Java SE programming

CODICE	**DT0068**
DURATA	**5 gg**
PREZZO	**1.790,00 €**
EXAM	

DESCRIZIONE

Java è sia un linguaggio di programmazione orientato agli oggetti che una piattaforma software sviluppata e pensata per essere indipendente nell'esecuzione dalla piattaforma Hardware su cui opera. I vantaggi sono la velocità di esecuzione ma soprattutto il fatto che l'esecuzione sia sempre la stessa indipendentemente dal device, dal software o dalla piattaforma utilizzata.

TARGET

Il corso è rivolto a coloro che vogliono acquisire le conoscenze fondamentali relative alla programmazione Java anche senza avere delle basi relative alla programmazione Object Oriented.

- Software Developer
- Project Manager
- System Administrator
- Java EE Application Server Administrator
- Team Leader
- Technical Administrator
- Technical Consultant
- Web Administrator

PREREQUISTI

Opzionalmente, potrebbe essere utile avere i rudimenti della programmazione ad oggetti

CONTENUTI

Introduzione a Java

- Descrivere la tecnologia Java e lo sviluppo Java
- Identificare le funzionalità chiave del linguaggio Java

I blocchi di costruzione dei programmi Java

- Scrivere il metodo *main()*
- Utilizzare le variabili e gli operatori
- Capire la struttura delle classi Java: campi, metodi e costruttori
- Scrivere i commenti e la documentazione *javadoc*
- Capire le *package declaration* e le istruzioni *import*
- Dichiarare, istanziare e inizializzare gli oggetti
- Lavorare con i riferimenti a oggetti
- Utilizzare gli array
- Capire la differenza fra tipi primitivi e tipi riferimento
- Promozione e *casting* (conversione del tipo) delle variabili e degli oggetti
- Come si distruggono gli oggetti, Garbage Collection automatico, il metodo *finalize()*

Operatori e costrutti di controllo del flusso

- Conoscere gli operatori e le istruzioni Java *if-then, if-then-else, switch, while, do-while, for, for-each.*
- Cicli nidificati, etichette, istruzioni *break* e *continue.*

Principali API Java

- Il pacchetto *java.lang*, la classe *Object.*
- Le classi *Wrapper* dei tipi semplici, *autoboxing* e *unboxing.*
- La classe *String.*
- Java Array e la classe *ArrayList*
- Lavorare con le date e il tempo.

Lavorare con Java Arrays

- Dichiarare, istanziare, inizializzare e utilizzare una matrice unidimensionale
- Dichiarare, creare un’istanza, inizializzare e utilizzare un array bidimensionale

Programmazione orientata agli oggetti – Incapsulamento e metodi

- Progettare i metodi: argomenti e valori restituiti
- Metodi e variabili *static*
- Passaggio degli argomenti ai metodi
- *Overloading* dei metodi
- Lavorare con i *varargs*
- Costruttori parametrizzati
- Incapsulamento dei dati, metodi *get()* e *set()*, modificatori di accesso *public, protected, default, private*

Programmazione orientata agli oggetti – Ereditarietà e polimorfismo

- Ereditarietà delle classi
- *Overriding* dei metodi della superclasse
- Polimorfismo
- Creare ed estendere una classe astratta

Interfacce

- Il paradigma *Design By Contract*
- Interfaccce: metodi di default
- Interfacce: metodi *static*
- Ereditarietà e polimorfismo con le interfacce

Eccezioni

- Principi della gestione delle eccezioni, blocchi *try-catch-finally* e *try-with-resources*
- Gestire le eccezioni multiple.
- Propagazione delle eccezioni, clausola *throws* ed istruzione *throw*
- Capire i tipi comuni di eccezioni
- Creare le eccezioni personalizzate.

Generics e Collections Framework

- Classi generiche e metodi generici
- Le interfacce e le classi principali del Java Collections Framework

Elementi della programmazione funzionale

- Interfacce funzionali ed espressioni lambda
- Collezioni, flussi (*stream*) e filtri
- Le interfacce funzionali della libreria standard

Elementi della programmazione modulare

- Il sistema modulare
- I JAR
- Dichiarazione di moduli
- JDK modulare