Java SE 25: Programming I Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Java SE 25: Programming I, Java programlamaya başlangıç seviyesinde yaklaşan ve katılımcılara Java'nın temel kavramlarını, dil yapılarını ve veri tiplerini öğretmeyi amaçlayan kapsamlı bir eğitimdir. Program kapsamında exception handling, lambda expressions ve modular programming gibi önemli Java konuları ele alınırken, uygulamalar güncel JDK sürümleri temel alınarak geliştirilir.

Her bölümde yer alan uygulamalı laboratuvar çalışmaları sayesinde katılımcılar öğrendikleri kavramları doğrudan kod üzerinde deneyerek Java programlama becerilerini geliştirme fırsatı bulur.

Prerequisites

Katılımcıların daha önce bir programlama dili üzerine eğitim almış olmaları veya Java programlama konusunda temel düzeyde bilgi sahibi olmaları önerilir.

Who Should Attend

Bu eğitim aşağıdaki profesyonellere yöneliktir:

- Yazılım Geliştiriciler
- Uygulama Uzmanları
- Sistem Entegrasyon Uzmanları
- Sistem Yöneticileri

What You Will Learn

Eğitimi tamamlayan katılımcılar:

- Programlama hedeflerini gerçekleştirmek için variables, arrays, conditional statements ve loop yapılarını kullanan Java kodları yazabilir.
- Modular programming prensiplerini tanımlayabilir ve ayırt edebilir.
- Static field ve method'ları oluşturabilir ve bunlara erişebilir.
- Access modifier'lar ve overloaded constructor'lar kullanarak class encapsulation uygulayabilir.
- Uygun Java operator'larını kullanarak numeric, text ve string verileri üzerinde işlem yapabilir.
- Java compiler ve runtime araçlarının doğru çalışmasını sağlayacak environment variable ayarlarını gerçekleştirebilir.
- Basit Java class'ları oluşturabilir ve object reference'lar aracılığıyla class içerisindeki field ve method'lara erişebilir.
- Bir Java interface'i implement ederek polymorphism kullanımını gösterebilir.
- Java uygulamalarındaki checked exception'ları yönetebilir.
- Bir method'a argüman olarak predicate lambda expression aktarabilir.
- Uygulama gereksinimlerini karşılayan temel bir class hierarchy tasarlayabilir ve uygulayabilir.

Java SE 25: Programming I Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Outline

Eğitime Giriş

- Eğitimin genel tanıtımı
- Eğitim hedefleri
- Hedef kitle
- Eğitim yapısı

Java'ya Giriş

- Java nedir?
- Java'nın çalışma yapısı
- Object-oriented programming prensipleri
- Classes ve objects
- Inheritance
- Java API'leri
- Java keywords, reserved words ve special identifiers
- Java naming conventions
- Java'nın temel syntax kuralları
- Java class tanımlama
- Farklı package'lar içerisindeki class'lara erişim
- Access modifier'larla encapsulation uygulama
- Main method içeren bir Java programı oluşturma
- Java programlarını derleme ve çalıştırma
- Comments ve documentation
- Javadoc içerisinde code snippets kullanımı
- External snippets

Primitive Types, Operators ve Flow Control Statements

- Java primitive types
- Primitive variable tanımlama ve başlatma
- Primitive declarations ve initializations için geçerli kısıtlamalar
- Java operators
- Assignment ve arithmetic operators
- Arithmetic operations ve type casting
- İleri matematiksel işlemler
- Binary number representation
- Bitwise logical operators
- Equality, relational ve conditional operators
- Short-circuit evaluation
- If/else ile program akışını kontrol etme
- Ternary operator
- Switch yapısıyla flow control
- Değer üreten switch expressions
- JShell (REPL) kullanımı
- JShell komutları
- Code snippets

Text, Date, Time ve Numeric Objects

- String initialization ve temel String işlemleri
- String indexing
- Mutable text objects
- Text blocks
- Primitive types için wrapper classes
- BigDecimal ile sayıların temsil edilmesi
- Method chaining
- Local Date and Time API
- Instants, durations ve periods
- Zoned date and time
- Dil ve ülke bilgilerinin temsil edilmesi
- Numeric, date ve time değerlerinin formatlanması ve parse edilmesi
- Localizable resources
- Message pattern'larının formatlanması

Classes ve Objects

- UML'e giriş
- Classes, interactions ve activities modelleme
- Class tasarımı
- Object oluşturma
- Instance variables ve methods tanımlama
- Object oluşturma ve erişim
- Local variables ve recursive object references
- Local variable type inference
- Constants tanımlama
- Static context ve erişim
- Static ve final kullanımının birleştirilmesi
- Diğer static context kullanım alanları
- IntelliJ IDE'ye giriş

Gelişmiş Class Tasarımı

- Method overloading
- Variable number of arguments
- Constructor tanımlama ve yeniden kullanma
- Access modifier'ların genel değerlendirmesi
- Encapsulation ve immutability tanımlama
- Constants ve immutability
- Enumerations ve complex enumerations
- Java memory allocation ve cleanup
- Parameter passing

Inheritance Uygulama ve Records Kullanımı

- Class'ları extend etme
- Object class
- Inheritance aracılığıyla parent class kodunun yeniden kullanılması
- Class instance oluşturma ve object'lere erişim
- Reference type casting kuralları
- Casting öncesinde object type doğrulama
- instanceof için pattern matching

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Mevcut veya parent object içerisindeki koda referans verme
- Subclass constructor'larının tanımlanması
- Flexible constructor bodies
- Class ve object initialization
- Method overriding ve polymorphism
- Overridden method içerisinde parent class logic'ini yeniden kullanma
- Abstract, final ve sealed classes ile methods tanımlama
- Object class operasyonlarını override etme: `toString`, `equals`, `hashCode`
- String object'lerini karşılaştırma
- Java records ve custom record constructors
- Record patterns
- Switch için pattern matching
- Factory methods

Interfaces ve Generics

- Java interfaces
- Multiple inheritance problemi
- Interface implementasyonu
- Interface'lerde default, private ve static methods
- Interface hierarchy
- Default method inheritance
- Bir type olarak interface kullanımı
- Functional interfaces
- Generics ve kullanım alanları
- Java interface örnekleri: Comparable, Comparator ve Cloneable
- Composition pattern

Arrays ve Loops

- Arrays: declaration, creation ve initialization
- Multidimensional arrays
- Array içeriğinin kopyalanması
- Arrays class
- Loop'lar ve array'lerin loop kullanılarak işlenmesi
- Complex for loops
- Nested loops
- Break ve continue kullanımı

Collections

- Java Collection API'ye giriş
- Collection API interfaces ve implementation classes
- List, Set ve Deque object'lerinin oluşturulması ve yönetilmesi
- HashMap object'lerinin oluşturulması ve yönetilmesi
- Collections üzerinde iteration
- Sequenced collections
- Diğer collection davranışları
- `java.util.Collections` class kullanımı
- Collections'a concurrent erişim
- Collection corruption'ın önlenmesi
- Legacy collection classes

Java SE 25: Programming I Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Nested Classes ve Lambda Expressions

- Nested class türleri
- Static nested classes
- Member inner classes
- Local inner classes
- Anonymous inner classes
- Anonymous inner classes ve functional interfaces
- Lambda expressions kavramı
- Lambda expression parameters ve body tanımlama
- Method references kullanımı
- Functional interface'lerde default ve static methods
- Comparator ve Predicate interface'lerinin default ve static method'larını kullanma

Java Streams API

- Stream'lerin temel özellikleri
- Stream API ile stream oluşturma
- Stream pipeline processing operations
- Functional interfaces kullanımı
- Functional interface'lerin primitive ve bi-argument varyantları
- Stream pipeline elemanlarında action, filtering ve mapping işlemleri
- `flatMap` kullanarak stream'leri birleştirme
- Diğer intermediate stream operations
- Custom intermediate stream operations: gatherers
- Short-circuit terminal operations
- `count`, `min`, `max`, `sum` ve `average` ile stream işleme
- `reduce` kullanarak stream verilerini aggregate etme
- Collect operasyonunun genel çalışma mantığı
- Temel collectors kullanımı
- Stream içeriğini dönüştürme, gruplama ve partition işlemleri
- Group ve partition'lar üzerinde mapping ve filtering
- Gatherers
- Parallel stream processing ve kullanım ilkeleri
- Parallel stream processing kısıtlamaları
- Spliterator ve özellikleri

Exception Handling, Logging ve Debugging

- Java Logging API kullanımı
- Logging method kategorileri
- Guarded logging
- Log yazımının yönetilmesi ve yapılandırılması
- Java exceptions
- Custom exception oluşturma
- Exception fırlatma ve yakalama
- Exceptions ve program execution flow
- Helpful NullPointerExceptions
- Unchecked ve checked exception fırlatma örnekleri
- Exception handling
- Resource auto-closure ve suppressed exceptions
- Exception cause yönetimi

Java SE 25: Programming I Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Java debugger ve debugger işlemleri
- Debug modunda program verilerinin değiştirilmesi
- Assertions kullanarak program logic doğrulama
- Exception bulunan ve bulunmayan durumlarda program akışı

Sınav ve Değerlendirme

Bu eğitim, ayrıca sunulan **Java SE 25 Developer Professional** sınavına hazırlanmak isteyen katılımcılar için gerekli bilgi ve becerilerin geliştirilmesine yardımcı olur.