



Formation Java – Introduction à la Programmation Objet/Java

Code : DJ06

Durée : 3 jours

Classe : Présentiel / à distance

Public

- Toute personne souhaitant apprendre à programmer en Java

Prérequis

- Aucune connaissance particulière requise.

Objectifs

- Comprendre les concepts fondamentaux de la programmation.
- Savoir écrire, compiler et exécuter un programme Java.
- Apprendre les bases de la programmation orientée objet.
- Développer des compétences pratiques en manipulation de données et débogage.

Programme détaillé

1. Les fondements de la programmation

- Qu'est-ce qu'un programme ? Qu'est-ce qu'un langage ? Les différents paradigmes. Quel langage pour quelle application ?
- Les compilateurs. Les exécutables.
- Les responsabilités d'un programmeur.
- Qu'est-ce qu'un algorithme ?
- Les besoins auxquels répond un algorithme.
- Le concept de pseudo-langage.
- Travaux pratiques:
- Présentation de différents langages (Java, C#, Visual Basic, C, C++). Ecriture d'un premier algorithme en pseudo-langage.

2. Genèse d'un premier programme

- Ecriture d'un programme simple : syntaxe et instructions.
- Compilation et exécution du programme.
- Qu'est-ce qu'une librairie ? Son rôle, son usage.
- Travaux pratiques:
- Découverte de l'environnement de développement et d'exécution. Ecriture, compilation et exécution d'un premier programme.

3. Règles de programmation

- Convention de nommage.
- Convention syntaxique.
- Utilisation des commentaires. Pourquoi commenter les développements ?
- Améliorer la lisibilité des programmes : indentation du code, découpage du code...

4. Les variables

- Qu'est-ce qu'une variable ?
- Pourquoi typer une variable ?
- Les types primitifs : entiers, chaînes de caractères, nombres réels, autres.
- Déclaration, définition et initialisation d'une variable.
- Les constantes.
- Saisie, affichage, affectation, conversion de type.
- Organiser ses données sous forme de tableaux.





Formation Java – Introduction à la Programmation Objet/Java

Code : DJ06

Durée : 3 jours

Classe : Présentiel / à distance

- Les types évolués : enregistrement, matrice, arbre.
- Travaux pratiques
- Ecriture de plusieurs programmes simples manipulant les variables.
- 5. **Opérateurs et expressions**
- Les différents opérateurs (multiplicatif, additif, comparaison, égalité, logique, affectation).
- Combinaison d'opérateurs.
- Expression booléenne.
- Travaux pratiques
- Manipulation des opérateurs et des expressions booléennes.
- 6. **Les structures de contrôle**
- Les sélections alternatives (si, si-alors-sinon, sélection cas).
- Les blocs d'instructions (notion de Début... Fin).
- Les boucles itératives (tant que-répéter, répéter-jusqu'à, pour-de-à).
- Imbrication des instructions.
- Les commentaires.
- Travaux pratiques
- Utilisation des structures de contrôle pour implémenter un algorithme.
- 7. **Les procédures et les fonctions**
- Définitions : procédure, fonction.
- Pourquoi sont-elles incontournables en programmation (réutilisabilité, lisibilité...)
- Le passage de paramètres.
- Le code retour d'une fonction.
- Sensibilisation aux limites du passage de la valeur d'une variable.
- Notion de passage par adresse.
- Appel de fonctions.
- 8. **Introduction à la programmation Objet**
- Les concepts associés à la programmation Objet : classe, attribut, méthode, argument.
- La modélisation Objet à partir des exigences fonctionnelles.
- Introduction aux bonnes pratiques d'organisation de conception et d'organisation d'un programme.
- Travaux pratiques
- Illustration des concepts Objets.
- 9. **L'accès aux bases de données**
- Organisation et stockage des données.
- Les traitements de base (connexion, requêtes, récupération des données).
- Application cliente et serveur de données.
- Affichage et manipulation des données dans l'application cliente.
- Travaux pratiques
- Création d'un formulaire de recherche d'informations dans une base de données.
- 10. **Maintenance, débogage et test des programmes**
- Savoir lire et interpréter les différents messages d'erreurs.
- Utiliser un débogueur : exécuter un programme pas à pas, points d'arrêts, inspecter les variables pendant l'exécution.
- Prévoir les tests unitaires.
- Travaux pratiques
- Utilisation d'un débogueur pour contrôler l'exécution des programmes.

