



# **JAVA SE 17 PROGRAMMING EĞİTİMİ 5 GÜN**



**Digital Vizyon**  
Akademi

[www.digitalvizyon.net](http://www.digitalvizyon.net)



## İçindekiler

Eğitim Hakkında.....	3
Neler Öğreneceksiniz? .....	3
Ön Koşullar .....	4
Kimler Katılmalı.....	4
Outline .....	4
Java Platform Overview .....	4
Java Syntax and Class Review .....	4
Encapsulation and Subclassing.....	5
Overriding Methods, Polymorphism, and Static Classes.....	5
Abstract and Nested Classes.....	5
Interfaces and Lambda Expressions .....	5
Collections and Generics .....	5
Collections Streams, and Filters .....	6
Lambda Built-in Functional Interfaces.....	6
Lambda Operations .....	6
Exceptions and Assertions.....	6
Java Date/Time API.....	7
I/O Fundamentals.....	7
File I/O (NIO.2).....	7
Concurrency .....	7
The Fork-Join Framework .....	7
Parallel Streams.....	7
Database Applications with JDBC.....	8

## Eđitim Hakkında

JAVA SE 17 Programming Eđitimi, Java Standard Edition 17 ya da Java 17 olarak adlandırılan Java programlama dilinin 17. sürümüdür. Java SE 17, Java geliştiricileri için güncellenmiştir. Bu nedenle, araçlar ve kütüphaneler sunar. Bununla birlikte, Java SE 17, uygulama geliştirme, veri işleme ve sunucu tarafı uygulamalar gibi birçok farklı amaç için kullanılabilir. Java SE 17 Eđitimi, Java SE 17 sürümünün özelliklerini, araçlarını ve kütüphanelerini kullanarak uygulama geliştirme becerilerini geliştirmeyi amaçlar.

Java SE 17 eđitimi, Java SE 17 sürümünün özelliklerini ve öğrenme yolunu anlatan bir programlama eđitimidir. Bunun yanı sıra, Java SE 17 eđitimi, Java programlama dilinin temel kavramlarını, söz dizimi ve yapısını öğrenmeyi amaçlamaktadır. Böylece katılımcılar Java SE 17 sürümü ile uygulama geliştirebilmektedir. Ayrıca, Java SE 17 eđitimi, katılımcıların Java SE 17 ile veri yapılarını, sınıfları, nesnelere, veri tiplerini, dosya işlemlerini, thread'leri ve diğer avantajları kullanma becerilerini geliştirmeyi amaçlar.

Java SE 17 eđitimi, Java'nın en güncel sürümünün özelliklerini kapsamlı bir şekilde öğrenmek isteyen programcılara yönelik olarak tasarlanmıştır. Bu eđitim, Java'nın temel kavramlarına ek olarak, Java SE 17'nin yeni özelliklerini, iyileştirmelerini ve bu sürümle gelen güncellemeleri derinlemesine inceler.

Eđitimde, Java dilinin sözdiziminden ve yapısından başlamaktadır. Bununla birlikte, Java SE 17 ile gelen yeni özelliklerin ve yeniliklerin ayrıntılı bir şekilde anlatılması hedeflenmektedir. Bu, katılımcılara, Java SE 17 ile etkileyici ve güçlü uygulamalar geliştirmek için gereken bilgi ve becerileri kazandırmaktadır.

Java SE 17'nin performans iyileştirmeleri, güvenlik güncellemeleri ve yeni API'ler gibi konular da bu eđitimin önemli parçalarıdır. Böylece, katılımcılar, Java SE 17'nin getirdiđi avantajları anlama ve uygulamalarına entegre etme becerisini kazanırlar.

Sonuç olarak, JAVA SE 17 Programming Eđitimi, programcılara Java'nın en yeni sürümünün özelliklerini, kullanımını ve uygulama geliştirmede nasıl kullanabileceklerini ayrıntılı bir şekilde anlatır. Bu, onların mevcut bilgilerini genişletir ve Java SE 17 ile uygulama geliştirme becerilerini artırır.

## Neler Öğreneceksiniz?

Java SE 17 programlama eđitiminde şunları öğrenebilirsiniz:

- Java dilinin temel kavramlarını
- Nesne yönelimli programlama prensiplerini
- Dosya işlemleri ve veri yapılarını
- Exception handling ve hata yönetimini
- Multithreading ve parallel programming'i
- GUI (Grafiksel Kullanıcı Arayüzü) programlamayı
- Veri tabanı erişimi ve SQL'i



- Network programming ve web servislerini
- Java SE 17 özellikleri ve fonksiyonaltelerini öğreneceksiniz.

Bu eğitimde öğrenilen konuların dışında, öğrenciler Java programlama dillerinin yaygın kullanım alanlarını ve uygulamalarını da öğrenebilirler.

## Ön Koşullar

Java SE 17 programlama eğitimi için genel olarak şu ön koşulların olması beklenir:

- Programlama temelleri: Variables, loops, conditional statements, functions vb.
- Bilgisayar bilgisi: Dosya sistemi, veri yapıları, veri tabanları vb.
- İngilizce bilgi: Eğitim dili genellikle İngilizce olduğundan İngilizce bilgi bir avantajdır.

Ancak bu ön koşullar her eğitim verilen yer ve verilen eğitim programı için geçerli olmayabilir. Eğitim verilen yerin veya eğitim programının belirlemiş olduğu ön koşulları öğrenmek için o yeri veya programı kontakt etmeniz önerilir.

## Kimler Katılnmalı

Java SE 17 programlama eğitimi, aşağıdaki kişiler için uygun olabilir:

- Programlama dilleri hakkında ilgi duyan başlangıç seviyesi programcılar
- Nesne yönelimli programlama prensiplerini öğrenmek isteyenler
- Java ile yazılmış uygulamaların geliştirilmesini veya bu uygulamaları kullanmak isteyen profesyoneller
- Gelişmiş Java programlama konuları hakkında eğitim almak isteyenler
- Başka programlama dilleri kullanmış olanlar, Java'yı öğrenmek isteyenler

Eğitim, Java SE 17 programlama dili hakkında daha detaylı bilgi edinmek isteyen herkes tarafından katılabilir. Ancak, programlama dilleri hakkında biraz deneyiminin olması beklenir.

## Outline

### Java Platform Overview

- 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

### Java Syntax and Class Review

- 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 Subclassing

- 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 virtual method invocation
- Using varargs to specify variable arguments
- 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
- Implementing an ArrayList
- Implementing a TreeSet
- Implementing a HashMap
- Implementing a Deque
- Ordering collections

## Collections Streams, and Filters

- Describing the Builder pattern
- Iterating through a collection using lambda syntax
- Describing the Stream interface
- Filtering a collection using lambda expressions
- Calling an existing method using a method reference
- Chaining multiple methods together
- Defining pipelines in terms of lambdas and collections

## Lambda Built-in Functional Interfaces

- Listing the built-in interfaces included in `java.util.function`
- Core interfaces – Predicate, Consumer, Function, Supplier
- Using primitive versions of base interfaces
- Using binary versions of base interfaces

## Lambda Operations

- Extracting data from an object using map
- Describing the types of stream operations
- Describing the Optional class
- Describing lazy processing
- Sorting a stream
- Saving results to a collection using the collect method
- Grouping and partition data using the Collectors class

## Exceptions and Assertions

- Defining the purpose of Java exceptions
- Using the try and throw statements
- Using the catch, multi-catch, and finally clauses
- Autoclose 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

## The Fork-Join Framework

- Parallelism
- The need for Fork-Join
- Work stealing
- RecursiveTask

## Parallel Streams

- Reviewing the key characteristics of streams
- Describing how to make a stream pipeline execute in parallel
- List the key assumptions needed to use a parallel pipeline
- Defining reduction
- Describing why reduction requires an associative function



- Calculating a value using reduce
- Describing the process for decomposing and then merging work
- Listing the key performance considerations for parallel streams

## 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