

Java SE 8 and Java EE 7

جاوا مقدماتی و پیشرفته

خلاصه :

این دوره در قالب یک پکیج فشرده، آموزش مقدماتی زبان برنامه نویسی جاوا را به همراه برنامه نویسی سیستمهای کاربردی-تجاری مبتنی بر وب با معماری چند لایه (Web Application) و با بکارگیری زبان جاوا را شامل می شود. در این دوره ابتدا طی ۲۵ ساعت اصول برنامه نویسی به زبان جاوا تدریس خواهد شد سپس ۳۵ ساعت دوم به آموزش جاوای پیشرفته سپری خواهد شد.

در این دوره ابتدا نسخه استاندارد جاوا (Java Standard Edition) تدریس خواهد شد سپس به نسخه تجاری جاوا (Java Enterprise Edition) متمرکز خواهد شد. پیش نیاز فراگیری نسخه تجاری جاوا (Java Enterprise Edition) یادگیری و مسلط بودن بر نسخه استاندارد جاوا (Java Standard Edition) می باشد که در این دوره بصورت فشرده تدریس می گردد. برنامه نویسی سیستمهای کاربردی-تجاری مبتنی بر وب با معماری چند لایه (Web Application) و با بکارگیری زبان جاوا بعنوان بخشی از نسخه تجاری جاوا (Java Enterprise Edition) در نظر گرفته می شود. در این دوره مباحثی همچون معماری چند لایه، الگوهای طراحی در نسخه تجاری جاوا (Java Enterprise Edition)، شیوه پیاده سازی مدل MVC به زبان جاوا، آشنایی با وب سرویس ها (Web Services) و نحوه پیاده سازی آنها در زبان جاوا و آشنایی با چارچوبهایی همچون JSF و EJB و Hibernate ارائه می گردد.

مدت دوره:

۶۰ ساعت

پیش نیاز:

حداقل یک سال تجربه برنامه نویسی ترجیحاً به زبان C# یا C/C++، آشنایی با مفاهیم بانک اطلاعات رابطه ای، آشنایی با مفاهیم پایه وب و HTML

مخاطب:

افرادی که قصد دارند در بازه زمانی فشرده زبان برنامه نویسی جاوا را بشناسند و به شناخت جاوای پیشرفته برای برنامه نویسی سیستمهای مبتنی بر وب بپردازند- کارشناسان نرم افزار

Java SE 8 Objectives

- Creating high-performing multi-threaded applications
- Creating Java technology applications that leverage the object-oriented features of the Java language, such as encapsulation, inheritance, and polymorphism
- Implementing input/output (I/O) functionality to read from and write to data and text files and understand advanced I/O streams
- Executing a Java technology application from the command line
- Manipulating files, directories and file systems using the JDK NIO.2 specification
- Creating applications that use the Java Collections framework
- Performing multiple operations on database tables, including creating, reading, updating and deleting using both JDBC and JPA technology
- Searching and filter collections using Lambda Expressions
- Implementing error-handling techniques using exception handling

Java EE 7 Objectives

- Defining how the Java language achieves platform independence
- Differentiating between the Java ME, Java SE, and Java EE Platforms
- Evaluating Java libraries, middle-ware, and database options
- Defining how the Java language continues to evolve
- Use Java Persistence and Java Transaction APIs.
- Create a flexible component model using EJB and CDI technology.

Java SE 8 Outline

- **Creating simple Java classes**
 - Creating primitive variables
 - Using operators
 - Creating and manipulate strings
 - Using if-else and switch statements
 - Iterating with loops: while,do-while,for,enhanced for
 - Creating arrays
 - Using Java fields, constructors, and methods
- **Encapsulation and Sub classing**
 - Using encapsulation in Java class design
 - Modeling business problems using Java classes
 - Making classes immutable
 - Creating and use Java subclasses

- Overloading methods
- **Overriding Methods, Polymorphism, and Static Classes**
 - Using access levels: private, protected, default, and public.
 - Overriding methods
 - Using the instanceof operator to compare object types
 - Using upward and downward casts
 - Modeling business problems by using the static keyword
 - Implementing the singleton design pattern
- **Abstract and Nested Classes**
 - Designing general-purpose base classes by using abstract classes
 - Constructing abstract Java classes and subclasses
 - Applying final keyword in Java
 - Distinguish between top-level and nested classes
- **Interfaces and Lambda Expressions**
 - Defining a Java interface
 - Choosing between interface inheritance and class inheritance
 - Extending an interface
 - Defaulting methods
 - Anonymous inner classes
 - Defining a Lambda Expression
- **Collections and Generics**
 - Creating a custom generic class
 - Using the type inference diamond to create an object
 - Creating a collection by using generics
 - Ordering collections
- **Exceptions and Assertions**
 - Defining the purpose of Java exceptions
 - Using the try and throw statements
 - Using the catch, multi-catch, and finally clauses
 - Auto close resources with a try-with-resources statement
 - Recognizing common exception classes and categories
 - Creating custom exceptions
 - Testing invariants by using assertions
- **Java Date/Time API**
 - Creating and manage date-based events
 - Creating and manage time-based events
 - Combining date and time into a single object
 - Working with dates and times across time zones
 - Managing changes resulting from daylight savings
 - Defining and create timestamps, periods and durations
 - Applying formatting to local and zoned dates and times
- **I/O Fundamentals**
 - Describing the basics of input and output in Java
 - Read and write data from the console
 - Using streams to read and write files
 - Writing and read objects using serialization
- **File I/O (NIO.2)**

- Using the Path interface to operate on file and directory paths
- Using the Files class to check, delete, copy, or move a file or directory
- Using Stream API with NIO2
- **Concurrency**
 - Describing operating system task scheduling
 - Creating worker threads using Runnable and Callable
 - Using an ExecutorService to concurrently execute tasks
 - Identifying potential threading problems
 - Using synchronized and concurrent atomic to manage atomicity
 - Using monitor locks to control the order of thread execution
 - Using the java.util.concurrent collections
- **Database Applications with JDBC**
 - Defining the layout of the JDBC API
 - Connecting to a database by using a JDBC driver
 - Submitting queries and get results from the database
 - Specifying JDBC driver information externally
 - Performing CRUD operations using the JDBC API
- **Localization**
 - Describing the advantages of localizing an application
 - Defining what a locale represents
 - Read and set the locale by using the Locale object
 - Building a resource bundle for each locale
 - Calling a resource bundle from an application
 - Changing the locale for a resource bundle

Developing Applications for the Java EE 7 Platform Outline

Introduction to Java EE

- Standards, containers, APIs, and services
- Application component functionalities mapped to tiers and containers
- Interconnect Application Components with CDI Annotations and JNDI
- Web container technologies, Business logic implementation technologies, and web service technologies
- Packaging and deployment
- Enterprise JavaBeans, managed beans, and CDI beans
- Understanding lifecycle and memory scopes
- Linking components together with annotations, injections, and JNDI

Managing Persistence by Using JPA Entities

- Create JPA entities with Object-Relational Mappings (ORM)
- Use Entity Manager to perform database operations with JPA entities
- Handle entity data with conversions, validations, and key generation
- Describe persistence management and locking mechanisms
- Create and execute JPQL statements

Implementing Business Logic by Using EJBs

- Create Session EJB components
- Create EJB business methods
- Manage EJB life cycle with container callbacks
- Use asynchronous EJB operations
- Control transactions
- Create EJB timers
- Create and apply interceptors

Using Java Message Service API

- Describe Java Message Service (JMS) API messaging models
- Implement Java SE and Java EE message producers and consumers
- Use durable and shared topic consumer subscriptions
- Create message-driven beans
- Use transactions with JMS

Implementing SOAP Services by Using JAX-WS

- Describe a SOAP Web Service structure
- Create SOAP Web Services using JAX-WS API
- Create SOAP Web Service clients

Creating Java Web Applications by Using Servlets

- Describe HTTP basics
- Create Java servlet classes and map them to URLs
- Handle HTTP headers, parameters, cookies
- Use servlets to handle different content types
- Manage servlet life cycle with container callback methods
- Use CDI Managed Beans
- Use Interceptors and Filters
- Implement asynchronous servlets and use NIO API

Creating Java Web Applications by Using JSPs

- Describe JSP life cycle
- Describe JSP syntax
- Use Expression Language (EL)
- Use CDI Beans
- Use Tag Libraries
- Handle errors

Implementing REST Services using JAX-RS API

- Understand REST service conventions
- Create REST services using JAX-RS API
- Consume REST service within the client tier

Develop Web Applications Using JavaServer Faces

- Describe JSF lifecycle and architecture, and understand JSF syntax
- Use JSF Tag Libraries
- Apply Validators and Converters to UIComponents
- Use UI templates
- Define navigation, and handle localization
- Produce messages
- Use Expression Language (EL), and CDI Beans

- Add AJAX support

Introduction to Java EE Open Source Frameworks

- Hibernate Persistence Framework
- Spring Framework
- Maven Framework
- Quartz Scheduler