

Java

Objectifs pédagogiques

- Utiliser l'IDE Eclipse pour vos projets Java
- Ecrire, compiler, exécuter et déboguer des programmes Java
- Appliquer les concepts de programmation orientée objet au langage Java
- Utiliser la bibliothèque de classes Java
- Gérer les erreurs et mettre en œuvre la journalisation de vos applications
- Réaliser des tests unitaires à l'aide de JUnit

Prérequis

- Une bonne connaissance d'un langage de programmation structuré est impérative ;
- Avoir développé et livré une application en autonomie ou en équipe ;
- Idéalement, mais pas obligatoire : connaître les grands principes liés à la programmation orientée objet.

Programme détaillé

Introduction

- Objectifs de la formation
- Historique de Java
- Caractéristiques de Java
- Cycle de construction Java
- Les fichiers
- Démonstration – Installation de l'environnement

Les bases du langage Java

- Les instructions
- Les commentaires
- Le point d'entrée du programme, sa mise en œuvre et ses arguments
- Les outils de développement Java
- Présentation d'Eclipse IDE
- Le développement avec Eclipse
- Démonstration – Utilisation d'Eclipse
- Types, variables et constantes
- Les entiers
- Les littéraux binaires
- Les réels
- Les littéraux numériques avec underscore
- Les booléens et les caractères
- Conversions de types simples
- Les tableaux
- Les tableaux à dimensions multiples
- Les chaînes de caractères
- L'inférence de types

- Les opérateurs unaires
- Les opérateurs arithmétiques
- Les opérateurs binaires
- Les opérateurs de comparaison
- Les opérateurs logiques
- L'opérateur ternaire
- Les opérateurs d'affectation
- Les structures de contrôle
- La structure if
- L'opérateur ternaire, l'opérateur d'alternative
- La structure switch – case
- L'expression switch
- Les structures itératives
- Les ruptures de séquences itératives
- La portée des variables
- Démonstration – Algorithmie en Java
- Les procédures et fonctions : méthodes
- Paramètres, surcharge et ellipse des méthodes
- Bonnes pratiques
- **TP à réaliser**

La programmation orientée objet

- Introduction
- Historique et concepts fondamentaux
- UML
- La classe Java
- Les packages, les systèmes de fichiers et les classes
- Les attributs et les méthodes
- Classe et instance
- Démonstration – Définition d'une classe
- Les modules
- La déclaration d'un module
- Le module path
- Démonstration – Les modules
- Le constructeur et le destructeur
- Initialiseur, surcharge et constructeur par défaut
- Le destructeur
- Les membres
- Les membres de classe
- L'encapsulation
- Démonstration – Les objets et l'encapsulation
- **TP à réaliser**
- Les collaborations entre classes et objets
- Les classes enveloppes (wrappers)
- Les énumérations
- La classe String, les opérations sur les chaînes et les Text Blocks
- L'héritage
- Le transtypage
- La réutilisation de code

- La visibilité des membres
- Les classes scellées
- Démonstration – L’héritage
- Le polymorphisme
- Les classes et méthodes abstraites
- Les classes finales et outils Eclipse
- Démonstration – Le polymorphisme
- La classe Object
- Les méthodes getClass, toString et equals
- D’autres classes spécifiques : les Records
- Les interfaces et l’héritage multiple
- Les interfaces
- Les interfaces vides
- L’interface Serializable
- L’interface Cloneable
- Les interfaces fonctionnelles
- Les interfaces : méthodes par défaut
- Les expressions lambdas
- Les interfaces fonctionnelles de l’API Java
- Les références de méthodes
- Démonstration – Les interfaces
- **TP à réaliser**
- Gestion des erreurs : les exceptions
- Les exceptions : origine de l’erreur
- Les exceptions : traitement de l’erreur
- Les exceptions : différents scénarios
- Les exceptions : chaînage
- Les différents types d’exception
- Les exceptions utilisateur
- Les bonnes pratiques sur les exceptions
- Les exceptions : finally
- La structure try-with-resource
- Amélioration de la gestion des exceptions
- Démonstration – Les exceptions
- **TP à réaliser**

La bibliothèque de classes Java

- Introduction
- Les collections
- ArrayList, Iterator et HashMap
- Les génériques
- Démonstration – Les collections
- Les streams
- Caractéristiques et manipulation des streams
- L’obtention d’un stream
- Opérations intermédiaires sur les streams
- Autres opérations terminales sur les streams
- Démonstration – Les streams
- Les entrées/sorties et lecture de données au clavier

- La classe Scanner
- La gestion des dates et du temps
- La création d'une date ou d'une heure, traitements et calculs
- Manipulation et formatage des dates et des heures
- **TP à réaliser**

La journalisation en Java

- Introduction
- Les bibliothèques de journalisation Java
- Log4j
- La configuration de Log4j
- Les Loggers
- Les Appenders
- La configuration du PatternLayout
- La configuration des Loggers
- Démonstration – Log4j
- **TP à réaliser**

Les tests unitaires avec JUnit

- Introduction
- Test et test unitaire
- JUnit : concepts, évolutions et installation
- Démonstration – Mise en oeuvre de JUnit dans Eclipse
- Structuration du code des tests avec JUnit
- Création d'un cas de test avec un IDE
- Mise en oeuvre des tests dans le code Java
- Factorisation et fixture
- Suite de tests
- Traitement des exceptions
- Runners
- Démonstration – JUnit
- **TP à réaliser**