

# MTK467 Nesneye Yönelik Programlama

---

Ders 3

Zümra Kavafoğlu  
*<https://zumrakavafoglu.github.io/>*

# Bağıntısal Operatörler

---

operatör	tanımı
<	küçüktür
>	büyüktür
<=	küçük eşittir
>=	büyük eşittir
=	eşittir
!=	eşit değildir

# Bağıntısal Operatörler

---

- Bağıntısal ifadelerin değeri boolean tipinde yani true ya da false'dur.

$3 < 4$  → true

$2 > 3$  → false

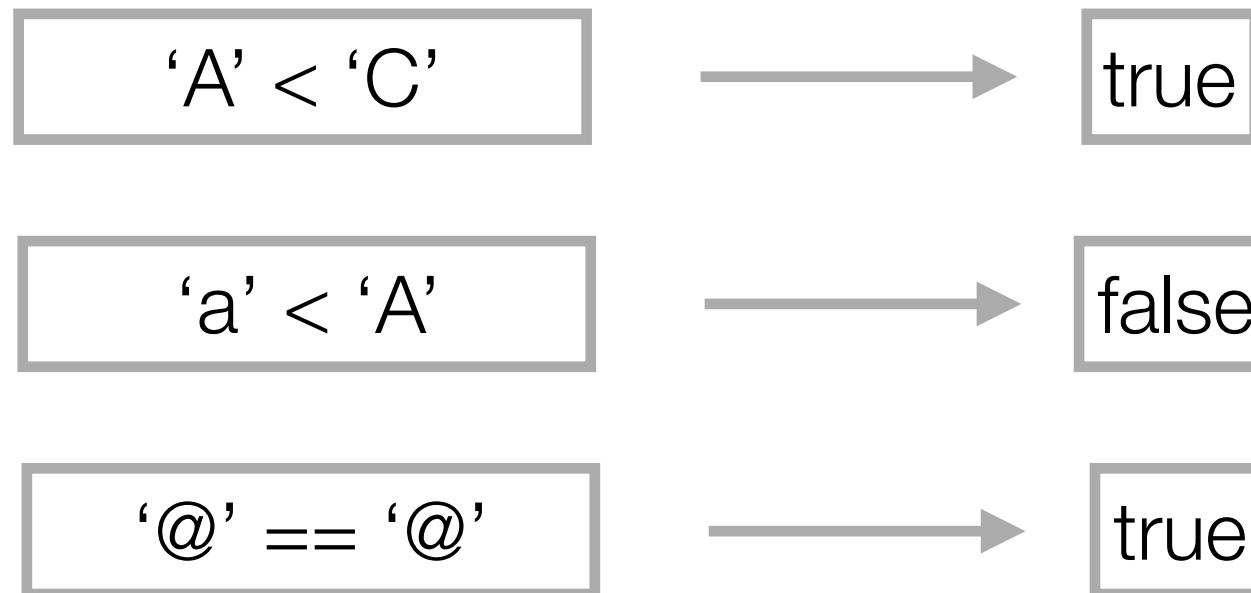
$4 == 4$  → true

- double ve float için == işleci her zaman beklenen sonucu vermeyebilir

# Bağıntısal Operatörler : karakter karşılaştırması

---

- char tipinde değişkenler de bağıntısal operatörlerle karşılaştırılabilirler. Bunun için bu değişkenlerin decimal kodları baz alınır.



# Karakterlerin decimal kodları

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	\$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	<b>DEL</b>

# Mantıksal Operatörler

---

operatör	tanımı
<b>&amp;&amp;</b>	ve
<b>&amp;</b>	ve
<b>  </b>	veya
<b> </b>	veya
<b>!</b>	değil
<b>^</b>	exclusive veya

# Mantıksal Operatörler için Doğruluk Tablosu

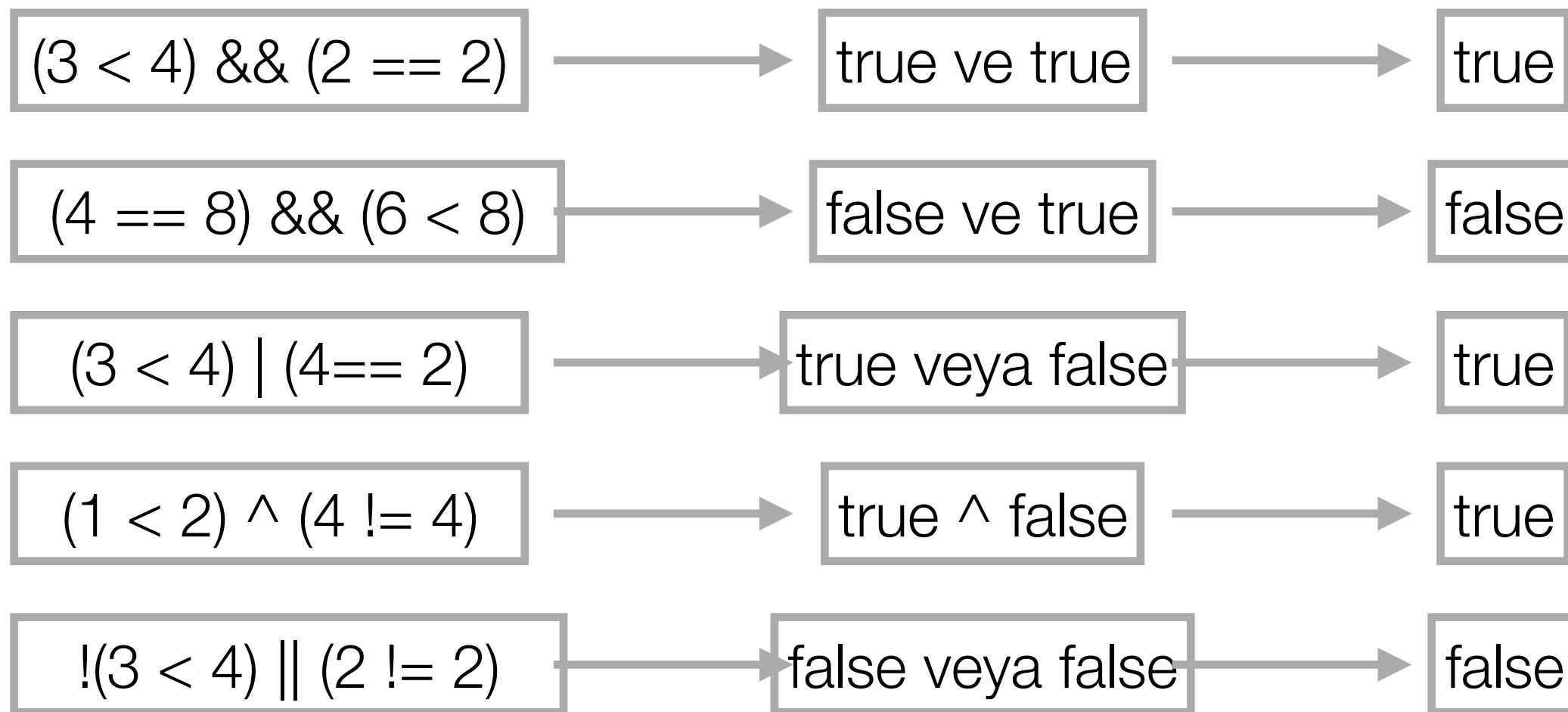
---

p	q	$p \& q$ (p&q)	$p \mid q$ (p q)	$\neg p$	$p \wedge q$
<b>TRUE</b>	TRUE	TRUE	TRUE	FALSE	FALSE
<b>FALSE</b>	TRUE	FALSE	TRUE	TRUE	TRUE
<b>TRUE</b>	FALSE	FALSE	TRUE	FALSE	TRUE
<b>FALSE</b>	FALSE	FALSE	FALSE	TRUE	FALSE

# Mantıksal Operatörler

---

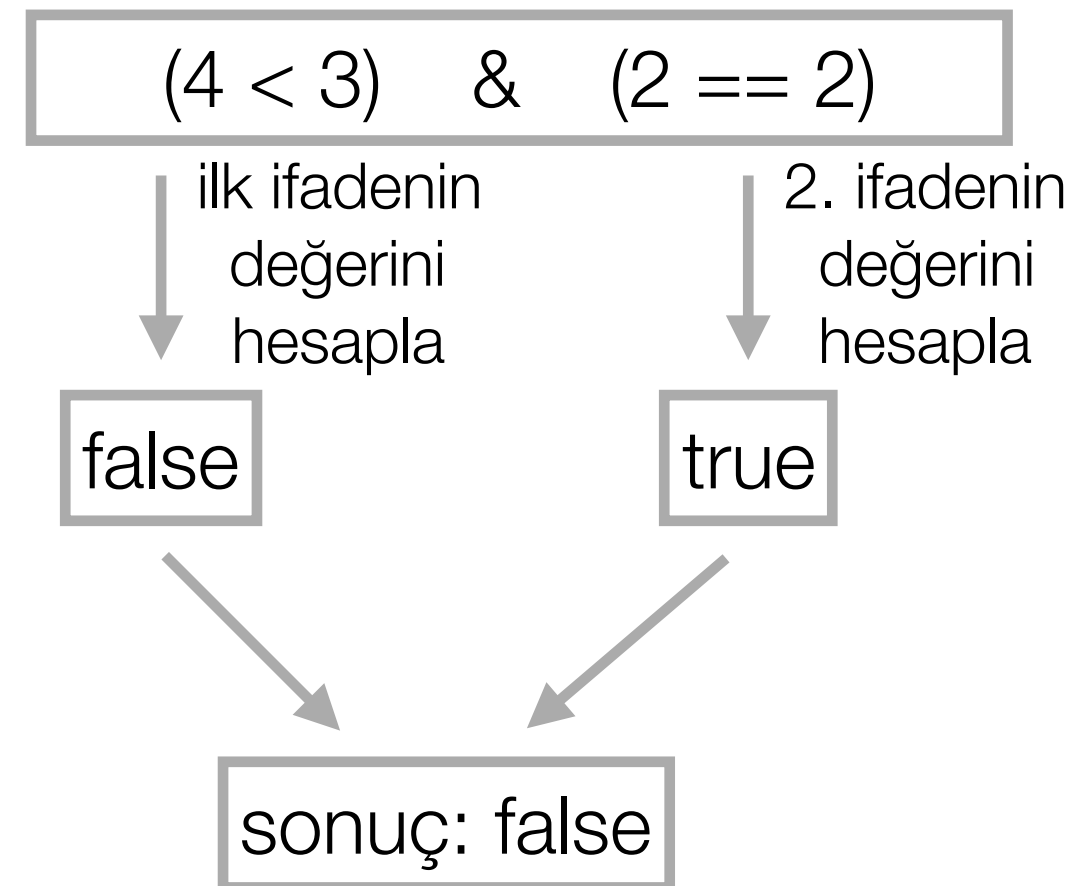
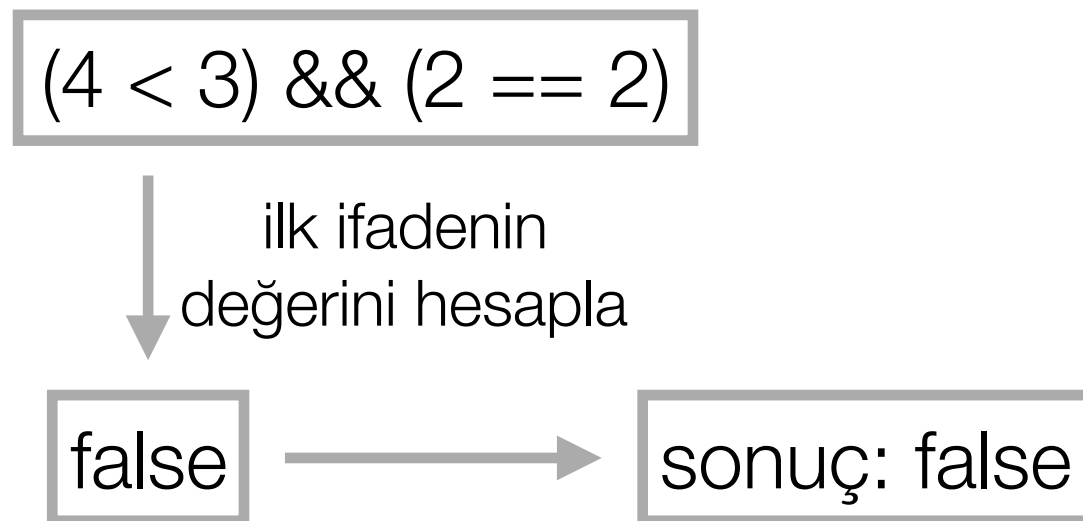
- Exclusive veya  $\wedge$
- ifade1  $\wedge$  ifade2
  - ifade1 ve ifade2 **aynı** değere sahipse **false**
  - ifade1 ve ifade2 **farklı** değerlere sahipse **true**





# Mantıksal Operatörler: && ile & arasındaki fark

- && kullanıldığında, ilk ifade false ise ikinci ifadenin değeri hiç hesaplanmadan sonuç false olarak bulunur.
- & kullanılırsa, ilk ifade false olsa bile iki ifadenin de değeri hesaplanır.



# Operatör öncelik tablosu

TABLE 2.10 Operator Precedence Chart	
<i>Precedence</i>	<i>Operator</i>
Highest Order	casting
	++ and -- (prefix)
	! (not)
	*, /, %
	+, -
	<, <=, >, >=
	==, !=
	& (Unconditional AND)
	^ (Exclusive OR)
	(Unconditional OR)
	&&
Lowest Order	=, +=, -=, *=, /=, %=

# if koşul ifadesi

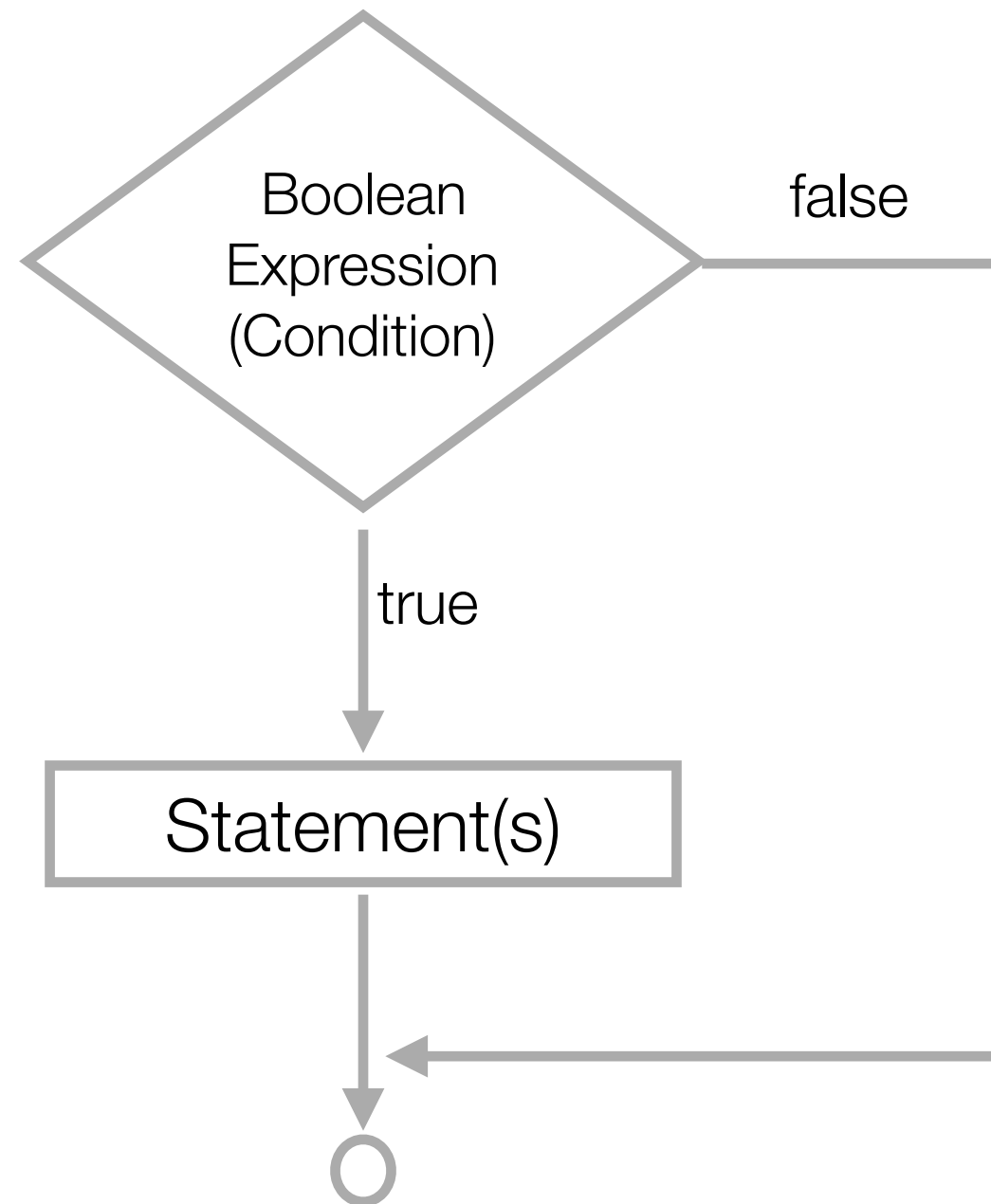
---

```
if(koşul) {  
    koşul doğruysa çalıştırılacak ifade;  
}
```

Burada koşul boolean değerli yani değeri true veya false olan bir ifade olmalıdır.

# if koşul ifadesi

---



## if koşul ifadesi : Örnek Program

---

**Problem:** Kullanıcıdan öğrencinin notunu isteyen ve geçme notunu geçtiyse sınavdan geçtiğini ekrana yazdıran bir program yazınız.

# if koşul ifadesi : Örnek Program

---

```
1  import java.util.Scanner;
2
3  ▶ public class StudentPassExam {
4
5  ▶      public static void main(String[] args) {
6
7          final double passingGrade = 55;
8
9          double grade;
10
11         Scanner input = new Scanner(System.in);
12
13         System.out.println("Enter student's grade: ");
14
15         grade = input.nextDouble();
16
17         if(grade >= passingGrade) {
18             System.out.println("Student passed the exam");
19         }
20     }
21 }
```

# if koşul ifadesi : Örnek Program

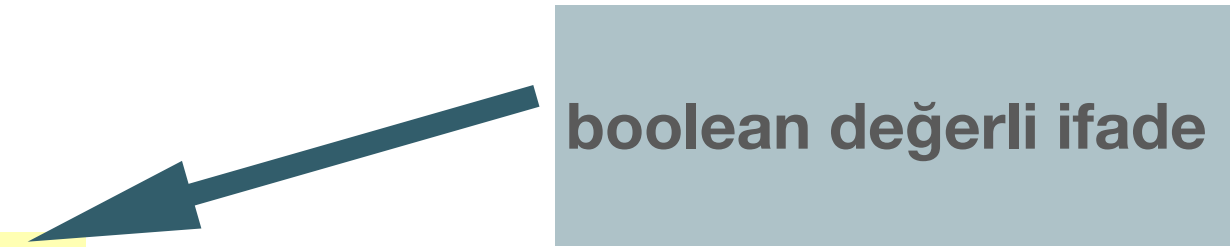
```
1  import java.util.Scanner;
2
3  ▶ public class StudentPassExam {
4
5  ▶     public static void main(String[] args) {
6
7         final double passingGrade = 55;
8
9         double grade;
10
11         Scanner input = new Scanner(System.in);
12
13         System.out.println("Enter student grade:");
14
15         grade = input.nextDouble();
16
17         if(grade >= passingGrade) {
18             System.out.println("Student passed the exam");
19         }
20     }
21 }
```

grade değeri, passingGrade'den  
büyükse konsola  
*Student passed the exam*  
yazdır.

# if koşul ifadesi : Yazım kuralları

---

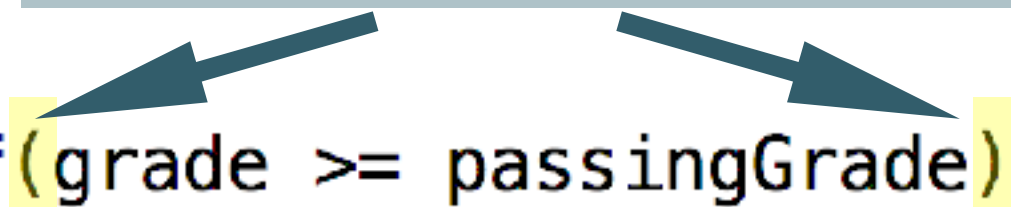
```
if(grade >= passingGrade) {  
    System.out.println("Student passed the exam");  
}
```



boolean değerli ifade

boolean değerli ifade her zaman  
parantezlerin arasına yazılmalı

```
if(grade >= passingGrade) {  
    System.out.println("Student passed the exam");  
}
```





## if koşul ifadesi : Parantezler

---

if bloğunun içindeki ifade tek satırdan oluşuyorsa süslü parantezler yazılmayabilir.

```
if(grade >= passingGrade) {  
    System.out.println("Student passed the exam");  
}
```

=

```
if(grade >= passingGrade)  
    System.out.println("Student passed the exam");
```

## if koşul ifadesi : Parantezler

---

if bloğunun içindeki birden fazla satırdan oluşuyorsa blok süslü parantezle açılıp kapatılmalıdır.

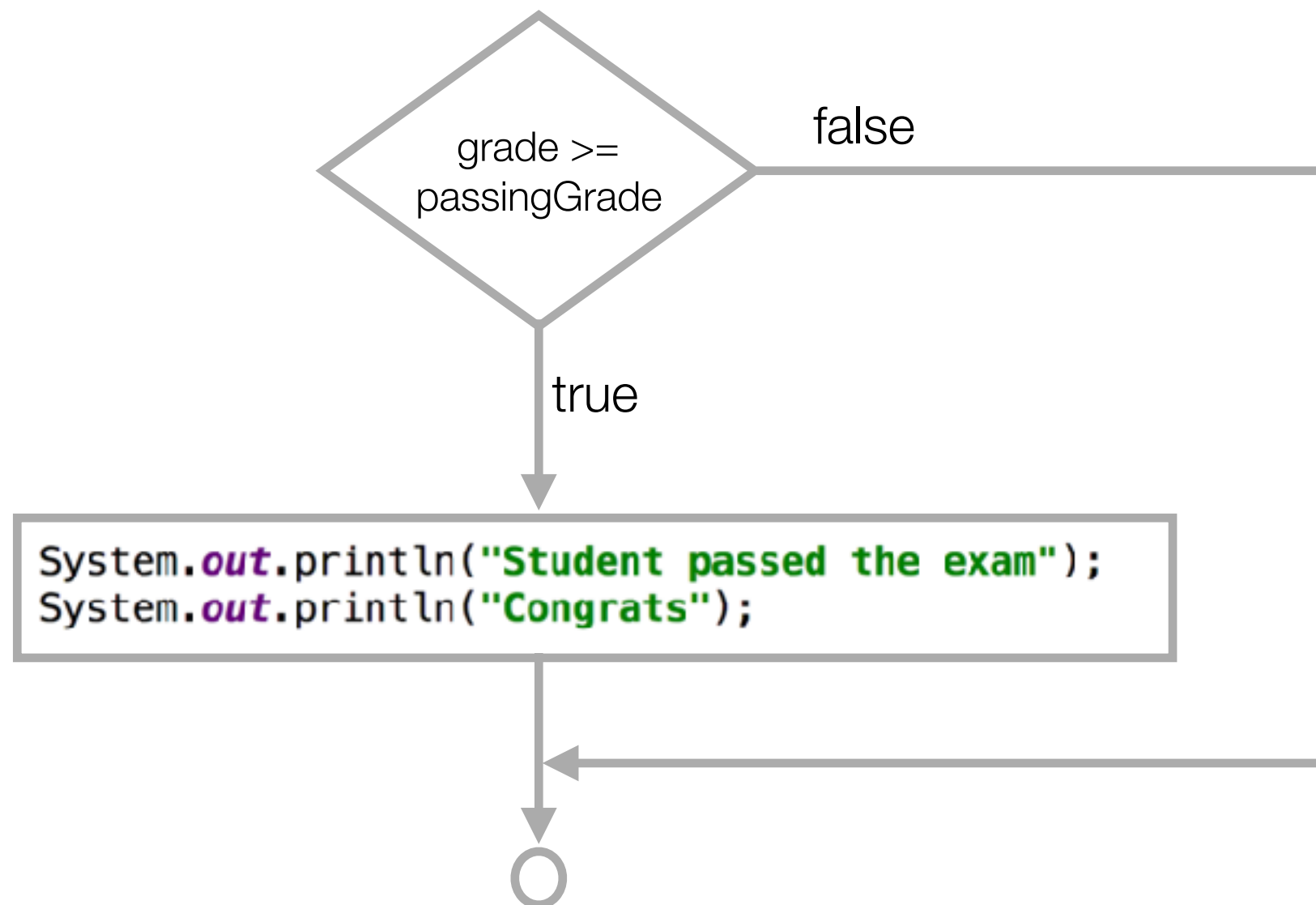
```
if(grade >= passingGrade) {  
    System.out.println("Student passed the exam");  
    System.out.println("Congrats");  
}
```



```
if(grade >= passingGrade)  
    System.out.println("Student passed the exam");  
    System.out.println("Congrats");
```

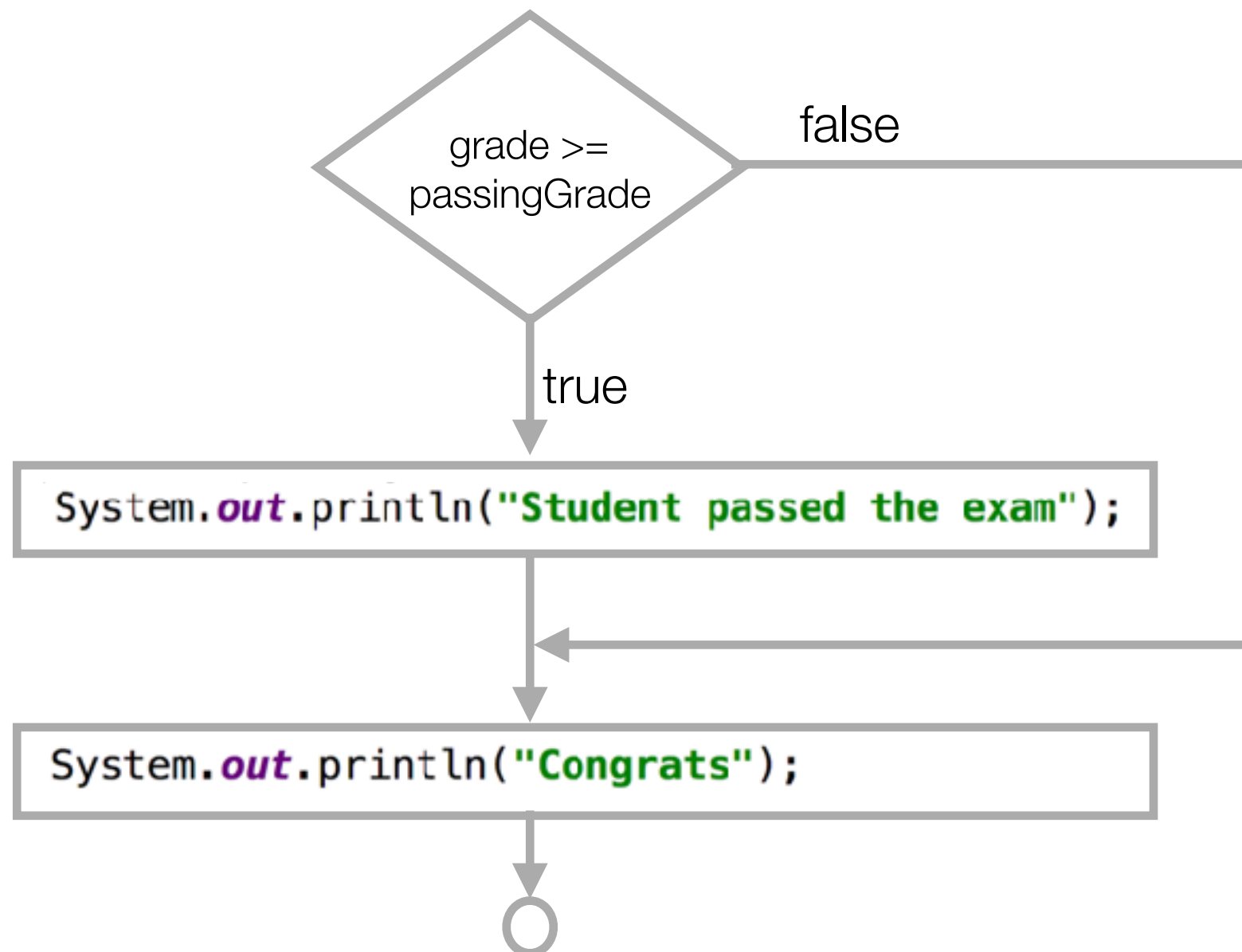
# if koşul ifadesi: Parantezler

```
if(grade >= passingGrade) {  
    System.out.println("Student passed the exam");  
    System.out.println("Congrats");  
}
```



# if koşul ifadesi: Parantezler

```
if(grade >= passingGrade)
    System.out.println("Student passed the exam");
    System.out.println("Congrats");
```



# if koşul ifadesi: Parantezler

```
if(grade >= passingGrade)
    System.out.println("Student passed the exam");
    System.out.println("Congrats");
```

```
/Library/Java/JavaVirtualMachines/jdk-9.jdk/
Connected to the target VM, address: '127.0.0.1:63556'
Enter student's grade:
75
/Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/
Connected to the target VM, address: '127.0.0.1:63556'
Enter student's grade:
75
Disconnected from the target VM, address: '127.0.0.1:63556'
Student passed the exam
Congrats

Process finished with exit code 0
|
```

```
/Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/
Connected to the target VM, address: '127.0.0.1:63556'
Enter student's grade:
35
/Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/
Connected to the target VM, address: '127.0.0.1:63556'
Enter student's grade:
35
Disconnected from the target VM, address: '127.0.0.1:63556'
Congrats

Process finished with exit code 0
```

# if- else koşul ifadesi

---

```
if(koşul) {
```

```
    koşulun değeri true ise çalıştırılacak ifade;
```

```
} else{
```

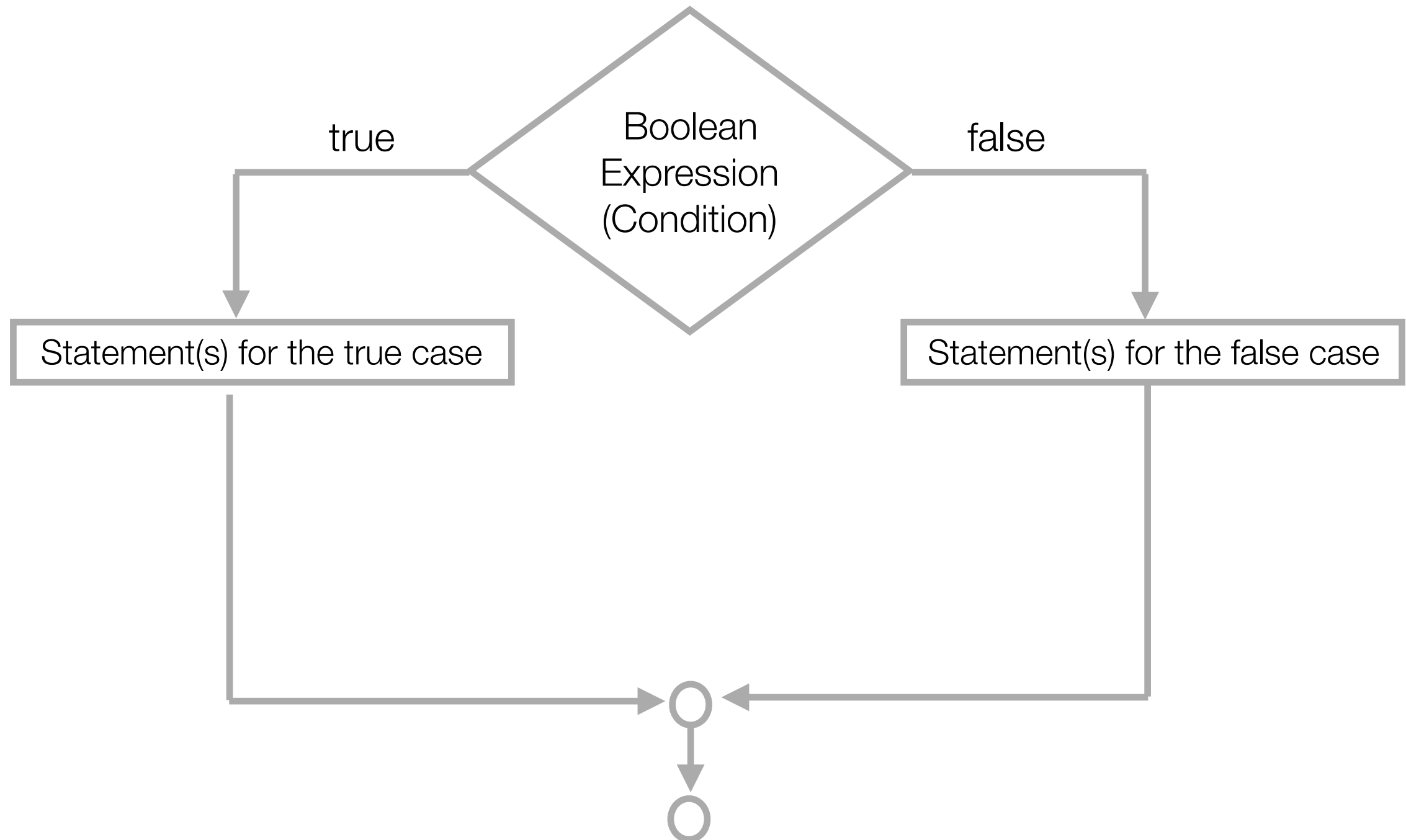
```
    koşulun değeri false ise çalıştırılacak ifade;
```

```
}
```

- Burada koşul boolean değerli yani değeri true veya false olan bir ifade olmalıdır.
- else bloğu için süslü parantezlerin kullanımı if bloğunda olduğu gibidir.

# if- else koşul ifadesi

---



# if- else koşul ifadesi

---

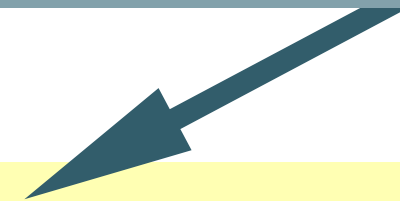
```
1  import java.util.Scanner;
2
3  ▶ public class StudentPassFailExam {
4
5  ▶      public static void main(String[] args) {
6
7          final double passingGrade = 55;
8
9          double grade;
10
11          Scanner input = new Scanner(System.in);
12
13          System.out.println("Enter student's grade: ");
14
15          grade = input.nextDouble();
16
17          if(grade >= passingGrade){
18              System.out.println("Student passed the exam");
19          }else{
20              System.out.println("Student failed the exam");
21          }
22      }
23  }
```



# if- else koşul ifadesi

```
1  import java.util.Scanner;
2
3  ▶ public class StudentPassFailExam {
4
5  ▶     public static void main(String[] args) {
6
7         final double passingGrade = 55;
8
9         double grade;
10
11         Scanner input = new Scanner(System.in);
12
13         System.out.println("Enter student's grade:");
14
15         grade = input.nextDouble();
16
17         if(grade >= passingGrade){
18             System.out.println("Student passed the exam");
19         }else{
20             System.out.println("Student failed the exam");
21         }
22     }
23 }
```

grade değeri, passingGrade'den büyükse  
*Student passed the exam*  
yazdır, değilse;  
*Student failed the exam*  
yazdır.



# İç içe *if-else* yapıları

---

- *else-if* yapısı çok esnek bir kullanım şekline sahiptir. Çünkü istediğimiz sayıda *else-if* yapısını birbirinin içine yerleştirebiliriz. Bu şekilde kullanılan *else-if* yapılarına iç içe *else-if* yapıları denir.
- İç içe *else-if* yapıları çok kullanışlı olmasına rağmen bir çok hataya da neden olabilir. Bunlardan en önemlisi *else* komutunun yanlış *if* komutuyla eşleştirilmesidir.


# İç içe *if* yapıları

---

Bu soruna bir örnek verelim: x ve y isimli iki değişken verilsin ve

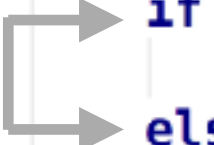
- eğer değişkenlerin ikisi de pozitifse konsola "x ve y pozitiftir"
  - eğer x negatifse konsola "x negatiftir"
- yazılsın

**Doğru ifade**



```
if(x>0) {  
    if (y > 0)  
        System.out.println("x and y are positive");  
}  
else  
    System.out.println("x is negative");
```

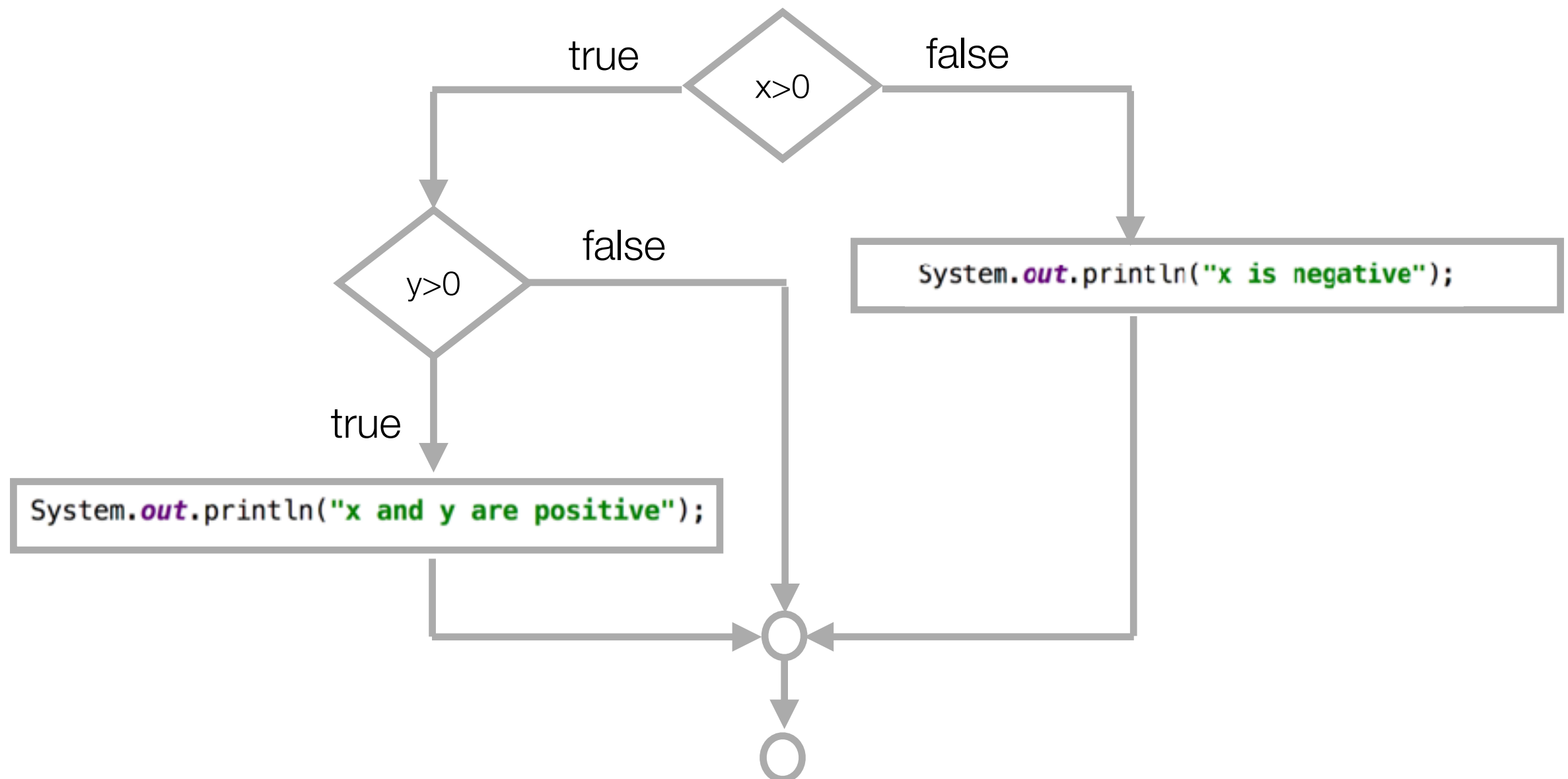
**Yanlış ifade**



```
if(x>0)  
    if (y > 0)  
        System.out.println("x and y are positive");  
else  
    System.out.println("x is negative");
```

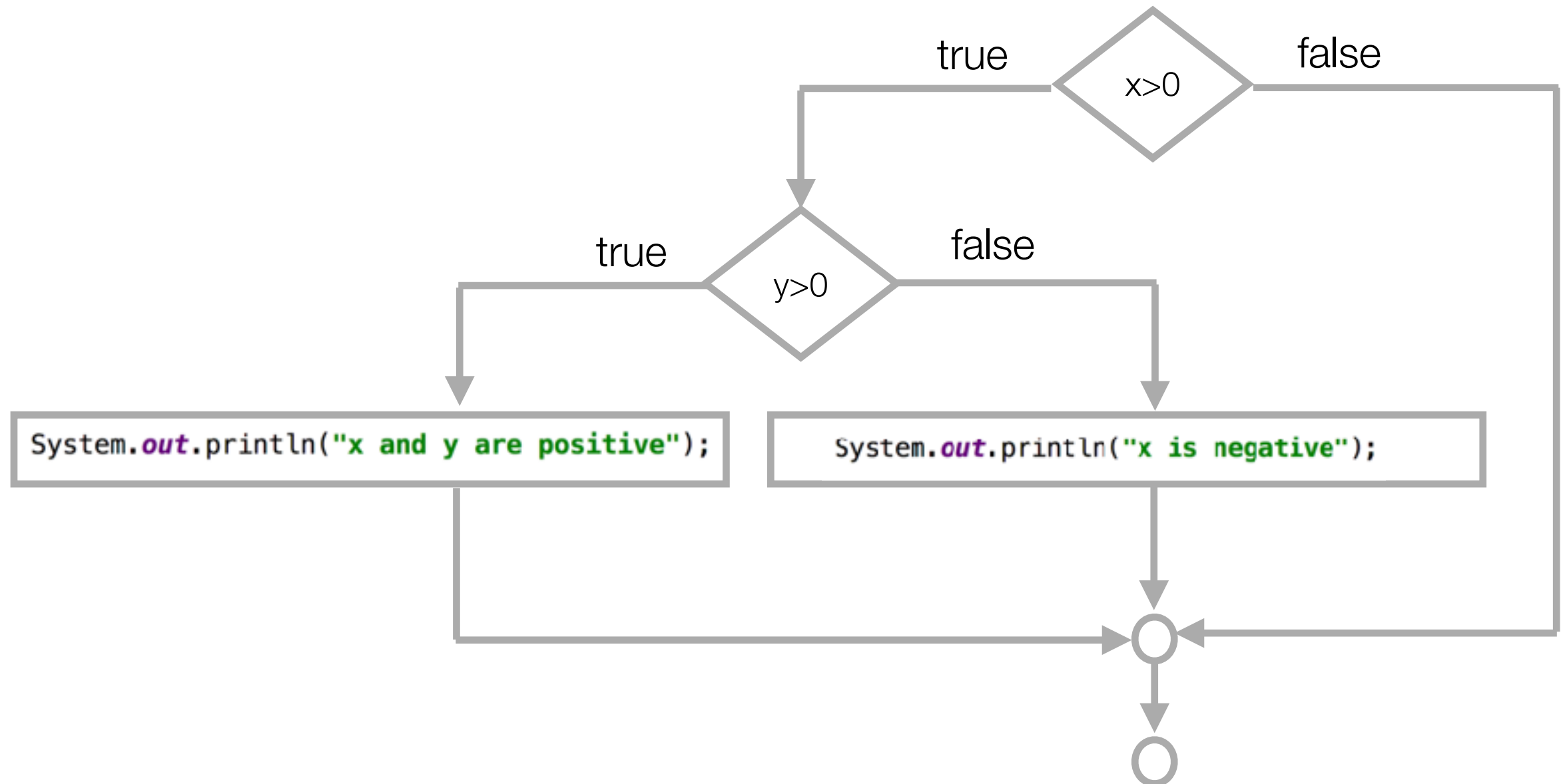
# İç içe if yapıları

```
if(x>0) {  
    if (y > 0)  
        System.out.println("x and y are positive");  
}  
else  
    System.out.println("x is negative");
```



# İç içe if yapıları

```
if(x>0)
    if (y > 0)
        System.out.println("x and y are positive");
    else
        System.out.println("x is negative");
```



# else if komutu

---

```
if (testscore >= 90) {  
    grade = 'A';  
} else if (testscore >= 80) {  
    grade = 'B';  
} else if (testscore >= 70) {  
    grade = 'C';  
} else if (testscore >= 60) {  
    grade = 'D';  
} else {  
    grade = 'F';  
}
```

=

```
if (testscore >= 90) {  
    grade = 'A';  
} else {  
    if (testscore >= 80) {  
        grade = 'B';  
    } else {  
        if (testscore >= 70) {  
            grade = 'C';  
        } else {  
            if (testscore >= 60) {  
                grade = 'D';  
            } else {  
                grade = 'F';  
            }  
        }  
    }  
}
```

# İç içe *if* yapıları

---

- İç içe if/else yapılarında karşılaşılan bir başka hata ise programcının test edilecek koşulların sırasını iyi ayarlayamamasıyla ortaya çıkar.
- Örneğin Celcius cinsinden girilen sıcaklık değerine göre ekranda bir mesaj yazdırmak istiyoruz.
  - `temp > 15` ise "ılık"
  - `temp > 25` ise "sıcak"

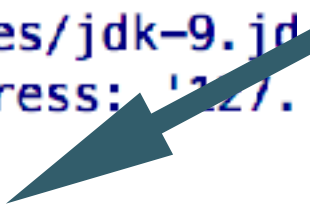
# İç içe *if* yapıları

---

```
if(temperature > 15)
    System.out.println("It's warm!");
else if(temperature > 25)
    System.out.println("It's hot!");
```

Mantık hatası: 15'den büyük her derece için (dolayısıyla 25'den de büyük) "It's warm" mesajını yazdıracak.

```
/Library/Java/JavaVirtualMachines/jdk-9.jd
Connected to the target VM, address: '127.
Enter current temperature:
30
Disconnected from the target VM, address:
It's warm!
```



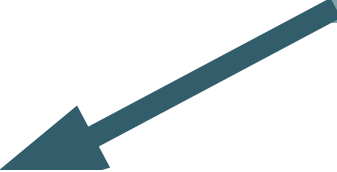


# İç içe *if* yapıları

---

```
if(temperature > 25)
    System.out.println("It's hot!");
else if(temperature > 15)
    System.out.println("It's warm!");
```

Doğru sıralamayla yazıldığında mantık hatası ortadan kalkar.



```
/Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home/bin/java
Connected to the target VM, address: '127.0.0.1:64058', transport:
Enter current temperature:
30
Disconnected from the target VM, address: '127.0.0.1:64058', transp
It's hot!
```

```
Process finished with exit code 0
```

# Koşullu operatör (Conditional Operator)

---

Koşullu operatör (?:) esas olarak sıkıştırılmış bir if/else yapısıdır. Koşullu operatör üçlü bir operatördür , yani koşullu bir ifade yaratmak için üç tane argüman alır.

```
if(grade >= 70){  
    System.out.println("Passed");  
}else{  
    System.out.println("Failed");  
}
```

=

```
System.out.println(grade >= 70 ? "Passed" : "Failed");
```

# Koşullu operatör (Conditional Operator)

---

```
if(x>2){  
    y = 5;  
}else{  
    y = 8;  
}
```

=

```
y = (x > 2 ? 5 : 8);
```

# switch komutu


- Kullanıcıdan haftanın kaçınıcı gününde olduğumuzu girmesin isteyen ve o günün adını yazan bir program yazınız.

```
1  import java.util.Scanner;
2
3  public class SwitchDays {
4
5      public static void main(String[] args) {
6
7          int dayNumber;
8          String dayName = "";
9
10         System.out.print("Enter the day number: ");
11         Scanner input = new Scanner(System.in);
12         dayNumber = input.nextInt();
13
14         if(dayNumber == 1)
15             dayName = "Monday";
16         else if(dayNumber == 2)
17             dayName = "Tuesday";
18         else if(dayNumber == 3)
19             dayName = "Wednesday";
20         else if(dayNumber == 4)
21             dayName = "Thursday";
22         else if(dayNumber == 5)
23             dayName = "Friday";
24         else if(dayNumber == 6)
25             dayName = "Saturday";
26         else if(dayNumber == 7)
27             dayName = "Sunday";
28         else
29             System.out.print("You entered wrong number!");
30
31         System.out.print(dayName);
32     }
33 }
34
```

# switch komutu

- Kullanıcıdan haftanın kaçınıcı gününde olduğumuzu girmesini isteyen ve o günün adını yazan bir program yazınız.

```
1  import java.util.Scanner;
2
3  public class SwitchDays {
4
5      public static void main(String[] args) {
6
7          int dayNumber;
8          String dayName = "";
9
10         System.out.print("Enter the day number: ");
11         Scanner input = new Scanner(System.in);
12         dayNumber = input.nextInt();
13
14         if(dayNumber == 1)
15             dayName = "Monday";
16         else if(dayNumber == 2)
17             dayName = "Tuesday";
18         else if(dayNumber == 3)
19             dayName = "Wednesday";
20         else if(dayNumber == 4)
21             dayName = "Thursday";
22         else if(dayNumber == 5)
23             dayName = "Friday";
24         else if(dayNumber == 6)
25             dayName = "Saturday";
26         else if(dayNumber == 7)
27             dayName = "Sunday";
28         else
29             System.out.print("You entered wrong number!");
30
31         System.out.print(dayName);
32     }
33 }
34
```



```
switch (dayNumber) {
    case 1:
        dayName = "Monday";
        break;
    case 2:
        dayName = "Tuesday";
        break;
    case 3:
        dayName = "Wednesday";
        break;
    case 4:
        dayName = "Thursday";
        break;
    case 5:
        dayName = "Friday";
        break;
    case 6:
        dayName = "Saturday";
        break;
    case 7:
        dayName = "Sunday";
        break;
    default:
        System.out.print("You entered wrong number");
        break;
}
```

# switch komutu

---

```
switch(switch değişkeni){  
    case değer1:  
        ifade(ler)1;  
        break;  
    case değer2:  
        ifade(ler)2;  
        break;  
    .  
    .  
    .  
    case değerN:  
        ifade(ler)N;  
        break;  
    default: default durum için ifade(ler);  
}
```

# switch komutu

```
switch (dayNumber) {  
    case 1:   
        dayName = "Monday";  
        break;  
    case 2:   
        dayName = "Tuesday";  
        break;  
    case 3:   
        dayName = "Wednesday";  
        break;  
    case 4:   
        dayName = "Thursday";  
        break;  
    case 5:   
        dayName = "Friday";  
        break;  
    case 6:   
        dayName = "Saturday";  
        break;  
    case 7:   
        dayName = "Sunday";  
        break;  
    default:   
        System.out.print("You entered wrong number");  
        break;  
}
```

switch değişkeni

değer1

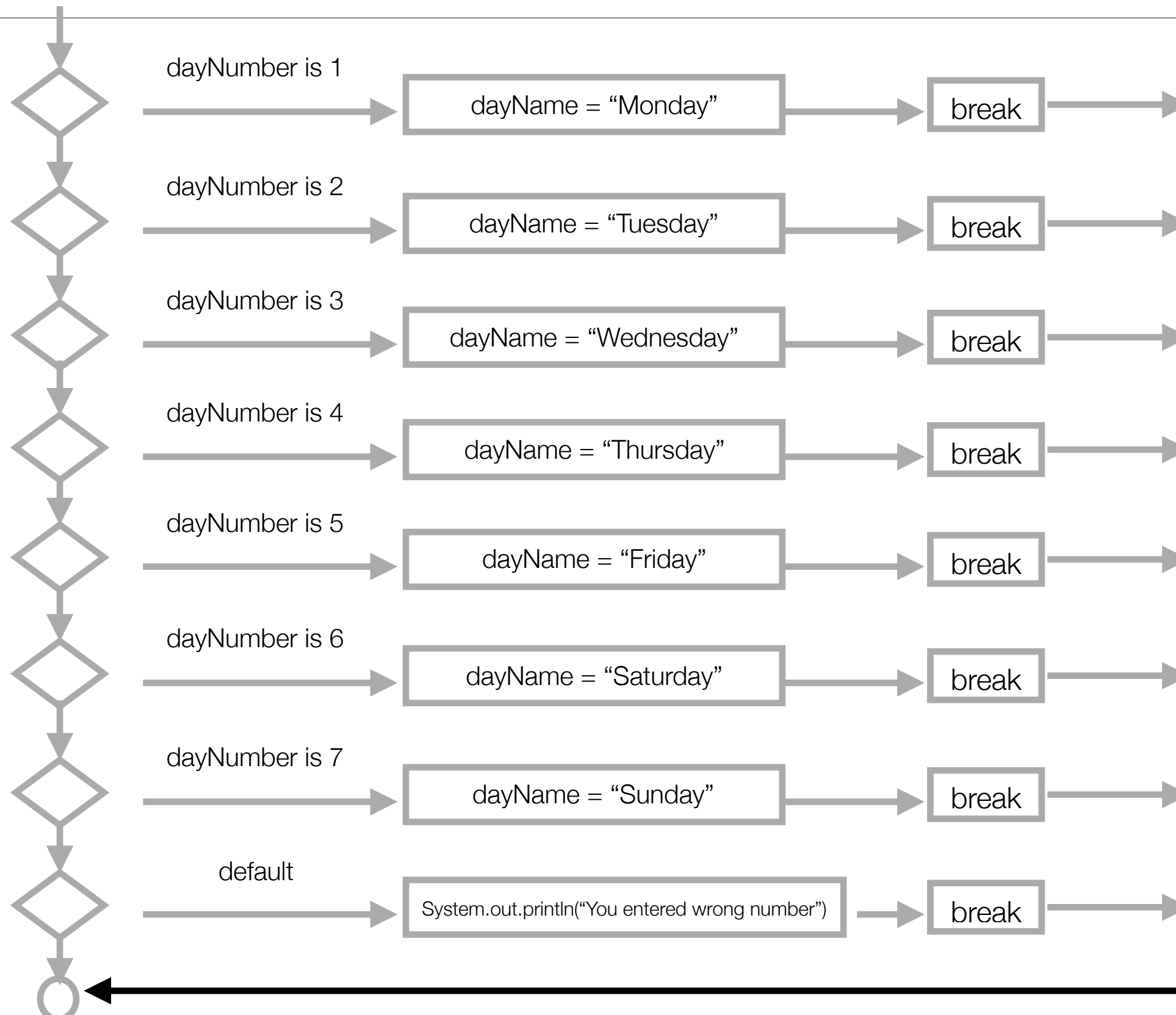
ifade1

değer2

ifade2

default ifade

# switch komutu





# switch komutu

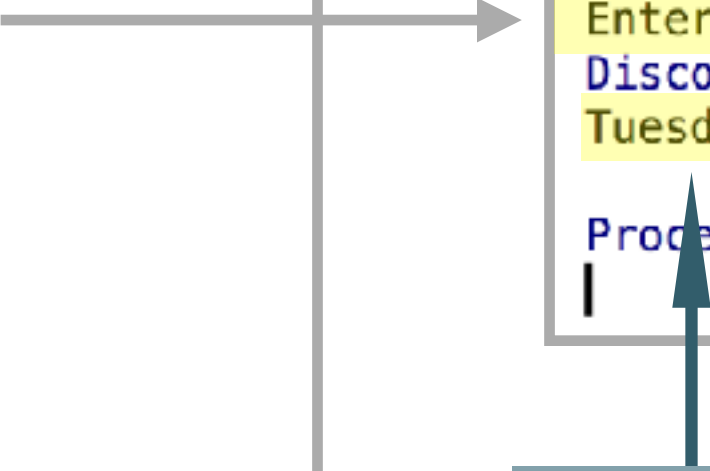
---

- **switch** değişkeni char, byte, short, int ya da String tipinde olmalı ve her zaman parantez içinde olmalı
- değer1 ... değerN switch değişkeni ile aynı tipte olmalıdır.
- **break** ifadesi bütün switch bloğunu sonlandırır. Eğer kullanılmazsa durumun ne olduğuna bakılmaksızın sonraki ifadeler çalıştırılır.
- **default** durumu belirlenen durumlardan hiçbirinin doğru olmadığı durumlar için kullanılır. **default** her zaman switch bloğunun en sonunda yer alır.

# switch komutu: break

Bir durumun sonunda break ifadesi kullanılmazsa synthax hatası oluşmaz ama önemli bir mantık hatası oluşabilir.

```
switch (dayNumber) {  
    case 1:  
        System.out.println("Monday");  
        break;  
    case 2:  
        System.out.println("Tuesday");  
        break;  
    case 3:  
        System.out.println("Wednesday");  
        break;  
    case 4:  
        System.out.println("Thursday");  
        break;  
    case 5:  
        System.out.println("Friday");  
        break;  
    case 6:  
        System.out.println("Saturday");  
        break;  
    case 7:  
        System.out.println("Sunday");  
        break;  
    default:  
        System.out.println("You entered wrong number");  
        break;  
}
```



```
/Library/Java/JavaVirtualMachines/jdk-9.  
Connected to the target VM, address: '12  
Enter the day number: 2  
Disconnected from the target VM, address  
Tuesday  
  
Process finished with exit code 0  
|
```

doğru sonuç

# switch komutu: break

Bir durumun sonunda break ifadesi kullanılmazsa synthax hatası oluşmaz ama önemli bir mantık hatası oluşabilir.

```
switch (dayNumber) {  
    case 1:  
        System.out.println("Monday");  
        break;  
    case 2:  
        System.out.println("Tuesday");  
    case 3:  
        System.out.println("Wednesday");  
        break;  
    case 4:  
        System.out.println("Thursday");  
        break;  
    case 5:  
        System.out.println("Friday");  
        break;  
    case 6:  
        System.out.println("Saturday");  
        break;  
    case 7:  
        System.out.println("Sunday");  
        break;  
    default:  
        System.out.println("You entered wrong number");  
        break;  
}
```

İfadeden sonra break yok

```
/Library/Java/JavaVirtualMachines/  
Connected to the target VM, address  
Enter the day number: 2  
Disconnected from the target VM, address  
Tuesday  
Wednesday  
Process finished with exit code 0
```

hatalı sonuç

# Çalışma zamanı hatası(Run-time error)

- Derleyicinin algılayamadığı ama programın çalışması sırasında ortaya çıkan hatalardır.
- Bazı örnekler:
- **InputMismatchException:** Scanner ile kullanıcıdan istenen verinin tipinde uyumsuzluk olduğunda ortaya çıkar.

```
1  import java.util.Scanner;
2
3  ▶ public class RunTimeErrorDemo {
4  ▶      public static void main(String[] args) {
5
6          int inputValue;
7
8          Scanner input = new Scanner(System.in);
9
10         System.out.print("Enter an integer: ");
11
12         inputValue = input.nextInt();
13     }
14 }
```

```
/Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home/bin/j
Connected to the target VM, address: '127.0.0.1:55277', transpo
Enter an integer: 3.7
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:860)
    at java.base/java.util.Scanner.next(Scanner.java:1497)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2161)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2115)
    at RunTimeErrorDemo.main(RunTimeErrorDemo.java:12)
Disconnected from the target VM, address: '127.0.0.1:55277', tr
Process finished with exit code 1
|
```

# Çalışma zamanı hatası(Run-time error)

---

- **ArithmeticException:** Bir sayının 0 ile bölümünde ortaya çıkar.

```
1
2 ▶ public class ArithmeticExceptionDemo {
3 ▶     public static void main(String[] args) {
4
5         int x = 3/0;
6
7         System.out.println(x);
8     }
9 }
```

```
/Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home/bin/java -a
Connected to the target VM, address: '127.0.0.1:57477', transport: 's
Disconnected from the target VM, address: '127.0.0.1:57477', transpo
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at ArithmeticExceptionDemo.main(ArithmeticExceptionDemo.java:5)
```

```
Process finished with exit code 1
```

# Mantık hatası

- Mantık hataları programa yaptırmak istediğiniz bir görevin yanlış yapılması sonucu ortaya çıkar. Derleme zamanında veya çalışma zamanında herhangi bir hata verilmemesine rağmen programın çıktısı istediğiniz çıktı değildir. Bu istemsiz durumun kodun hangi bölümünden kaynaklandığını, yani mantık hatasının yerini bulmak özellikle geniş kapsamlı programlarda çok zor olabilir.
- Belli başlı mantık hatalarına örnekler:
  - Operatör önceliklerinde hata  
 $5+4*3$  ile  $(5+4)*3$  farklı sonuçlar verir
  - Bir koşulun yanlış olduğu halde doğru olduğunu varsaymak
  - Kayar noktalı sayılarla(floating point numbers, double / float) eşitlik kontrolü yapmak
  - İki tamsayı tipinde değişkenin bölümünün ondalık sayı çıkacağını varsaymak
  - Noktalı virgülü yanlış yere koymak

# float ve double tipinde değişkenlerin eşitliğinin karşılaştırılması

---

- floating-point değerleriyle aritmetik işlemler yapılırken bazı küçük yuvarlamalar sonucu teorik olarak eşit olan iki floating-point değişkeni çok küçük bir değer farkı nedeniyle eşit değilmiş gibi anlaşılabilir. Bu yanlış anlaşılma sonucu doğru olması gereken bir koşul yanlış olarak alınır ve programda ayıklanması çok güç mantık hataları oluşur.
- Dolayısıyla floating-point tipi değişkenlerle program yazarken eşitlik yerine yaklaşık eşitliği kontrol etmek daha iyi bir fikirdir. Örneğin double tipinde bir değişken olan x'in 10.0 'a eşit olup olmadığına bakmaktansa ,  $|x-10.0| \leq 1E-10$  eşitsizliğine bakmak daha mantıklıdır.



# float ve double tipinde değişkenlerin eşitliğinin karşılaştırılması

```
1 ▶ public class FloatingPointEquality {
2
3 ▶     public static void main(String[] args)
4     {
5         final double EPS = 1.0E-14;
6
7         double sinX;
8
9         sinX = Math.sin(2*Math.PI);
10
11         System.out.print("Equality test : ");
12
13         if(sinX==0)
14             System.out.println("sin(2*PI) is equal to zero");
15         else
16             System.out.println("sin(2*PI) is NOT equal to zero");
17
18         System.out.print("Approximate equality test : ");
19
20         if(Math.abs(sinX-0.0) < EPS )
21             System.out.println("sin(2*PI) is equal to zero");
22         else
23             System.out.println("sin(2*PI) is NOT equal to zero");
24
25         System.out.println("\nReal values :");
26         System.out.println("sin(2*PI) = " + sinX);
27
28     }
29 }
```



# float ve double tipinde değişkenlerin eşitliğinin karşılaştırılması

---

/Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home

Equality test :  $\sin(2\pi)$  is NOT equal to zero

Approximate equality test :  $\sin(2\pi)$  is equal to zero

Real values :

$\sin(2\pi) = -2.4492935982947064E-16$

Process finished with exit code 0

# Mantık hatası

- Noktalı virgülü yanlış yere koymak

```
1 ▶ public class IfError {  
2 ▶     public static void main(String args[]) {  
3  
4         int x = 10;  
5  
6         if (x < 0);  
7             System.out.println("x is negative");  
8  
9     }  
10 }
```

x, 0'dan küçükse boş satır çalıştır.

```
/Library/Java/JavaVirtualMachines/jdk-9.  
Connected to the target VM, address: '12  
Disconnected from the target VM, address:  
x is negative
```

```
Process finished with exit code 0
```