10. Ders: JAVA ile Nesne Yönelimli Programlama Interfaces (Arayüzler)

Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Bölümü

YMH112 Algoritma ve Programlama-II

Dr. Öğr. Üyesi Yaman Akbulut

JAVA ile Nesne Yönelimli Programlama

- http://www.kriptarium.com/algoritma.html (Yardımcı kaynak)

- JAVA ile Nesne Yönelimli Programlama

  - Ders 29: Java Polimorfizmi / Yöntem Aşırı Yüklemesi (Method Overloading) (izle)
  - Ders 30: Java Polimorfizmi / Yöntem Geçersiz Kılma (Method Overriding) (izle)
  - Ders 31: super Anahtar Sözcüğünün Kullanımı (izle)
  - Ders 32: final Anahtar Sözcüğünün Kullanımı (izle)
  - Ders 33: Soyut Sınıf (Abstract Class) (izle)
  - Ders 34: Arayüz (Interface) (izle)

Java Keywords

abstract	double	int	super
assert	else	interface	switch
boolean	enum	long	synchronized
break	extends	native	this
byte	final	new	throw
case	finally	package	throws
catch	float	private	transient
char	for	protected	try
class	goto	public	void
const	if	return	volatile
continue	implements	short	while
default	import	static	
do	instanceof	strictfp	

Kaynak: Introduction to JAVA Programming 10th Edition, Y. Daniel Liang

Arayüz (Interface)

Arayüz (interface), yalnızca sabitler ve soyut metotlar içeren sınıf benzeri bir yapıdır.

Birçok yönden, bir arayüz soyut bir sınıfa benzer, ancak amacı ilgili sınıfların veya ilgisiz sınıfların nesneleri için ortak davranışı belirlemektir.

Örneğin, uygun arayüzleri kullanarak, nesnelerin karşılaştırılabilir olup olmadığını, yenilebilir olup olmadığını veya klonlanabilir olup olmadığını belirleyebiliriz.

Arayüz (Interface)

Java, bir arayüzü (interface) bir sınıftan ayırmak için ve bir arayüzü tanımlamak için aşağıdaki sözdizimini kullanır:

```
erişim interface ArayuzAdi {
    /** Sabit tanımlamaları */
    /** Abstract metot imzaları */
}


public interface Yenilebilir {
    /** Nasıl yenileneceğini tanımla */
    public abstract String nasilYenir();
}
```

Arayüz (Interface)

Java'da arayüz özel bir sınıf gibi ele alınır.

Her arayüz, normal bir sınıf gibi ayrı bir byte kodu dosyasında derlenir.

Bir arayüzü, soyut bir sınıfı kullandığınız gibi hemen hemen aynı şekilde kullanabiliriz.

Örneğin, bir referans değişkeni için veri türü olarak, çevrimin sonucu olarak bir arayüz vb. kullanabiliriz.

Soyut bir sınıfta olduğu gibi, new operatörünü kullanarak bir arayüzden örnek oluşturamayız.

Arayüz (Interface)

Bir nesnenin yenilebilir olup olmadığını belirlemek için **Yenilebilir** arayüzünü kullanabiliriz.

Bu, nesne sınıfının **implements** anahtar sözcüğünü kullanarak bu arayüzü uygulamasına izin vererek gerçekleştirilir.

Örnek 1'deki Tavuk ve Meyve sınıfları (satır 20, 39), Yenilebilir arayüzünü uygular (implement eder).

Sınıf ve arayüz arasındaki ilişki, arayüz kalıtımı olarak bilinir. Arayüz kalıtımı ve sınıf kalıtımı temelde aynı olduğundan, her ikisine de kalıtım olarak değineceğiz.

TestYenilebilir

```java
public class TestYenilebilir {
    public static void main(String[] args) {
        Object[] nesneler = {new Kaplan(), new Tavuk(), new Elma()};
        for (int i = 0; i < nesneler.length; i++) {
            if (nesneler[i] instanceof Yenilebilir)
                System.out.println(((Yenilebilir)nesneler[i]).nasilYenir());

            if (nesneler[i] instanceof Hayvan) {
                System.out.println(((Hayvan)nesneler[i]).ses());
            }
        }
    }
}

abstract class Hayvan {
    /** Return animal sound */
    public abstract String ses();
}

class Tavuk extends Hayvan implements Yenilebilir {
    @Override
    public String nasilYenir() {
        return "Tavuk: Kizartma yap";
    }

    @Override
    public String ses() {
        return "Tavuk: git-git-gidak";
    }
}

class Kaplan extends Hayvan {
    @Override
    public String ses() {
        return "Kaplan: RROOAARR";
    }
}
```

```java
public interface Yenilebilir {
    /** Nasıl yenileceğini tanımla */
    public abstract String nasilYenir();
}
```

```java
abstract class Meyve implements Yenilebilir {
    // Veri alanları, yapıcılar ve metotlar...
}

class Elma extends Meyve {
    @Override
    public String nasilYenir() {
        return "Elma: Elma suyu yapalim";
    }
}

class Portakal extends Meyve {
    @Override
    public String nasilYenir() {
        return "Portakal: Portakal suyu yapalim";
    }
}
```

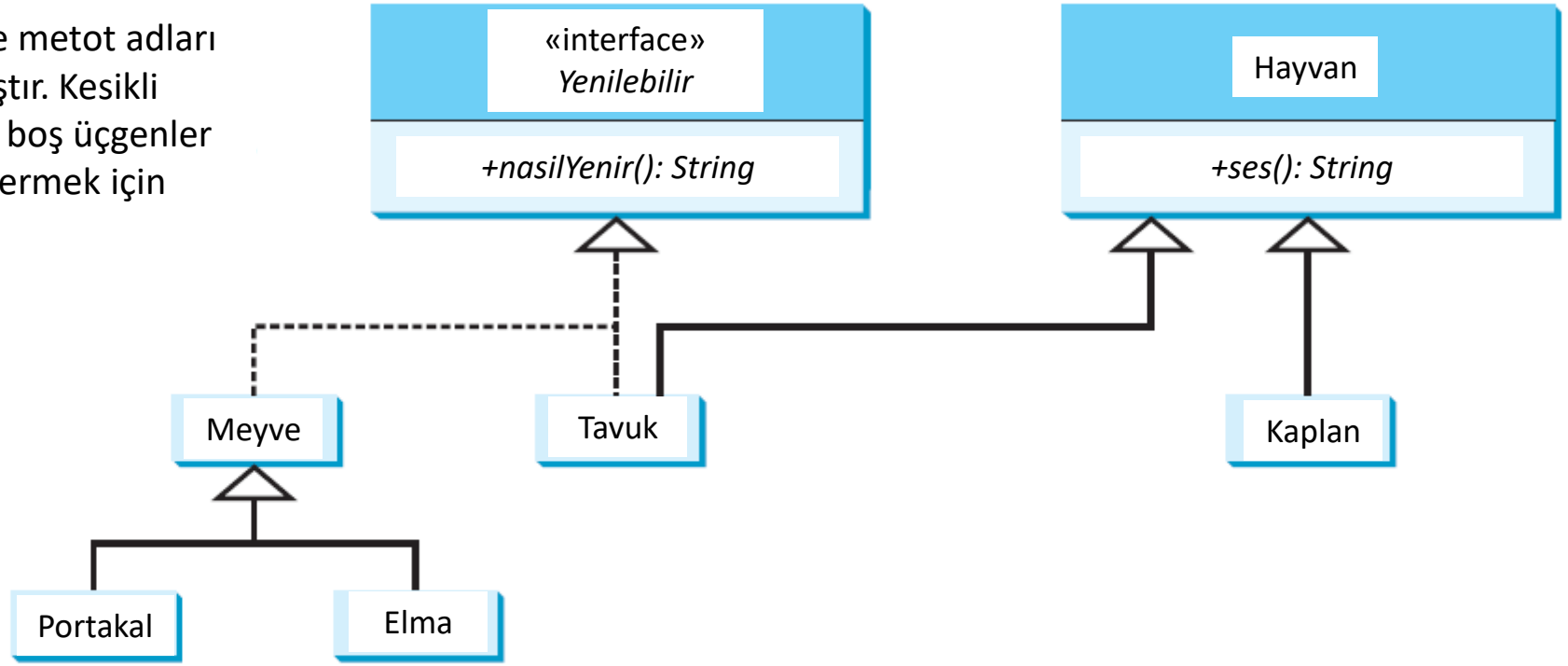
```
C:\Program Files\Java\jdk-16.0.1\bin\yeni2>javac Yenilebilir.java

C:\Program Files\Java\jdk-16.0.1\bin\yeni2>javac TestYenilebilir.java

C:\Program Files\Java\jdk-16.0.1\bin\yeni2>java TestYenilebilir
Kaplan: RROOAARR
Tavuk: Kizartma yap
Tavuk: git-git-gidak
Elma: Elma suyu yapalim
```

Yenilebilir Arayüz (Interface)

Gösterim:
Arayüz adı ve metot adları italik yazılmıştır. Kesikli çizgiler ve içi boş üçgenler arayüzü göstermek için kullanılır.



Yenilebilir, Tavuk ve Meyve için üst tiptir. Hayvan, Tavuk ve Kaplan için üst tiptir. Meyve, Portakal ve Elma için üst tiptir.

Kalıtım

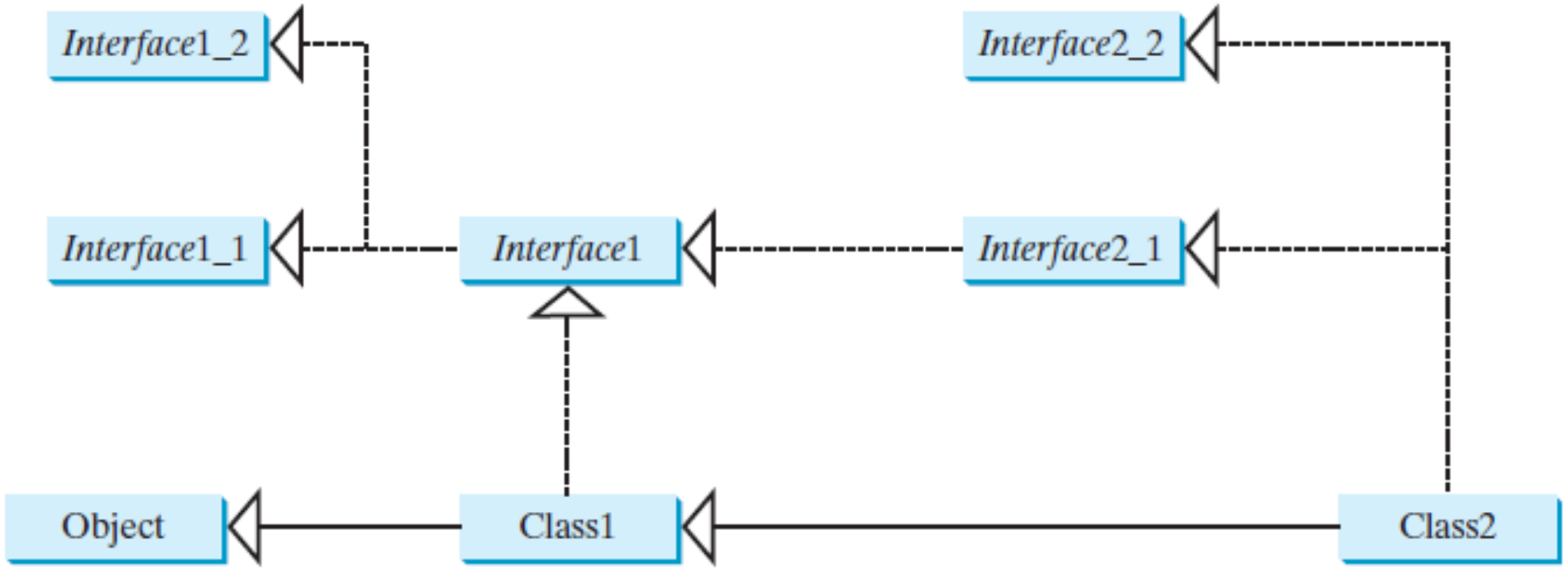


FIGURE 13.7 `Class1` implements `Interface1`; `Interface1` extends `Interface1_1` and `Interface1_2`. `Class2` extends `Class1` and implements `Interface2_1` and `Interface2_2`.

Comparable Interface (Karşılaştırılabilir Arayüz)

Comparable arayüzü, nesneleri karşılaştırmak için compareTo metodunu tanımlar.

İki öğrenci, iki tarih, iki çember, iki dikdörtgen veya iki kare gibi aynı türden iki nesneden daha büyük olanı bulmak için genel bir metot tasarlamak istediğimizi varsayalım.

Bunu başarmak için, iki nesnenin karşılaştırılabilir olması gerekir, bu nedenle nesneler için ortak davranış karşılaştırılabilir olmalıdır.

Java, bu amaç için Comparable arayüzü sağlar.

Comparable Interface (Karşılaştırılabilir Arayüz)

Arayüz şu şekilde tanımlanır:

```
// Interface for comparing objects,
defined in java.lang
package java.lang;

public interface Comparable<E> {
        public int compareTo(E o);
}
```

compareTo metodu, belirtilen o nesnesiyle bu nesnenin sırasını belirler

ve bu nesne o değerinden küçük, ona eşit veya ondan büyükse

negatif bir tamsayı, sıfır veya pozitif bir tamsayı döndürür.

Comparable Interface (Karşılaştırılabilir Arayüz)

Comparable arayüz genel bir arayüzdür.

Genel tip E, bu arayüz uygulanırken somut bir tiple değiştirilir.

Java kütüphanesindeki birçok sınıf, nesneler için doğal bir sıra tanımlamak üzere Comparable'ı uygular.

Byte, Short, Integer, Long, Float, Double, Character, BigInteger, BigDecimal, Calendar, String ve Date sınıflarının tümü Comparable arayüzünü uygular.

Örneğin, Integer, BigInteger, String ve Date sınıfları, Java API'sinde aşağıdaki gibi tanımlanır:

Comparable Interface (Karşılaştırılabilir Arayüz)

Örneğin, Integer, BigInteger, String ve Date sınıfları, Java API'sinde aşağıdaki gibi tanımlanır:

```
public class Integer extends Number
    implements Comparable<Integer> {
  // class body omitted

  @Override
  public int compareTo(Integer o) {
    // Implementation omitted
  }
}
```

```
public class BigInteger extends Number
    implements Comparable<BigInteger> {
  // class body omitted

  @Override
  public int compareTo(BigInteger o) {
    // Implementation omitted
  }
}
```

```
public class String extends Object
    implements Comparable<String> {
  // class body omitted

  @Override
  public int compareTo(String o) {
    // Implementation omitted
  }
}
```

```
public class Date extends Object
    implements Comparable<Date> {
  // class body omitted

  @Override
  public int compareTo(Date o) {
    // Implementation omitted
  }
}
```

SiralaKarsilastirilabilirNesne

```java
import java.math.*;

public class SiralaKarsilastirilabilirNesne {
    public static void main(String[] args) {
        String[] sehirler = {"Savannah", "Boston", "Atlanta", "Tampa"};
        java.util.Arrays.sort(sehirler);
            for (String sehir: sehirler)
                System.out.print(sehir + " ");
        System.out.println();

        BigInteger[] buyukSayilar = {new BigInteger("2323231092923992"),
                                     new BigInteger("432232323239292"),
                                     new BigInteger("54623239292")};
        java.util.Arrays.sort(buyukSayilar);
            for (BigInteger sayi: buyukSayilar)
                System.out.print(sayi + " ");
    }
}
```

```
C:\Program Files\Java\jdk-16.0.1\bin\yeni2>javac SiralaKarsilastirilabilirNesne.java

C:\Program Files\Java\jdk-16.0.1\bin\yeni2>java SiralaKarsilastirilabilirNesne
Atlanta Boston Savannah Tampa
54623239292 432232323239292 2323231092923992
```