

ALGORİTMA VE PROGRAMLAMA I

HAFTA 4 PROGRAMLAMA DİLLERİ

Ders İçeriği: Hafta 4

- Programlama Dili
- Programlamanın Genel Kavramları
- Java Programlama Dili
- Java Paketleri

Programlama Dili



Program

- Belirli bir işi gerçekleştirmek için gerekli komutlar dizisi olarak tanımlanabilir.



Programlama

- Bir programı oluşturabilmek için gerekli komutların belirlenmesi ve uygun biçimde kullanılmasıdır.

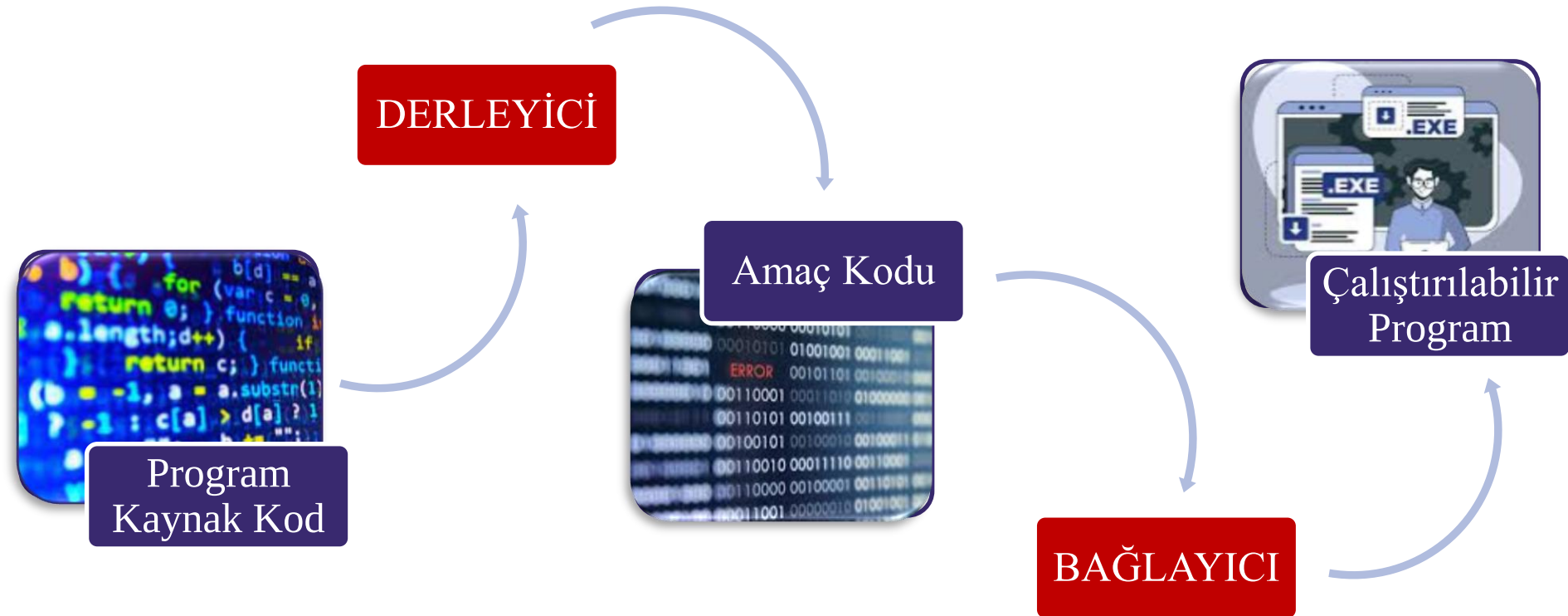


Programlama Dilleri

- Bir programın oluşturulmasında kullanılan komutlar, tanımlar ve kuralların belirtildiği programlama araçlarıdır.
- Bilgisayara ne yapmaları gerektiğini söylememizi sağlayan özel bir dil.

Programlamanın Genel Kavramları

➤ Bilgisayar Programlarının Temel Çalışma Mantığı



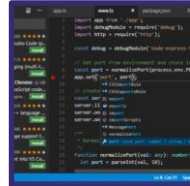
Programlamanın Genel Kavramları

Kaynak Kod

```
1 #include <stdio.h>
2 int main()
3 {
4     char *name = "Ali";
5     printf("Merhaba %s", name);
6     return 0;
7 }
```

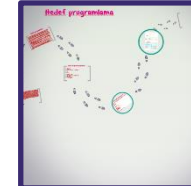
- Herhangi bir yazılım, makine diline dönüştürülerek işlenip yorumlanmasından önce insanların okuyabildikleri ve üzerinde çalışabildikleri kodlar olarak tanımlanabilmektedir.

Kod Editörü



- Yazılım geliştirmek isteyen kullanıcıların bu yazılımları ortaya çıkarmak için kod yazarak üzerinde çalıştığı platformdur. Her yazılımcının kod yazmak için bir düzenleyiciye (kod editörüne) ihtiyacı vardır.

Amaç Program



- Kaynak programın derlenmesi ile elde edilen makine diline dönüştürülmüş programa amaç program (object program) denir. Programa dili ile yazılan kaynak programları makine dili ile yazılan amaç programlara çevirmek için derleyiciler kullanılır.

Programlamanın Genel Kavramları



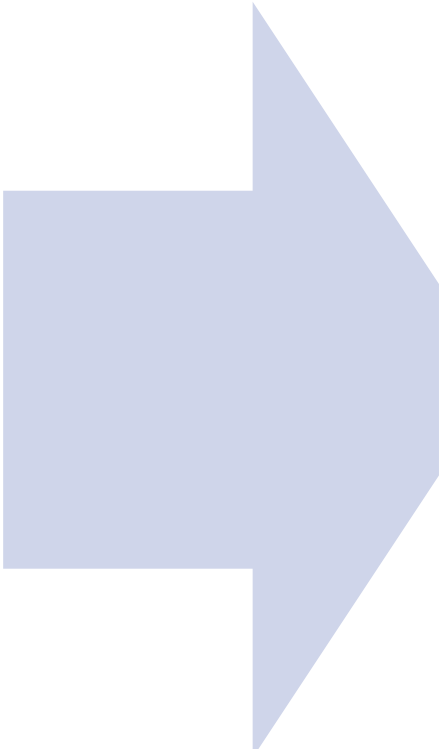
DERLEYİCİ: Bir dildeki kodu (kaynak kodu yada source code) başka dile çeviren programdır. Genelde üretilen bu kod ortama göre çalıştırılabilir kod (executable code) olarak üretilmektedir. Örneğin C dilinde yazılan bir programı PASCAL diline çeviren programlara derleyici adı verilebilir.



YORUMLAYICI: Yazılımı kısım kısım ele alarak doğrudan çalıştırır. Yorumlayıcılar standart bir çalıştırılabilir kod üretmezler. Yorumlama işlemi aşama aşama yapılmadığı için genellikle ilk hatanın bulunduğu yerde programın çalışması kesilir.

Programlamanın Genel Kavramları

➤ Derleyici ve Yorumlayıcı Farkı



Derleyiciler, yorumlayıcılara göre genellikle daha hızlı çalışır. Bunun nedeni, derleyicilerin programın tüm kodunu bir seferde alıp makine diline çevirip çalıştırmasıdır. Bu süreçte eğer hata bulunursa, tüm hatalar programcıya tek seferde rapor edilir. Yorumlayıcılar ise farklı çalışır; kodu satır satır yorumlayarak her bir satırı anında çalıştırır ve karşılaştığı ilk hatayı bildirir. Bu, hatanın çözülene kadar diğer hataların tespit edilememesi anlamına gelir, çünkü yorumlayıcılar satır satır işlem yapmaktadır.

Programlamanın Genel Kavramları

- **IDE (Integrated Development Environment – Tümleşik Geliştirme Ortamı)**
 - Program geliştiriciler için bir platform ve yazılım türüdür.
 - Hızlı ve rahat programlama imkanı sunar.
 - Geliştirme sürecini organize eden çeşitli araçları içerir.
 - Verimli bir geliştirme süreci için gerekli tüm araçları bir arada barındırır.

Programlamanın Genel Kavramları

- **IDE (Integrated Development Environment – Tümleşik Geliştirme Ortamı)**
- Tümleşik geliştirme ortamlarında bulunması gereken temel özellikler:
 - Programlama diline uygun sözdizimi renklendirmesi yapabilen kod düzenleyici.
 - Kod dosyalarının hiyerarşik yapıda gerçek zamanlı olarak görüntülenebilmesi için dosya tarayıcı.
 - Tümleşik derleyici, yorumlayıcı ve hata ayıklayıcı.
 - Yazılımı derleme, bağlama ve çalıştırma gibi süreçleri otomatikleştiren yardımcı araçlar.

Java Programlama Dili

➤ Kısa Tarihçe:

Makine
Dili

```
mov r0,#1
mov r1,#1
l:
add r2,r0,r1
str r2,[r3]
add r3,#4
mov r0,r1
mov r1,r2
b l
```

Assebley
Dili

Yapısal
Diller

Nesneye
Yönelik
Diller

Java Programlama Dili

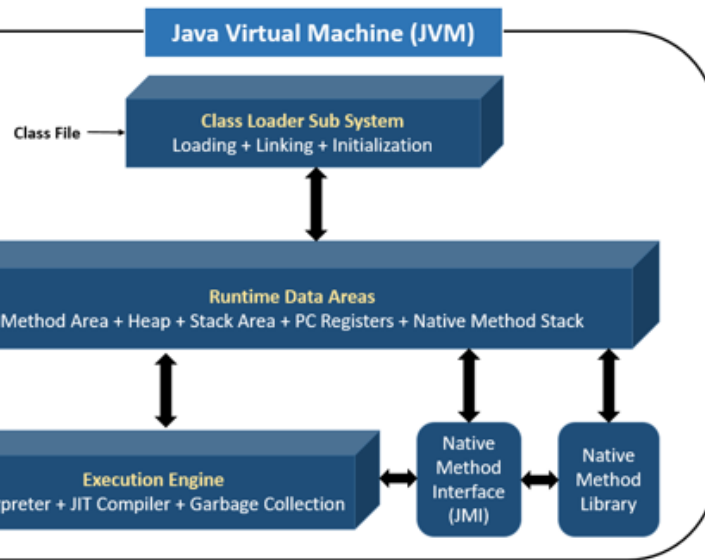
➤ Java'nın Doğuşu:

- 1995 yılında, Sun Microsystems'tan James Gosling, her platformda çalışabilen Java programlama dilini geliştirdi.
- Bu amaçla, farklı işletim sistemleri ve donanımlarda çalışabilen ortak bir sanal makine tasarlandı.
- JVM (Java Virtual Machine – Java Sanal Makinesi) adı verilen bu sanal makine, ücretsiz olarak sunuldu.
- Programcılar, platform bağımsız yazılım geliştirebilir hale geldi ve işlemciye göre kodun çevrilmesi işi JVM tarafından yapıldı.

Java Programlama Dili

➤ Java'nın Çalışma Prensipleri

Java Development Kit (JDK)



Class
Libraries



Compiler

Debugger

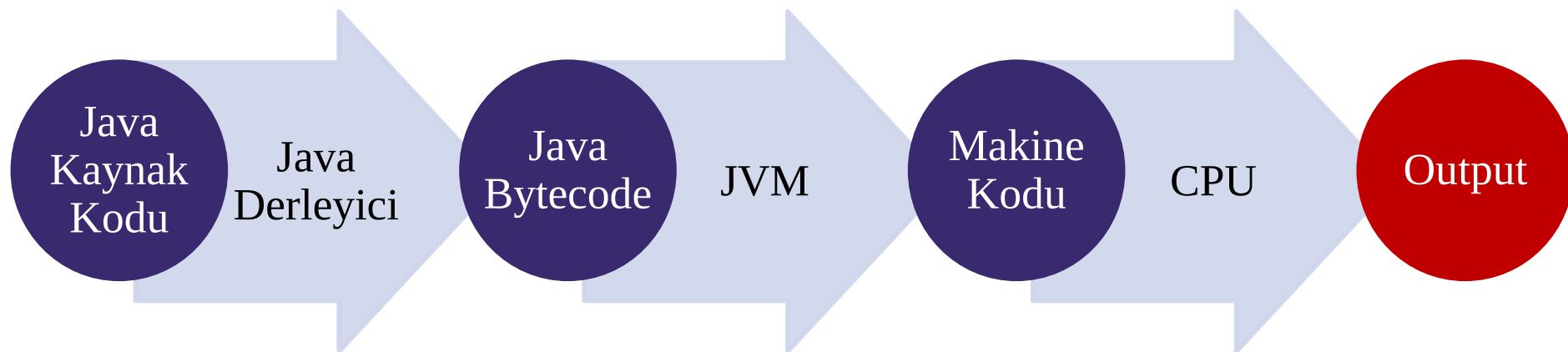
JavaDoc

Java Runtime Environment (JRE)

Java Programlama Dili

➤ Java Sanal Makinesi (JVM)

- Java kodunu derlemek ve yürütmekle sorumlu soyut bir makinedir.
- Java kaynak kodunun platformdan bağımsız çalışmasını sağlar.
- "Bir kez yaz, her yerde çalıştır" mantığına dayanır.
- Java kaynak kodunu makine koduna çevirmek için JIT (Just-in-Time) derleyicisini kullanır.



Java Programlama Dili

- **Java Çalışma Ortamı (JRE – Java Runtime Environment)**
 - Java programlarını bağımsız olarak çalıştırmak için JVM ve sınıf kütüphanelerini içeren bir yazılımdır.
 - Kod çalıştırma işlevine sahiptir, ancak tam bir uygulama geliştirme deneyimi için Java Geliştirme Kiti (JDK) ile birlikte gelir.
 - Hata ayıklayıcı gibi geliştirme araçlarını içermez.
 - util, math, AWT, lang gibi önemli paket sınıflarını ve çeşitli çalışma zamanı kütüphanelerini barındırır.

Java Programlama Dili

➤ Java Geliştirme Kiti (JDK – Java Development Kit)

- Java programlama dilini kullanarak uygulama geliştirmek için eksiksiz bir yazılım ortamıdır.
- Platforma bağımlıdır, bu nedenle farklı işletim sistemlerine uygun sürümleri mevcuttur.
- Java programlarının yazılması, derlenmesi ve çalıştırılmasını sağlar.
- Derleyici, hata ayıklayıcı, uygulama görüntüleyici ve Java uygulama başlatıcı gibi bileşenler içerir.
- Java uygulama başlatıcısı, JRE'yi yükleyip çalıştırır ve ardından programın ana metodunu çalıştırmak için gerekli detayları yükler.

Java Programlama Dili

➤ Neden Java?

- **Basit:** Öğrenmesi ve kullanması kolaydır.
- **Mimari Yapıdan Bağımsız:** Farklı platformlarda sorunsuz çalışır ("Bir kez yaz, her yerde çalıştır").
- **Nesne Yönelimli:** Yeniden kullanılabilir ve yönetilebilir kod yazmayı teşvik eder.
- **Taşınabilir:** JVM sayesinde her ortamda çalıştırılabilir.
- **Dağıtık:** Ağ tabanlı uygulamalar geliştirmek için idealdir.
- **Yüksek Performanslı:** JIT derleyici sayesinde hızlı çalışır.
- **Çoklu İş Yeteneği:** Çoklu iş parçacığı desteği ile aynı anda birden fazla iş yapabilir.
- **Sağlam:** Güçlü hata yakalama mekanizmaları ve bellek yönetimi sunar.
- **Dinamik:** Değişikliklere hızlı uyum sağlayan esnek yapıya sahiptir.
- **Güvenli:** Güvenlik özellikleri sayesinde ağ tabanlı uygulamalar için uygundur.

Java Programlama Dili

➤ Java API (Application Programming Interface)

- Java yazılımlarında kullanılan yazılım kütüphanelerinin genel adıdır.
- Java API, disk, grafik, ağ, veri tabanı ve güvenlik gibi birçok alanda kullanıcıların erişim ve işlem yapmasına olanak tanır.
- Java API, J2SDK'nın (“Java 2 Software Development Kit”) bir parçasıdır.
- Farklı uygulama ortamlarını desteklemek için farklı API'lar tanımlanmıştır:
 - **Java Platform, Micro Edition (Java ME):** Avuç içi cihazlar gibi kısıtlı kaynağa sahip ortamlarda kullanılır.
 - **Java Platform, Standard Edition (Java SE):** Masaüstü bilgisayarlar ve iş istasyonları için uygundur.
 - **Java Platform, Enterprise Edition (Java EE):** Geniş dağıtık sistemler ve web sunucuları için tasarlanmıştır.

Java Programlama Dili

➤ Java Dilinin Başlıca Uygulamaları

- **Sınıf Kütüphaneleri:** Başka uygulamalarda kullanılacak kütüphane sınıfları yaratır.
- **Konsol Uygulamaları:** Satır komutu arayüzü için görsel Java uygulamaları yaratır.
- **Web Uygulamaları:** Web kullanıcı arayüzü için görsel Java uygulamaları yaratır.
- **XML Uygulamaları:** xml Web servisleri yaratır.
- **Veri tabanı Uygulamaları:** Veri tabanlarını yönetmek için uygulama programları yaratır.
- **Mobile Web Uygulamaları:** Cep telefonu, tablet gibi taşınabilir cihazlar için uygulama programları yaratır.
- **Oyun Programları:** Java ile gelişmiş oyun programları yazılır.

Java Programlama Dili

➤ Java Kaynak Dosyasının Yapısı

- Programlama kodlarının yazıldığı dosyalara kaynak dosyası denir.
- Java kaynak dosyaları .java uzantısıyla kaydedilir.
- Java kaynak dosyalarında önemli bir kural, dosyanın içinde dosyayla aynı adı taşıyan bir sınıf veya arayüz bulunması gerektiridir.
- Bir java kaynak dosyası üç ana bölümden oluşur. Bunlar:
 - Paket Bildirimi
 - Kütüphane Ekleme
 - Kod Bloğu

Java Programlama Dili

➤ Java Kaynak Dosyasının Yapısı

■ Paket Bildirimi

- Java'da kaynak dosyalar ve derlenmiş halleri, dosya sistemine benzer bir hiyerarşide tutulur.
- Tüm dosyaların bir dizinde tutulması gibi, Java kaynak dosyaları da bir paket içinde yer alır ve bu, package anahtar kelimesi ile tanımlanır.
- Paketler, başka paketlerin içinde de bulunabilir.
 - Örneğin, Java'da dosya işlemleri yapan File sınıfı, java paketinin içindeki io (input/output) paketine aittir. Bu nedenle, File sınıfının en üst kısmında package java.io; ifadesi yer alır.
 - Bir kaynak dosya yalnızca bir pakete ait olabilir; bu nedenle her dosyada sadece bir adet paket bildirimi bulunur.

Java Programlama Dili

➤ Java Kaynak Dosyasının Yapısı

■ Kütüphane Ekleme

- Daha önceden yazılmış ve programcının kullanımına sunulmuş kaynak dosyalar yada bunların derlenmiş hallerine kütüphane denilmektedir.
- **import** anahtar kelimesi ile kütüphane eklenir.
- Java'da kütüphane bildirimi, paket bildiriminin ardından yapılmaktadır:
 - Örneğin az önce bahsettiğimiz File kaynak dosyasından üretilmiş File sınıfını yeni yazacağımız kodda kullanmak istersek «import java.io.File;» ifadesini kullanmamız gerekmektedir.

Java Programlama Dili

➤ Java Kaynak Dosyasının Yapısı

■ Kod Bloğu

- Kaynak kodunun paket bilgisi tanımlandıktan sonra ilgili kütüphaneler eklenir.
- Kaynak dosyası içerisinde program kodlarının yazıldığı kısma kod bloğu denir.
- Süslü parantezler «{ ... }» arasına yazılan programlama deyimleri kod bloklarını oluşturur.
- Sınıf, fonksiyon, değişken tanımı gibi tüm kod blokları paket bilgisinin ardından yazılmalıdır.

Java Paketleri

- Java kütüphanesinde çok sayıda metot ve değişken tanımları vardır.
- Metotların ve değişkenlerin her biri bir sınıf içindedir.
- Metotları ve değişkenleri içeren sınıfların sayıları da çok fazladır.
 - Erişimi kolaylaştırmak için sınıflar gruplara ayrılmışlardır.
 - Birbirleriyle ya da aynı konuyla ilişkili olan sınıflar bir grup içinde toplanır.
 - Bu grupların her birine bir paket denir. Başka bir deyişle, bir paket birbirleriyle ilişkili sınıfları içeren bir ambardır (dizin).

Java Paketleri

- Örneğin giriş/çıkış ile ilgili olan bütün sınıflar java.io paketi içindedir.
- Bir pakette aynı adı taşıyan iki sınıf olamaz.
- Ama farklı paketler içinde aynı adı taşıyan sınıflar olabilir.
- Paket adları sınıfların karışmasını önler.
- İleride programların sayısı arttıkça, sınıf adlarının birbirleriyle karışmasını önlemek için birbirleriyle ilgili sınıfları aynı pakete koymayı alışkanlık haline getirmeniz gerekir.

Java Paketleri

➤ Çok Kullanılan Java Paketleri

java.lang

Java sınıf paketleri arasında en önemlilerinden birisidir. import deyimi yazmaya gerek kalmaksızın, her java programına kendiliğinden girer. Java dilinin özünü oluşturan sınıfları içerir.

java.io

Giriş-çıkış işlemlerini yapan sınıfları içerir.

java.text

Metinleri işleyen sınıflardan oluşur. Farklı dil ve kültürlerle göre tarih, sayı ve iletilerin yazılmasını sağlar.

java.math

Duyarlı matematiksel işlemler yapmak için gerekli sınıfları içerir.

java.awt

Kullanıcı arayüzü (UI-user interface) tasarımı ve olay-yönlendirici (event-handler) için gerekli standart öğeleri içerir.

Java Paketleri

➤ Çok Kullanılan Java Paketleri

<code>javax</code>	Java dilinin genişlemesini sağlayan sınıfları içerir.
<code>java.applet</code>	Java appletleri yaratmak için gerekli sınıfları içerir.
<code>java.sql,</code> <code>javax.sql</code>	JDBC API yardımıyla veri tabanlarına erişimini ve veri işlemeyi sağlayan sınıfları içerirler.
<code>java.net</code>	Ağ (network) uygulamaları yaratmak için gerekli sınıfları içerir.
<code>java.security</code>	Kriptografik güvenlik için gerekli sınıfları içerir.

Java Paketleri

➤ Çok Kullanılan Java Paketleri

■ java.io Paketi

- Bir program koşarken, sistemin dış çevre birimlerinden veri alması ya da onlara veri göndermesi gerekebilir.
- Kullanıcının bilgisayarla etkileşim içinde olabilmesi için, onunla iletişim kurabilmelidir.
- Bunun tek yolu, bilgisayara girdi (input) göndermek ve ondan çıktı (output) almaktır.
- Hemen her dil bu iş için çeşitli yöntemlere sahiptir. Giriş/Çıkış birimleri dediğimiz birimler bu işe yarar.
- Örneğin, klavye, dosya, uzaktaki bilgisayar, ağ gibi araçlar bilgisayara girdi yapabilirler. Ekran, yazıcı, dosya, ses-aygıtları, ağ gibi araçlarla bilgisayardan çıktı alınabilir.

Java Paketleri

➤ Çok Kullanılan Java Paketleri

■ java.io Paketi

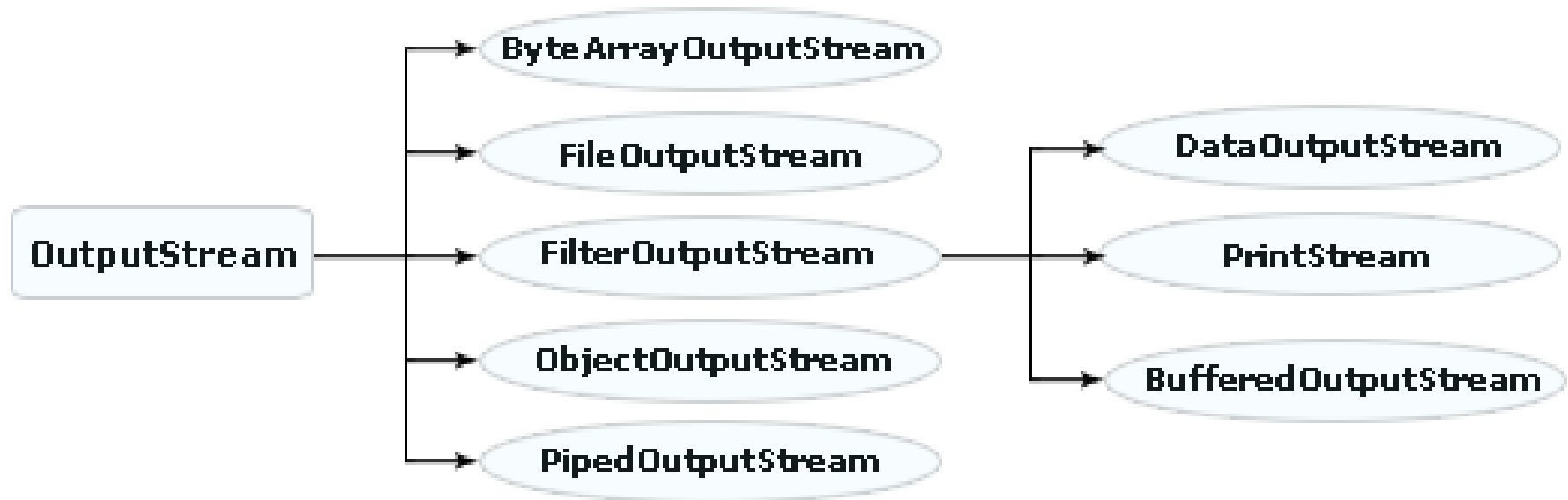
- Bu paketin içerdiği iki önemli sınıf:
 - InputStream
 - OutputStream
- Stream (akım) sözcüğü sisteme giren ya da sistemden çıkan karakter dizileridir.
- Bütün veriler birer karakter dizisi olarak sisteme girer veya sistemden çıkarlar.
- InputStream sınıfı girişleri, OutputStream sınıfı çıkışları düzenler.

Java Paketleri

➤ Çok Kullanılan Java Paketleri

▪ java.io Paketi

OutputStream Sınıfının alt sınıfları

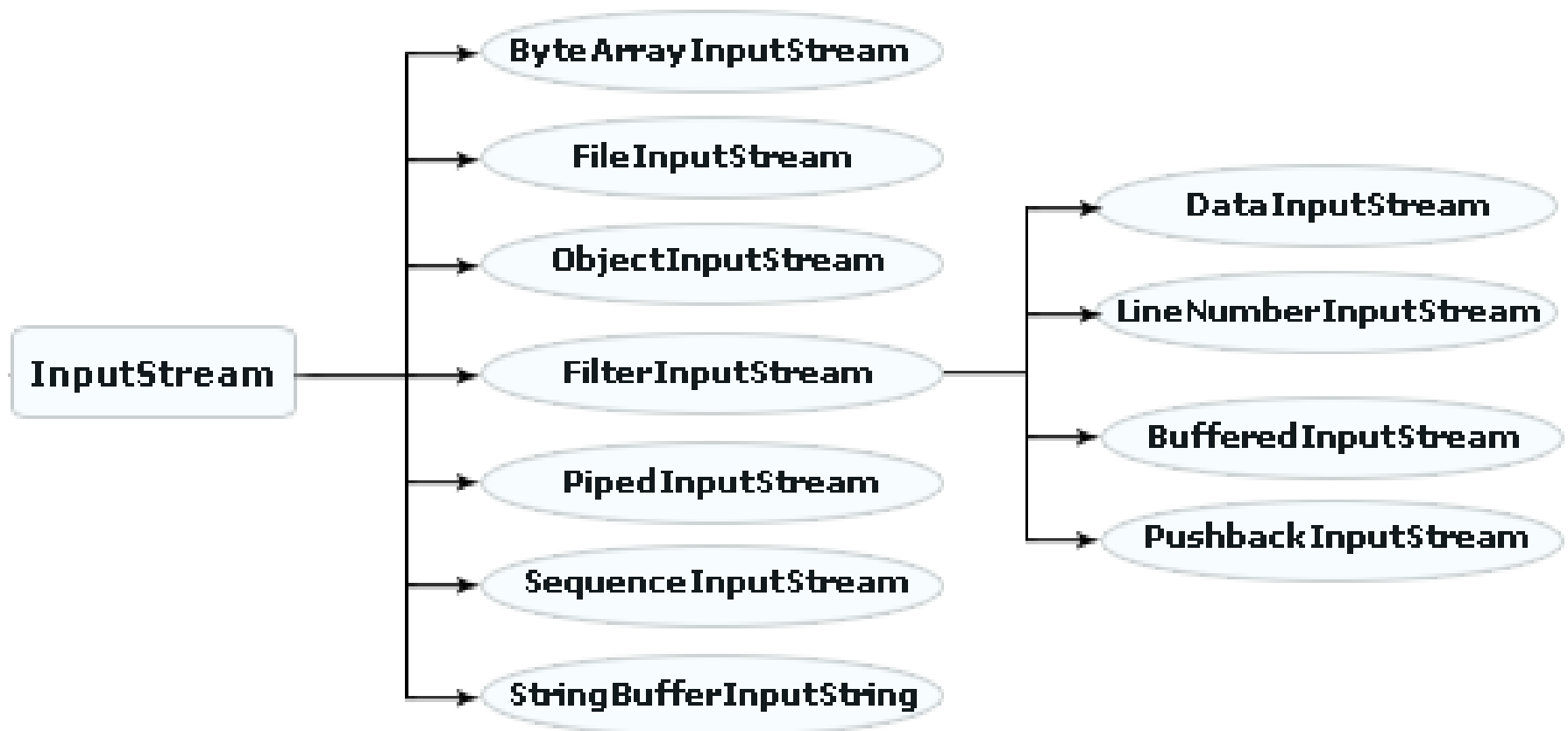


Java Paketleri

➤ Çok Kullanılan Java Paketleri

▪ java.io Paketi

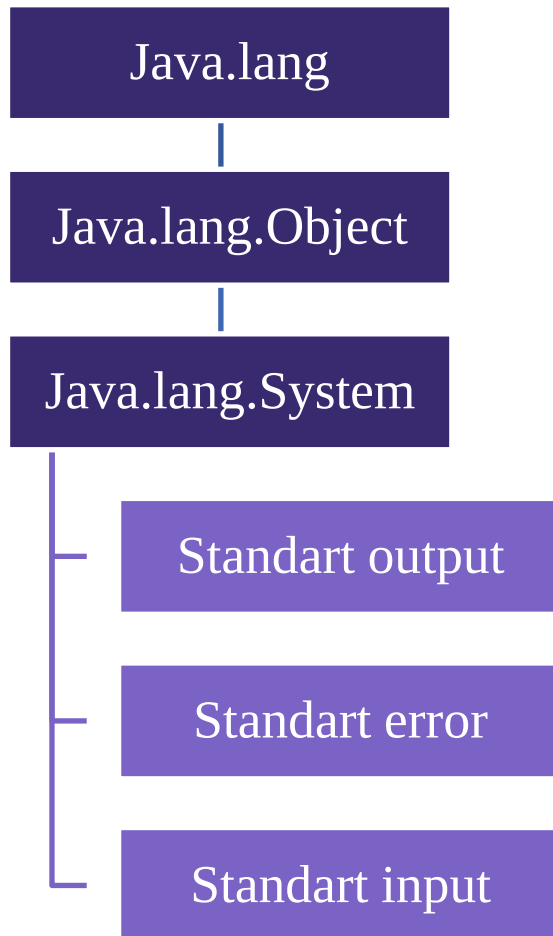
InputStream Sınıfının alt sınıfları



Java Paketleri

➤ Çok Kullanılan Java Paketleri

▪ java.io Paketi – System Sınıfı



- Standart giriş-çıkış işlemlerini yapan ve hata iletilerini veren bir sınıftır.
- Genel olarak, sistemin dışarıyla olan ilişkilerini yönetir.
- Çevre değişkenlerine erişir, veri ve dosya girdi-çıkıtısından sorumludur.
- Bunları yapabilmesi için, `java.io` paketi içindeki `InputStream` ve `OutputStream` sınıflarını kullanır (implement).

Java Paketleri

➤ Çok Kullanılan Java Paketleri

- java.io Paketi – System Sınıfı

- G/Ç akım yolları (I/O Streams)

- Bir java uygulaması başladığında, Sytem sınıfı kendiliğinden in, out, error giriş/çıkış akım-yolunu açar.
- Bir java uygulaması başlayınca, kullanıcının klavye ile yazdıkları in akım-yoluna atanan değerdir. Sistem bu girdiyi algılar.
- Sistemin çıktıları out akım-yoluna atanan değerdir. O değer ekrana yazılır ve kullanıcıya iletilmiş olur.
- Bu işlemler sırasında bir hata oluşursa, o hata error akım-yoluna atanır ve otomatik olarak ekrana yazılır.

Java Paketleri

➤ Çok Kullanılan Java Paketleri

- java.io Paketi – System Sınıfı
- G/Ç için kullanılan metotlar

<code>print()</code>	Standart çıkışa (ekran) çıktı gönderir.	<code>System.out.print(parametre);</code>
<code>println()</code>	<code>print()</code> metodu ile aynı işi yapar, ancak çıktıyı gönderdikten sonra ayrıca satırbaşı yapar.	<code>System.out.println(parametre);</code>
<code>printf()</code>	Konsola biçimli (formatlı) çıktı gönderir.	<code>System.out.printf(parametre);</code>