

Java : Niveau intermédiaire

Approfondir les API indispensables du développement Java

Ces packages constituent pour nous la boîte à outils indispensable du développeur Java. C'est aussi l'occasion de revoir en détail les concepts avancés du langage.

Cette formation constitue le complément idéal pour passer de la connaissance de la syntaxe de Java et de ses mécanismes de base à son utilisation effective dans les développements.

Notre parti pris est cependant de rester dans le cadre des packages java (et non javax). Ainsi des concepts importants comme la gestion de fichiers xml ou les interfaces riches avec swing sont abordés dans des formations spécifiques.

Détails

- Code : JA-PGI
- Durée : 3 jours (21 heures)

Public

- Architectes
- Chefs de projets
- Concepteurs
- Consultants
- Développeurs

Pré-requis

- Avoir suivi la formation Java les bases ou posséder les connaissances équivalentes

Objectifs

- Consolider les connaissances acquises lors du cours Java Base
- Passer en revue les packages essentiels du développeur Java

Programme

Syntaxe avancée

- Les classes internes et anonymes
- Les constructeurs
- Les blocs d'initialisation
- Les types génériques
- La syntaxe des lambdas (Java 8)
- La syntaxe des références de méthode (Java 8)

Le package java.lang

- La classe Object
- L'interface Comparable : égalité et comparaison
- L'interface Clonable : copie d'objet
- Les wrappers et l'autoboxing : conversions de type
- Les chaînes : String, CharSequence, Appendable, StringBuffer, StringBuilder, ...
- Les extensions syntaxiques : Iterable, AutoCloseable
- Les énumérations : classe Enum
- Les opérations mathématiques : Math et StrictMath
- Les annotations standards (@Deprecated, @Override, ...)
- Les exceptions : Throwable, Exception, Error, RuntimeException, ...
- Les classes utilitaires : System, Runtime, Process et ProcessBuilder
- Les bases du parallélisme : Runnable, Thread, ...
- L'introspection : Class, Package, ClassLoader ...
- Les autres éléments du package java.lang

Le package java.math

- Les nombres réels et les erreurs d'arrondis
- Les nombres étendus : BigInteger, BigDecimal
- La gestion des arrondis : MathContext et RoundingMode

Le package java.util

- Les collections : Collection, List, Queue, Set, Map, ...
- Itérer sur les collections : Enumeration et Iteration
- Les classes d'implémentations de collections
- Les classes utilitaires : Collections et Arrays
- La gestion du temps : Date, Calendar, ...
- La représentation de la monnaie : Currency
- Le paramétrage : Properties
- L'internationalisation : Locale, ResourceBundle, Formatter, ...
- Les classes utilitaires : Scanner, StringTokenizer, Random, ...
- Les autres éléments du package java.util : Observer, Observable, ServiceLoader, ...

Le package java.text

- La comparaison des chaînes de caractères : Collator, RuleBasedCollator
- Le formatage textuel : Format, MessageFormat, NumberFormat, DateFormat, ...

Le package java.io

- La gestion de fichiers : File, FileFilter, FilenameFilter, ...

- La gestion des flux binaires : InputStream, OutputStream, ...
- La gestion des flux textes : Reader, Writer, ...
- La gestion des flux d'objets : Serializable, Externalizable, ...
- Les classes utilitaires : Console, StreamTokenizer, RandomAccessFile

Le package java.nio

- La « nouvelle » gestion de fichiers : FileStore, FileSystem, Path, FileSystems, Files, Paths, PathMatcher, WatchService, ...
- Les transferts de données : Buffer, Channel, Channels, ...

Le package java.net

- Les accès réseau : URL, URLConnection, URLEncoder, URLDecoder
- Les protocoles TCP et UDP : InetAddress, NetworkInterface, Socket, ServerSocket, ...

Le package java.sql

- Le framework JDBC : DriverManager, Connection, Statement, PreparedStatement, ResultSet, ...

Le package java.util.concurrent

- Le parallélisme avancé (Futur, Executor, ExecutorService, Executors, ...)
- Les collections synchronisées : BlockingQueue, ConcurrentMap, ...

- Le package java.util.concurrent.atomic : les conteneurs thread-safe
- Le package java.util.concurrent.locks : la gestion explicite des locks

Le package java.time (Java 8)

- Les nouvelles classes temporelles : Instant, Duration, LocalDate, LocalTime, Period, YearMonth, Temporal, ...

Les packages java.util.fonction et java.util.stream (Java 8)

- Les FonctionalInterface : Consumer, Predicate, Fonction, Supplier, ...
- Les interfaces de streams : BaseStream, Stream, Collector, ...
- La construction des streams : stream(), paralleleStream(), iterate(), generate(), ...
- Les fonctions d'aggregats : forEach, filter, sorted, map, collect, ...

Aperçu de quelques autres packages

- Le package java.util.logging : les traces
- Le package java.util.prefs : la gestion des préférences utilisateurs
- Le package java.util.jar : la gestion des jar
- Le package java.util.zip : la gestion des zip
- Le package java.util.regex : les expressions régulières
- Le package java.awt : les interfaces graphiques natives
- Le package javax.swing : les interfaces graphiques riches

Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages