if Deyimi

if deyiminin genel biçimi şöyledir:

if (<boolean türden ifade>)
	<deyim>
[else]
	[<deyim>]

if anahtar sözcüğünden sonra parantez içerisinde boolean türden bir ifade bulunmak zorundadır. if deyimi doğruysa ve yanlışsa kısmından oluşur. Her iki kısımda da bir deyim bulunmak zorundadır. Bu deyim herhangi bir deyim olabilir. if deyiminin kendisi dışarıdan bakıldığında tek bir deyimdir. Örneğin:

ifade1;

if (x > 0) {
	ifade2;
	ifade3;
}
else {
	ifade4;
	ifade5;
}
ifade6;

Burada toplam 3(üç) deyim vardır.

if deyimi çalışma sistemi şöyledir: Önce if parantezi içerisindeki ifadenin değeri hesapalanır. Bu değer true ise doğruysa kısmındaki deyim, false ise yanlışsa kısmındaki deyim çalıştırılır. Bundan sonra if deyiminin çalışması biter.

if deyiminde else kısmı olmak zorunda değildir. Örneğin:

```
package csd;

class App {
	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);

		System.out.println("Bir sayı giriniz");
		int val = Integer.parseInt(kb.nextLine());

		if (val > 0)
			System.out.println("Pozitif");

		System.out.println("if sonrası");

		kb.close();
	}
}
```

Derleyici if deyiminin doğruysa kısmından sonra else anahtar sözcüğü gelmiş mi diye bakar. Eğer gelmemişse bunun else kısmı olmayan bir if olduğuna karar verir.

Bazen if deyimi yanlışlıkla boş deyim ile kapatılabilir. Örneğin:

```
if (ifade);
	ifade1;
```

Bu durumda herhangi bir hata oluşmaz ancak bu bir programlama hatasıdır. Bu durumda kod anlamsızlaşır.

Aşağıdaki gibi bir kod doğru olmasına karşın programcı bu biçimde kullanmamalıdır:

```
if (x > 0)
	;
else
	<deyim>
```

if deyiminin doğru kısmında boş deyim yazmak "hiçbir şey yapma" anlamına gelir. Bu işlem

```
if (x <= 0)
	<deyim>
```

şeklinde yapılmalıdır.

if deyiminin doğruysa kısmında başka bir if deyimi bulunabilir. Örneğin:

```
package csd;

class App {
	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);

		System.out.println("Bir sayı giriniz");
		int val = Integer.parseInt(kb.nextLine());

		if (val > 0)
			if (val % 2 == 0)
				System.out.println("Pozitif çift sayı");
			else
				System.out.println("Pozitif tek sayı");

		System.out.println("Program sonu");

		kb.close();
	}
}
```

Bir grup koşuldan biri doğruyken diğerlerinin doğru olma olasılığı yoksa bu koşullara ayrık koşullar denir. Örneğin:

a > 0, a == 0 ve a < 0

koşulları ayrıktır. Ancak:

a > 10 ve a > 8 koşulları ayrık değildir. Ayrık koşulların ayrı if deyimleri ile yapılması kötü tekniktir. Örneğin:

```
package csd;

class App {
	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);

		System.out.println("Bir sayı giriniz");
		int val = Integer.parseInt(kb.nextLine());

		if (val > 0)
			System.out.println("Pozitif");
		if (val == 0)
			System.out.println("Sıfır");
		if (val < 0)
			System.out.println("Negatif");

		kb.close();
	}
}
```

Burada örneğin birinci koşul doğruysa diğerlerine de gereksiz biçimde bakılacaktır. Bunun en iyi yöntemi ayrık koşulların else if biçiminde yapılmalıdır. Örneğin:

```
package csd;

class App {
	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);

		System.out.println("Bir sayı giriniz");
		int val = Integer.parseInt(kb.nextLine());

		if (val > 0)
			System.out.println("Pozitif");
		else if (val == 0)
			System.out.println("Sıfır");
		else
			System.out.println("Negatif");

		kb.close();
	}
}
```

Burada bir else if merdiveni oluşturulmuştur.

İç içe if deyimlerinde tek bir else varsa else kısmı içteki if deyimine ait olmaktadır (dangling else). Örneğin:

```
package csd;

class App {
	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);

		System.out.println("Bir sayı giriniz");
		int val = Integer.parseInt(kb.nextLine());

		if (val > 0)
			if (val % 2 == 0)
				System.out.println("Pozitif çift sayı");
			else
				System.out.println("Pozitif tek sayı");

		System.out.println("Program sonu");

		kb.close();
	}
}
```

Eğer else kısmının dıştaki if deyimine ait olması isteniyorsa bloklama yapılması gerekmektedir. Örneğin:

```
package csd;

class App {
	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);

		System.out.println("Bir sayı giriniz");
		int val = Integer.parseInt(kb.nextLine());

		if (val > 0) {
			if (val % 2 == 0)
				System.out.println("Pozitif çift sayı");
		}
		else
			System.out.println("Pozitif tek sayı");

		System.out.println("Program sonu");

		kb.close();
	}
}
```

Burada bloklama sonrasında artık else kısmı dıştaki if deyimine ait olur. Aşağıdaki gibi bir programda dıştaki if deyiminin else kısmı için herhangi bir bloklama yapmaya gerek yoktur:

```
package csd;

class App {
	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);
```

```
		System.out.println("Bir sayı giriniz");
		int val = Integer.parseInt(kb.nextLine());

		if (val > 0)
			if (val % 2 == 0)
				System.out.println("Pozitif çift sayı");
			else
				System.out.println("Pozitif tek sayı");
		else
			System.out.println("Pozitif değil");

		System.out.println("Program sonu");

		kb.close();
	}
}
```

Burada else kısımlarının hangi if deyimine ait olduğu derleyici tarafından belirlenebilmektedir. Buna göre ilk else kısmı içteki if deyimine ikinci else kısmı ise dıştaki if deyimine aittir.

if deyimi parantezi içerisinde boolean türden değişkenlerin kullanılmasına oldukça sık olarak rastlanmaktadır. Bayrak (flag) değişkenler buna örnek olarak verilebilir. Örneğin boolean türden bir flag değişkeninin true olması durumu aşağıdaki gibi yazılabilir:

```
if (flag)
	<deyim>
```

Burada flag değişkeni boolean türden olduğundan ayrıca true değeriyle karşılaştırılmasına gerek yoktur. Yani aslında,

```
if (flag)
	<deyim>
```

deyimi

```
if (flag == true)
	<deyim>
```

deyimi ile tamamen eşdeğerdir. Benzer biçimde flag değişkenin false olması durumu da aşağıdaki gibi test edilebilir:

```
if (!flag)
	<deyim>
```

Buradaki deyim de

```
if (flag == false)
	<deyim>
```

ile tamamen eşdeğerdir.

Geri dönüş değeri boolean türden olan metotlar da if deyiminin parantezi içerisinde çağrılabilir. Örneğin:

```
package csd;

class App {
	public static boolean isEven(int val)
	{
		return val % 2 == 0;
	}

	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);
		System.out.println("Bir sayı giriniz");
		int val = Integer.parseInt(kb.nextLine());

		if (isEven(val))
			System.out.println("Çift");
		else
			System.out.println("Tek");

		kb.close();
	}
}
```

if (isEven(val)) ile isEven metodu çağrıldıktan sonra geri dönüş değerinin true olması durumu test edilmektedir. if (isEven(val) == true) ile tamamen eşdeğerdir. Benzer biçimde if (!isEven(val)) ile if (isEven(val) == false) tamamen eşdeğerdir

Örnek Programlar:

1. Klavyeden girilen bir sayı [1, 8] (kapalı) aralığında ise sayının değerini değilse " Sayı [1, 8] aralığında değil " mesajını ekrana yazdıran program:

```
package csd;

class App {
	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);
		System.out.println("Bir sayı giriniz");
		int val = Integer.parseInt(kb.nextLine());

		if (val >= 1 && val <= 8)
			System.out.println(val);
		else
			System.out.println("Sayı [1, 8] aralığında değil");

		kb.close();
	}
}
```

2. Katsayıları klavyeden alınan ikinci dereceden denklemin köklerini bulan program:

```
package csd;
```

```
class App {
	public static void main(String[] args)
	{
		java.util.Scanner kb = new java.util.Scanner(System.in);

		System.out.print("a?");
		double a = Double.parseDouble(kb.nextLine());

		System.out.print("b?");
		double b = Double.parseDouble(kb.nextLine());

		System.out.print("c?");
		double c = Double.parseDouble(kb.nextLine());

		double delta = b * b - 4 * a * c;

		if (delta > 0) {
			double sqrtDelta = Math.sqrt(delta);
			double x1 = (-b + sqrtDelta) / (2 * a);
			double x2 = (-b - sqrtDelta) / (2 * a);

			System.out.printf("x1 = %f\n", x1);
			System.out.printf("x2 = %f\n", x2);
		}
		else if (delta == 0) {
			System.out.println("Çakışık Kökler");
			double x = -b / (2 * a);
			System.out.printf("x = %f\n", x);
		}
		else
			System.out.println("Reel Kök Yok");

		kb.close();
	}
}
```