Java SE 21: Programming II Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Oracle Resmi Müfredatına dayanan bu {*eğitim}, Java dilini öğrenen geliştiriciler için ikinci seviye bir kurstur. Önceki "Java SE: Programming I" eğitiminin üzerine inşa edilen bu kursta, nesne yönelimli programlama (OOP), modüler yapı, güvenli kodlama, veritabanı bağlantısı, Lambda kullanımı ve çok iş parçacıklı (multithreaded) uygulama geliştirme gibi konulara odaklanılır.

Eğitim boyunca Java'nın güncel versiyonu olan **Java SE 21** kullanılır.

Prerequisites

Java SE 21: Programming I eğitiminin tamamlanmış olması veya eşdeğer bilgi birikimi

Who Should Attend

- Java geliştiricileri
- Veri Bilimcil
  - Java geliştiricileri
  - Veri Bilimciler (Data Scientist)
  - Java tabanlı sistemlerde çalışan uygulama geliştiricileri
  - Yeni nesil Java teknolojileriyle uygulama geliştirmek isteyen tüm yazılım uzmanları
- er (Data Scientist)

- Java tabanlı sistemlerde çalışan uygulama geliştiricileri
- Yeni nesil Java teknolojileriyle uygulama geliştirmek isteyen tüm yazılım uzmanları

Java SE 21: Programming II Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

Bu {*eğitim} sonunda aşağıdaki konularda bilgi ve beceri kazanmış olacaksınız:

- Kapsülleme, kalıtım ve çok biçimlilik (polymorphism) gibi nesne yönelimli özellikleri içeren uygulamalar geliştirebilir,
- Komut satırından Java uygulamaları çalıştırabilir,
- Java Collections Framework yapısını etkili bir şekilde kullanabilir,
- Lambda ifadeleriyle koleksiyonları filtreleyebilir ve işleyebilir,
- Exception handling ile hataları etkili bir şekilde yönetebilir,
- Dosya ve veri işlemleri için G/Ç (I/O) fonksiyonlarını uygulayabilir,
- JDK NIO.2 API'sini kullanarak dosya sistemlerini yönetebilir,
- JDBC ve JPA teknolojilerini kullanarak veritabanı işlemleri gerçekleştirebilir (CRUD: Create, Read, Update, Delete),
- Lambda ile paralel işlem ve concurrency mekanizmalarını uygulayabilir,
- Modüler programlama tekniklerini kullanarak sürdürülebilir yazılım tasarlayabilir,
- Performanslı ve çok iş parçacıklı uygulamalar geliştirebilir.

Outline

1. Fonksiyonel Programlama

- **Lambda Expressions** ve **Functional Interface** kavramları
- Java'nın hazır fonksiyonel arayüzleri (`Predicate`, `Consumer`, `Function` vs.)
- Koleksiyonlar üzerinde `stream()`, `filter()`, `map()` işlemleri
- Stream API ile temiz, okunabilir ve kısa kod yazımı

2. Modüler Programlama

- Java 9 ile gelen **modüler sistemin (JPMS)** temelleri
- `module-info.java` dosyasının kullanımı
- Uygulamalarda modül ve servis tanımı
- Legacy projelerin modüler yapıya taşınması (Migration)

Java SE 21: Programming II Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

3. Stream ve Paralel Stream İşlemleri

- **Parallel Stream** kavramı
- Performans artışı için stream'leri paralel hale getirme
- Collectors API'si ile terminal işlemleri
- Özel (custom) stream yapılandırmaları

4. Concurrency (Eşzamanlılık)

- Çok iş parçacıklı uygulamalar (Multithreading)
- ExecutorService, Callable, Runnable kullanımı
- Thread senkronizasyonu ve yarış durumu (race condition) yönetimi

5. Java API ve Güvenli Kodlama

- Giriş/Çıkış (I/O) temelleri: InputStream, OutputStream, Reader, Writer
- **NIO.2** ile modern dosya yönetimi (Path, Files, DirectoryStream)
- Güvenli kod yazım teknikleri ve açıkların önlenmesi

6. Veritabanı Uygulamaları (JDBC & JPA)

- JDBC API ile temel bağlantı ve sorgu işlemleri
- PreparedStatement ve SQL Injection koruması
- ORM mantığı ve JPA kullanımı
- CRUD operasyonları uygulamalı örneklerle

7. Localization (Yerelleştirme)

- Locale, ResourceBundle, DateFormat, NumberFormat kullanımı
- Çok dilli uygulama desteği sağlama

8. Koleksiyonlar ve Generics

- Java Collections Framework: List, Set, Map, Queue
- Iterator ve Iterable arayüzleri
- **Generics** ile tür güvenliği sağlama
- Wrapper sınıflar (Integer, Double, Boolean vs.)

9. Exception Handling ve Assertions

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Checked ve unchecked exception ayrımı
- Özel exception sınıfları oluşturma
- try, catch, finally, throw, throws kullanımı
- assert ifadesi ile koşul kontrolü

1. Fonksiyonel Programlama

- **Lambda Expressions** ve **Functional Interface** kavramları
- Java'nın hazır fonksiyonel arayüzleri (Predicate, Consumer, Function vs.)
- Koleksiyonlar üzerinde stream(), filter(), map() işlemleri
- Stream API ile temiz, okunabilir ve kısa kod yazımı

2. Modüler Programlama

- Java 9 ile gelen **modüler sistemin (JPMS)** temelleri
- module-info.java dosyasının kullanımı
- Uygulamalarda modül ve servis tanımı
- Legacy projelerin modüler yapıya taşınması (Migration)

3. Stream ve Paralel Stream İşlemleri

- **Parallel Stream** kavramı
- Performans artışı için stream'leri paralel hale getirme
- Collectors API'si ile terminal işlemleri
- Özel (custom) stream yapılandırmaları

4. Concurrency (Eşzamanlılık)

- Çok iş parçacıklı uygulamalar (Multithreading)
- ExecutorService, Callable, Runnable kullanımı
- Thread senkronizasyonu ve yarış durumu (race condition) yönetimi

5. Java API ve Güvenli Kodlama

- Giriş/Çıkış (I/O) temelleri: InputStream, OutputStream, Reader, Writer
- **NIO.2** ile modern dosya yönetimi (Path, Files, DirectoryStream)
- Güvenli kod yazım teknikleri ve açıkların önlenmesi

Java SE 21: Programming II Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

6. Veritabanı Uygulamaları (JDBC & JPA)

- JDBC API ile temel bağlantı ve sorgu işlemleri
- PreparedStatement ve SQL Injection koruması
- ORM mantığı ve JPA kullanımı
- CRUD operasyonları uygulamalı örneklerle

7. Localization (Yerelleştirme)

- Locale, ResourceBundle, DateFormat, NumberFormat kullanımı
- Çok dilli uygulama desteği sağlama

8. Koleksiyonlar ve Generics

- Java Collections Framework: List, Set, Map, Queue
- Iterator ve Iterable arayüzleri
- **Generics** ile tür güvenliği sağlama
- Wrapper sınıflar (Integer, Double, Boolean vs.)

9. Exception Handling ve Assertions

- Checked ve unchecked exception ayrımı
- Özel exception sınıfları oluşturma
- try, catch, finally, throw, throws kullanımı
- assert ifadesi ile koşul kontrolü