 6 : Table de multiplication\*\***

Demandez un nombre à l'utilisateur et affichez la table de multiplication correspondante (de 1 à 10).

#### 3. **\*\*Exercice 7 : Compteur de voyelles\*\***

Demandez à l'utilisateur de saisir une chaîne de caractères, puis affichez combien de voyelles elle contient.

#### 4. **\*\*Exercice 8 : Factorielle avec while\*\***

Écrivez un programme qui calcule la factorielle d'un nombre donné par l'utilisateur, en

utilisant une boucle `while`.

---

### ### Jour 3 : Fonctions et méthodes

**\*\*Objectif\*\*** : Apprendre à créer et utiliser des fonctions/méthodes.

#### 1. **\*\*Exercice 9 : Calcul de surface\*\***

Créez une méthode qui prend en paramètre la longueur et la largeur d'un rectangle et retourne sa surface.

#### 2. **\*\*Exercice 10 : Conversion de température\*\***

Créez une méthode qui convertit des degrés Celsius en Fahrenheit.

#### 3. **\*\*Exercice 11 : Somme de tableaux\*\***

Créez une méthode qui prend un tableau d'entiers en paramètre et retourne la somme de ses éléments.

#### 4. **\*\*Exercice 12 : Nombre premier\*\***

Créez une méthode qui prend un nombre en paramètre et retourne `true` s'il est premier, sinon `false`.

---

### ### Jour 4 : Tableaux et chaînes de caractères

**\*\*Objectif\*\*** : Manipuler les tableaux et les chaînes de caractères.

#### 1. **\*\*Exercice 13 : Tableau inversé\*\***

Créez un programme qui demande à l'utilisateur de saisir 5 nombres et les affiche dans l'ordre inverse.

#### 2. **\*\*Exercice 14 : Recherche dans un tableau\*\***

Demandez à l'utilisateur de saisir un tableau de 10 nombres et un autre nombre. Vérifiez si ce nombre fait partie du tableau.

#### 3. **\*\*Exercice 15 : Palindrome\*\***

Créez un programme qui vérifie si une chaîne de caractères donnée est un palindrome (se lit de la même manière dans les deux sens).

#### 4. **\*\*Exercice 16 : Tri par sélection\*\***

Implémentez l'algorithme du tri par sélection pour trier un tableau d'entiers.

---

### ### Jour 5 : Programmation orientée objet (POO) – Partie 1

**\*\*Objectif\*\*** : Introduction aux concepts de POO (classe, objet, constructeur, méthodes).

#### 1. **\*\*Exercice 17 : Classe Voiture\*\***

Créez une classe `Voiture` avec des attributs `marque`, `modèle` et `année`. Créez un constructeur et une méthode pour afficher les informations de la voiture.

#### 2. **\*\*Exercice 18 : Classe Compte Bancaire\*\***

Créez une classe `CompteBancaire` avec les attributs `solde` et `titulaire`. Ajoutez des méthodes pour `deposer` de l'argent et `retirer` de l'argent.

#### 3. **\*\*Exercice 19 : Classe Personne\*\***

Créez une classe `Personne` avec les attributs `nom`, `âge` et `adresse`. Ajoutez une méthode pour modifier l'adresse et afficher les informations.

#### 4. **\*\*Exercice 20 : Classe Livre\*\***

Créez une classe `Livre` avec les attributs `titre`, `auteur` et `année de publication`. Créez une méthode pour afficher les informations du livre.

---

### ### Jour 6 : Programmation orientée objet (POO) – Partie 2

**\*\*Objectif\*\*** : Approfondir les concepts de POO (héritage, polymorphisme, encapsulation).

#### 1. **\*\*Exercice 21 : Héritage\*\***

Créez une classe `Animal` avec un attribut `nom` et une méthode `manger()`. Créez deux sous-classes `Chien` et `Chat` qui héritent d'`Animal` et qui redéfinissent la méthode `manger()`.

#### 2. **\*\*Exercice 22 : Polymorphisme\*\***

Utilisez le polymorphisme pour créer un tableau d'objets `Animal` contenant à la fois des `Chien` et des `Chat`. Faites-les tous "manger" en utilisant une boucle.

#### 3. **\*\*Exercice 23 : Encapsulation\*\***

Créez une classe `CompteBancaire` avec des attributs privés `solde` et `titulaire`, et ajoutez des méthodes `getSolde()` et `setSolde()` pour y accéder.

#### 4. **\*\*Exercice 24 : Classe abstraite et interface\*\***

Créez une classe abstraite `Vehicule` avec une méthode abstraite `demarrer()`. Implémentez deux classes concrètes `Voiture` et `Moto` qui étendent `Vehicule`. Utilisez une interface `Reparable` avec une méthode `reparer()` que les deux classes implémenteront.

---

### ### Jour 7 : Gestion des exceptions et fichiers

**\*\*Objectif\*\*** : Apprendre à gérer les erreurs et à manipuler des fichiers.

#### 1. **\*\*Exercice 25 : Gestion des exceptions\*\***

Créez un programme qui demande à l'utilisateur d'entrer deux nombres et affiche le résultat de la division. Gérez les exceptions (comme la division par zéro) avec ``try-catch``.

#### 2. **\*\*Exercice 26 : Lecture d'un fichier texte\*\***

Créez un programme qui lit le contenu d'un fichier texte et l'affiche dans la console.

#### 3. **\*\*Exercice 27 : Écriture dans un fichier texte\*\***

Créez un programme qui demande à l'utilisateur de saisir une phrase, puis l'enregistre dans un fichier texte.

#### 4. **\*\*Exercice 28 : Sérialisation d'objets\*\***

Créez une classe ``Etudiant`` avec les attributs ``nom``, ``age`` et ``note``. Implémentez l'interface ``Serializable`` et écrivez un programme pour sérialiser et désérialiser un objet ``Etudiant`` dans un fichier.

---

Ces exercices vous permettront de développer vos compétences en Java de manière progressive, chaque jour abordant un aspect clé du langage