Bölüm 3: Seçimler (Selections)

Motivasyon

Daire alanını hesaplayan Java programında, yarıçap için negatif bir değer atarsanız, program geçersiz bir sonuç yazdırır. Yarıçap negatifse, programın alanı hesaplamasını istemezsiniz. Bu durumla nasıl başa çıkabilirsiniz?

Amaçlar

- boolean tipi ve karşılaştırma operatörleri kullanarak Boolean açıklamaları yazma (§3.2).
- Boolean açıklamaları ile AdditionQuiz programı yazma (§3.3).
- Seçim kontrolünü tek-yönlü if ifadeleri kullanarak uygulama (§3.4)
- Tek yönlü if ifadeleri kullanarak GuessBirthday oyunu yazma (§3.5).
- Seçim kontrolünü çift-yönlü if ifadeleri kullanarak uygulama(§3.6).
- Seçim kontrolünü içiçe geçmiş (nested) if ifadeleri kullanarak uygulama(§3.7).
- if ifadelerindeki yaygın hatalardan kaçınma (§3.8).
- Çeşitli örnekler (BMI, ComputeTax, SubtractionQuiz) (§3.9-3.11).
- Mantıksal (logical) operatörleri kullanarak koşulları birleştirme(&&, ||, and !) (§3.12).
- Seçim ifadelerini birleştirilmiş koşular ile programlama (LeapYear, Lottery) (§§3.13-3.14).
- switch ifadeleri ile seçim kontrolü (§3.15).
- Koşullu operatör kullanma (§3.16).
- System.out.printf metodu ve çıktı biçimleri (§3.17).
- Operatör önceliği ve ilişkilendirme ile ilgili kuralları inceleme (§3.18).

`boolean` veri tipi

Genellikle bir programda, i'nin j'den büyük olup olmadığı gibi iki değeri karşılaştırmanız gerekir. Java, iki değeri karşılaştırmak için kullanılabilecek altı karşılaştırma operatörü (ilişkisel operatör olarak da bilinir) sağlar. Karşılaştırma sonucu bir Boolean değeridir: TRUE veya FALSE.

`boolean` veri tipi

İlkel tip	Sınıf	Uzunluk (byte)	Kapsam
boolean	boolean	1	TRUE FALSE

boolean değişken_adı;

boolean değişken_adı = değişken_değeri;

```
boolean b = (1 > 2);
```

Karşılaştırma Operatörleri

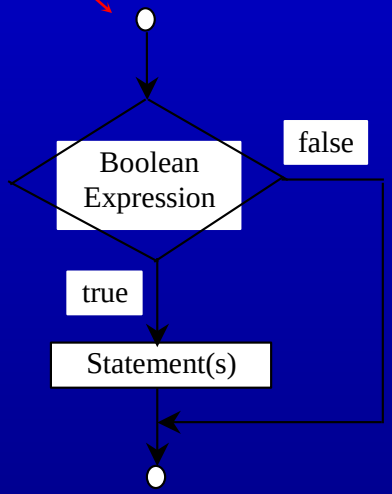
Operatör	*Ad*
<	küçüktür
<=	küçük veya eşit
>	büyüktür
>=	büyük veya eşit
==	eşit
!=	eşit değil

Problem: Basit Matematik Öğrenme Aracı

Bu örnekte 1. sınıf öğrencilerine toplama işlemi egzersizleri yaptıran bir program oluşturmak isteniyor. Program rastgele tek basamaklı sayı1 ve sayı2 gibi iki tane tam sayıları üretir ve öğrenciye "7 + 9 nedir?" gibi bu iki sayının toplamını sorar. Öğrenci cevabı yazdıktan sonra, program cevabının doğru veya yanlış olduğunu belirten bir mesaj verir.

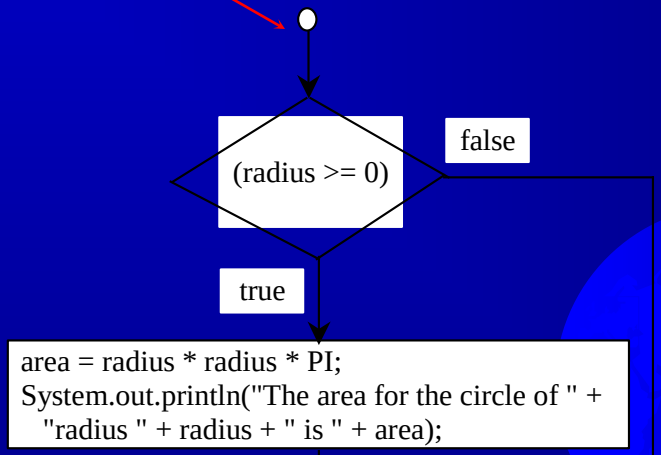
Tek-yönlü `if` ifadeleri

```
if (boolean-expression) {
  statement(s);
}
```



(A)

```
if (radius >= 0) {
  area = radius * radius * PI;
  System.out.println("The area"
    + " for the circle of radius "
    + radius + " is " + area);
}
```



(B)

Not

```
if i > 0 {
  System.out.println("i is positive");
}
```

(a) Yanlış

```
if (i > 0) {
  System.out.println("i is positive");
}
```

(b) Doğru

```
if (i > 0) {
  System.out.println("i is positive");
}
```

(a)

Denktir

```
if (i > 0)
  System.out.println("i is positive");
```

(b)

Basit if örneği

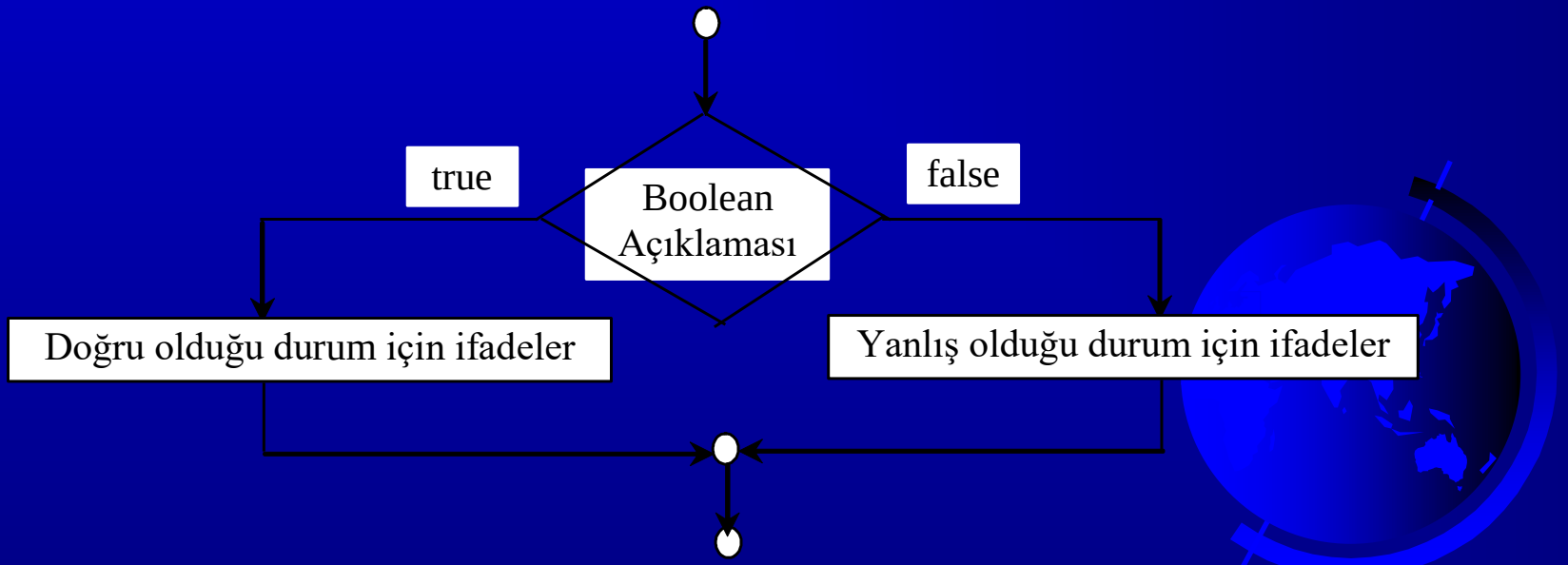
Kullanıcıdan bir tamsayı girmesini isteyen bir program yazın. Girilen sayı 5'in katıysa, HiFive , 2 ile bölünebiliyorsa HiEven yazdırın.

SimpleIfDemo

Run

İki-yönlü `if` ifadesi

```
if (boolean-expression) {
  statement(s)-for-the-true-case;
}
else {
  statement(s)-for-the-false-case;
}
```



if...else Örneği

```
if (radius >= 0) {
  area = radius * radius * 3.14159;

  System.out.println("The area for the "
    + "circle of radius " + radius +
    " is " + area);
}
else {
  System.out.println("Negative input");
}
```

İçiçe geçmiş (nested) if ifadeleri

```
if (score >= 90.0)
  grade = 'A';
else
  if (score >= 80.0)
    grade = 'B';
  else
    if (score >= 70.0)
      grade = 'C';
    else
      if (score >= 60.0)
        grade = 'D';
      else
        grade = 'F';
```

Equivalent

```
if (score >= 90.0)
  grade = 'A';
else if (score >= 80.0)
  grade = 'B';
else if (score >= 70.0)
  grade = 'C';
else if (score >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

animation

if-else ifadelerinin yürütülmesi

Skorun 70.0 olduğunu varsayalım

Koşul FALSE olur

```
if (score >= 90.0)
  grade = 'A';
else if (score >= 80.0)
  grade = 'B';
else if (score >= 70.0)
  grade = 'C';
else if (score >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

animation

if-else ifadelerinin yürütülmesi

Skorun 70.0 olduğunu varsayalım

Koşul FALSE olur

```
if (score >= 90.0)
  grade = 'A';
else if (score >= 80.0)
  grade = 'B';
else if (score >= 70.0)
  grade = 'C';
else if (score >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

animation

if-else ifadelerinin yürütülmesi

Skorun 70.0 olduğunu varsayalım

Koşul TRUE olur

```
if (score >= 90.0)
  grade = 'A';
else if (score >= 80.0)
  grade = 'B';
else if (score >= 70.0)
  grade = 'C';
else if (score >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

animation

if-else ifadelerinin yürütülmesi

Skorun 70.0 olduğunu varsayalım

Harf notu C olur

```
if (score >= 90.0)
  grade = 'A';
else if (score >= 80.0)
  grade = 'B';
else if (score >= 70.0)
  grade = 'C';
else if (score >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

animation

if-else ifadelerinin yürütülmesi

Skorun 70.0 olduğunu varsayalım

If ifadesinden çıkılır

```
if (score >= 90.0)
  grade = 'A';
else if (score >= 80.0)
  grade = 'B';
else if (score >= 70.0)
  grade = 'C';
else if (score >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

Not

Else cümlesi aynı bloktaki en son if cümlesiyle eşleşir.

```
int i = 1;
int j = 2;
int k = 3;

if (i > j)
  if (i > k)
    System.out.println("A");
else
    System.out.println("B");
```

(a)

Equivalent

```
int i = 1;
int j = 2;
int k = 3;

if (i > j)
  if (i > k)
    System.out.println("A");
  else
    System.out.println("B");
```

(b)

Not

Yukarıdaki ifade çıktı vermez. Else yan tümcesini ilk yan tümce ile eşleşmeye zorlamak için, bir çift parantez eklemek gerekir:

```
int i = 1;
int j = 2;
int k = 3;
if (i > j) {
  if (i > k)
    System.out.println("A");
}
else
  System.out.println("B");
```

Çıktı: B.

Yaygın hatalar

Yaygın hata: if cümlesi sonuna ; koymak

```
if (radius >= 0);
{
  area = radius*radius*PI;
  System.out.println(
    "The area for the circle of radius " +
    radius + " is " + area);
}
```

Wrong

Bu hatayı bulması zor çünkü derleme veya runtime hatası değildir. Bu bir mantık (logic) hatasıdır.

İpucu

```
if (number % 2 == 0)
  even = true;
else
  even = false;
```

(a)

Denktir

```
boolean even
  = number % 2 == 0;
```

(b)

Dikkat

```
if (even == true)
  System.out.println(
    "It is even.");
```

(a)

Denktir

```
if (even)
  System.out.println(
    "It is even.");
```

(b)

Problem: Geliştirilmiş Matematik Öğrenme Aracı

Bu örnek, birinci sınıf çocuğuna işlemini öğreten bir program oluşturmak içindir. Program rastgele olarak sayı1 ve sayı2 ile sayı1> sayı2 olan iki tek basamaklı tam sayı üretir ve öğrenciye "9 - 2 nedir?" gibi bir soru görüntüler. Öğrenci, giriş iletişim kutusuna cevap yazdıktan sonra, program cevapların doğru olup olmadığını belirten bir mesaj iletişim kutusu görüntüler.

SubtractionQuiz

Run

Problem: Vücut Kitle İndeksi

Vücut Kitle İndeksi (Body Mass Index (BMI)) kiloya göre sağlığın bir ölçüsüdür. Kilonuzu kilogram alarak ve boyunuzun metre karesine bölerek hesaplanabilir. BMI'nin 16 yaş ve üstü insanlar için yorumu aşağıdaki gibidir:

BMI	Yorum
16 altı	çok zayıf
16-18	zayıf
18-24	normal
24-29	biraz kilolu
29-35	aşırı kilolu
35 üzeri	obez

Problem: Vergi Hesaplama

ABD federal kişisel gelir vergisi, dosyalama durumuna ve vergilendirilebilir gelire göre hesaplanır. Dört dosyalama durumu vardır: tekli başvurular, ortak evli başvurular, ayrı evli başvurular ve hane reisi. 2009 yılı vergi oranları aşağıda gösterilmiştir.

Marginal Tax Rate	Single	Married Filing Jointly or Qualified Widow(er)	Married Filing Separately	Head of Household
10%	$0 – $8,350	$0 – $16,700	$0 – $8,350	$0 – $11,950
15%	$8,351– $33,950	$16,701 – $67,900	$8,351 – $33,950	$11,951 – $45,500
25%	$33,951 – $82,250	$67,901 – $137,050	$33,951 – $68,525	$45,501 – $117,450
28%	$82,251 – $171,550	$137,051 – $208,850	$68,525 – $104,425	$117,451 – $190,200
33%	$171,551 – $372,950	$208,851 – $372,950	$104,426 – $186,475	$190,201 - $372,950
35%	$372,951+	$372,951+	$186,476+	$372,951+

Problem: Vergi Hesaplama, devam

```
if (status == 0) {
  // Compute tax for single filers
}
else if (status == 1) {
  // Compute tax for married file jointly
}
else if (status == 2) {
  // Compute tax for married file separately
}
else if (status == 3) {
  // Compute tax for head of household
}
else {
  // Display wrong status
}
```

ComputeTax

Run

Mantıksal Operatörler (Logical Operators)

Operatör	*Ad*
!	not
&&	and
||	or
^	exclusive or

! Operatörünün doğruluk tablosu

p	!p
true	false
false	true

Örnek (age = 24, gender = 'M' olsun)

!(age > 18) is false, because (age > 18) is true.

!(gender != 'F') is false, because (gender != 'F') is true.

p	!p
true	false
false	true

Example

!(1 > 2) is true, because (1 > 2) is false.

!(1 > 0) is false, because (1 > 0) is true.

&& Operatörünün Doğruluk Tablosu

p1	p2	p1 && p2
false	false	false
false	true	false
true	false	false
true	true	true

Example (assume age = 24, gender = 'F')

(age > 18) && (gender == 'F') is true, because (age > 18) and (gender == 'F') are both true.

(age > 18) && (gender != 'F') is false, because (gender != 'F') is false.

p1	p2	p1 && p2
false	false	false
false	true	false
true	false	false
true	true	true

Example

(3 > 2) && (5 >= 5) is true, because (3 > 2) and (5 >= 5) are both true.

(3 > 2) && (5 > 5) is false, because (5 > 5) is false.

|| Operatörünün Doğruluk Tablosu

p1	p2	p1 || p2
false	false	false
false	true	true
true	false	true
true	true	true

Example (assume age = 24, gender = 'F')

(age > 34) || (gender == 'F') is true, because (gender == 'F') is true.

(age > 34) || (gender == 'M') is false, because (age > 34) and (gender == 'M') are both false.

p1	p2	p1 || p2
false	false	false
false	true	true
true	false	true
true	true	true

Example

(2 > 3) || (5 > 5) is false, because (2 > 3) and (5 > 5) are both false.

(3 > 2) || (5 > 5) is true, because (3 > 2) is true.

^ Operatörünün Doğruluk Tablosu

p1	p2	p1 ^ p2
false	false	false
false	true	true
true	false	true
true	true	false

Example (assume age = 24, gender = 'F')

(age > 34) ^ (gender == 'F') is true, because (age > 34) is false but (gender == 'F') is true.

(age > 34) || (gender == 'M') is false, because (age > 34) and (gender == 'M') are both false.

Örnek

Here is a program that checks whether a number is divisible by 2 and 3, whether a number is divisible by 2 or 3, and whether a number is divisible by 2 or 3 but not both:
Bir sayının 2 ve 3 ile bölünebilir olup olmadığını, bir sayının 2 veya 3 ile bölünebilir olup olmadığını ve bir sayının 2 veya 3 ile bölünebilir fakat ikisine birden bölünemeyen olup olmadığını denetleyen bir program:

TestBooleanOperators

Run

Örnek

```
System.out.println("Is " + number + " divisible by 2 and 3? " +
  ((number % 2 == 0) && (number % 3 == 0)));

System.out.println("Is " + number + " divisible by 2 or 3? " +
  ((number % 2 == 0) || (number % 3 == 0)));

 System.out.println("Is " + number +
   " divisible by 2 or 3, but not both? " +
   ((number % 2 == 0) ^ (number % 3 == 0)));
```

TestBooleanOperators

Run

Companion Website

& ve | Operatörleri

```
Eğer x, 1 ise bu açıklamadan sonra x
  ne olur?
(x > 1) & (x++ < 10)

Eğer x, 1 ise bu açıklamadan sonra x
  ne olur?
(1 > x) && ( 1 > x++)

(1 == x) | (10 > x++)?
(1 == x) || (10 > x++)?
```

Problem:Artık Yıl mı?

Bu program ilk önce kullanıcıdan int değeri olarak bir yıl girmesini ister ve artık yıl olup olmadığını kontrol eder.

Yıl, 4 ile bölünebilir ancak 100 ile bölünemezse veya 400 ile bölünebilirse artık bir yıldır.

```
(year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)
```

LeapYear

Run

Problem: Piyango

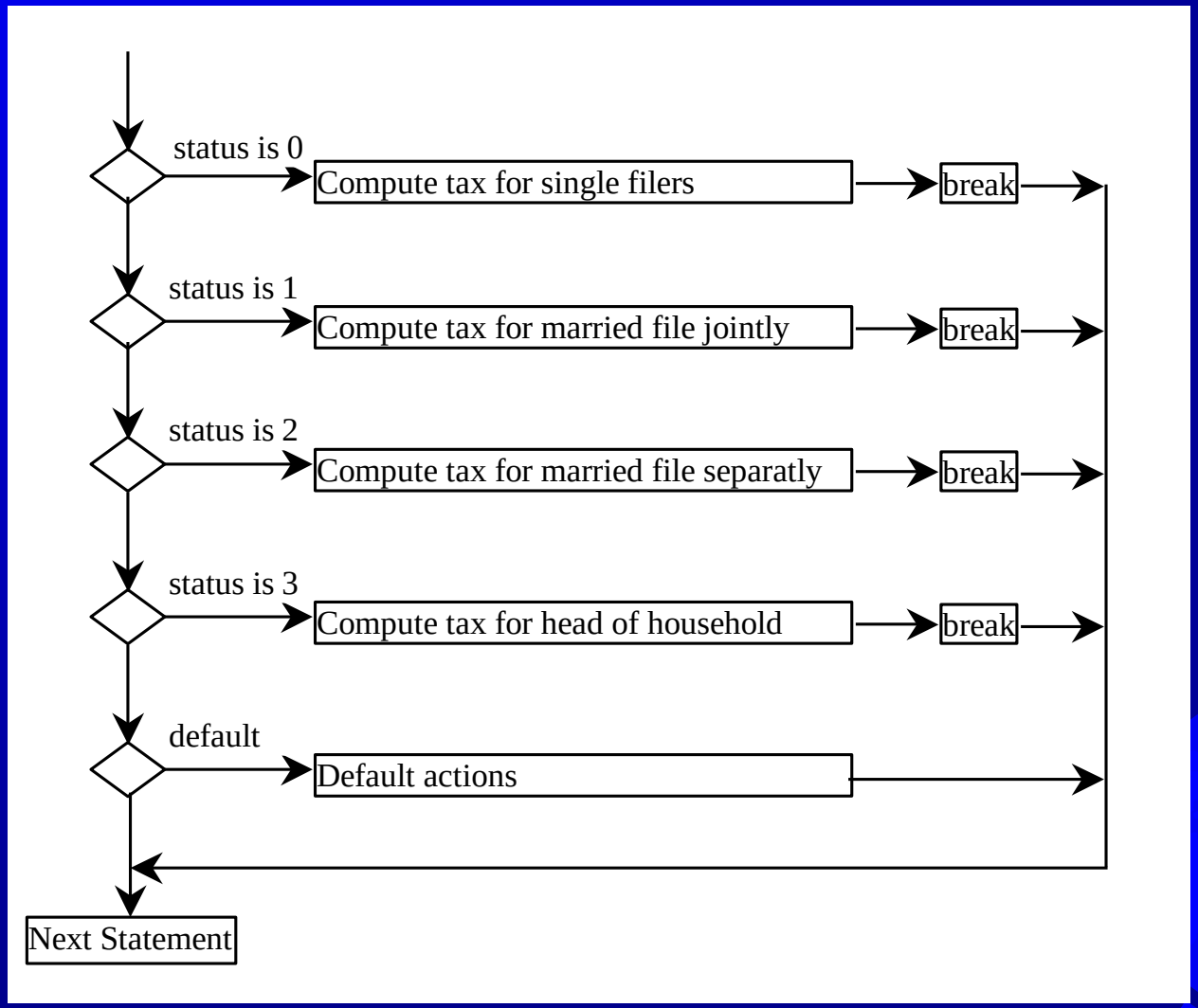
Rasgele, iki basamaklı bir sayıdan bir piyango oluşturan, kullanıcıdan iki basamaklı bir sayı girmesini isteyen ve kullanıcının aşağıdaki kurala göre kazanıp kazanmayacağını belirleyen bir program yazın:

- Eğer kullanıcı girişi çekilişe tam olarak uyuyorsa, ödül 10.000 $ 'dır.
- Eğer kullanıcı girişi piyango ile eşleşirse, ödül 3.000 $ 'dır.
- Kullanıcı girişindeki bir hane piyangodaki bir hane ile eşleşirse, ödül 1000 $ 'dır.

`switch` İfadeleri

```
switch (durum) {
  case 0: tekli başvurular  için vergi hesapla;
          break;
  case 1: ortak evli başvurular  için vergi hesapla ;
          break;
  case 2:  ayrı evli başvurular  için vergi hesapla ;
          break;
  case 3: hane reisi başvuruları  için vergi hesapla ;
          break;
  default: System.out.println("  Hata: geçersiz durum");
          System.exit(0);
}
```

`Switch` ifadesi akış diyagramı



`switch` ifadesi kuralları

Switch ifadesi char, bayt, short veya int türünde bir değer vermeli ve her zaman parantez içine alınmalıdır

value1, ... ve valueN, anahtar ifadesinin değeriyle aynı veri türüne sahip olmalıdır. Case ifadesindeki sonuç ifadeleri case ifadesindeki değer, switch ifadesinin değeriyle eşleştiğinde yürütülür. value1, ... ve valueN'nin sabit ifadeler olduğunu, ifadede 1 + x gibi değişkenleri içeremeyeceği anlamına gelir.

```
switch (switch-expression) {
  case value1:  statement(s)1;
                break;
  case value2: statement(s)2;
                break;
  …
  case valueN: statement(s)N;
                break;
  default: statement(s)-for-default;
}
```

`switch` ifadesi kuralları

Anahtar kelime break isteğe bağlıdır, ancak her durumun sonunda switch ifadesinin kalanını sonlandırmak için kullanılmalıdır. Break ifadesi yoksa, bir sonraki durum ifadesi çalıştırılır.

İsteğe bağlı olan default, belirtilen durumlardan hiçbiri switch ifadesiyle eşleşmediğinde eylemleri gerçekleştirmek için kullanılabilir.

```
switch (switch-expression) {
  case value1:  statement(s)1;
                break;
  case value2: statement(s)2;
                break;
  …
  case valueN: statement(s)N;
                break;
  default: statement(s)-for-default;
}
```

Durum (case) ifadeleri sırayla yürütülür, ancak durumları sırası (varsayılan case dahil) önemli değildir. Bununla birlikte, case'lerin mantıksal sırasını takip etmek ve default case'i en sonuna yerleştirmek iyi bir programlama stilidir.

animation

switch ifadesinin yürütülmesi

ch değeri, 'a‘ olsun:

```
switch (ch) {
  case 'a': System.out.println(ch);
  case 'b': System.out.println(ch);
  case 'c': System.out.println(ch);
}
```

animation

switch ifadesinin yürütülmesi

ch 'a‘ ile eşleşti:

```
switch (ch) {
  case 'a': System.out.println(ch);
  case 'b': System.out.println(ch);
  case 'c': System.out.println(ch);
}
```

animation

switch ifadesinin yürütülmesi

Bu satırı gerçekleştir

```
switch (ch) {
  case 'a': System.out.println(ch);
  case 'b': System.out.println(ch);
  case 'c': System.out.println(ch);
}
```

animation

switch ifadesinin yürütülmesi

Bu satırı gerçekleştir

```
switch (ch) {
  case 'a': System.out.println(ch);
  case 'b': System.out.println(ch);
  case 'c': System.out.println(ch);
}
```

animation

switch ifadesinin yürütülmesi

Bu satırı gerçekleştir

```
switch (ch) {
  case 'a': System.out.println(ch);
  case 'b': System.out.println(ch);
  case 'c': System.out.println(ch);
}
```

animation

switch ifadesinin yürütülmesi

Gelen ifadeyi gerçekleştir

```
switch (ch)
  case 'a': System.out.println(ch);
  case 'b': System.out.println(ch);
  case 'c': System.out.println(ch);
}
```

Next statement;

animation

switch ifadesinin yürütülmesi

ch değeri 'a‘ olsun:

```
switch (ch) {
  case 'a': System.out.println(ch);
            break;
  case 'b': System.out.println(ch);
            break;
  case 'c': System.out.println(ch);
}
```

animation

switch ifadesinin yürütülmesi

ch 'a' ile eşleşti:

```
switch (ch) {
  case 'a': System.out.println(ch);
            break;
  case 'b': System.out.println(ch);
            break;
  case 'c': System.out.println(ch);
}
```

animation

switch ifadesinin yürütülmesi

Bu satırı gerçekleştir

```
switch (ch) {
  case 'a': System.out.println(ch);
            break;
  case 'b': System.out.println(ch);
            break;
  case 'c': System.out.println(ch);
}
```

animation

switch ifadesinin yürütülmesi

Bu satırı gerçekleştir

```
switch (ch) {
  case 'a': System.out.println(ch);
            break;
  case 'b': System.out.println(ch);
            break;
  case 'c': System.out.println(ch);
}
```

animation

switch ifadesinin yürütülmesi

Gelen ifadeyi gerçekleştir

```
switch (ch)
  case 'a': System.out.println(ch);
            break;
  case 'b': System.out.println(ch);
            break;
  case 'c': System.out.println(ch);
}
```

Next statement;

Koşullu Operatör (Conditional Operator)

Aşağıdaki ifadeler denktir:

```
if (x > 0)
  y = 1          ➔        y = (x > 0) ? 1 : -1;
else
  y = -1;
```

(boolean-expression) ? expression1 : expression2

Ternary operator
Binary operator
Unary operator

Koşullu Operatör

```
if (num % 2 == 0)
  System.out.println(num + “is even”);
else
  System.out.println(num + “is odd”);


System.out.println(
  (num % 2 == 0)? num + “is even” :
  num + “is odd”);
```

Koşullu Operatör

```
(boolean-expression) ? exp1 : exp2
```

Çıktıyı Formatlama

Printf ifadesi çıktı için kullanılır.

System.out.printf(format, items);

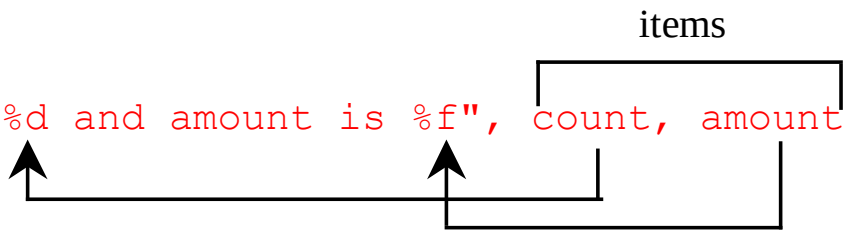
Biçim, alt dizgelerden (substrings) ve biçim belirticilerinden (format specifiers) oluşabilen bir dizgedir. Bir format belirteci, bir öğenin nasıl gösterilmesi gerektiğini belirtir. Bir öğe, sayısal bir değer, karakter, boolean değer veya bir dizge olabilir. Her belirleyici, yüzde işareti ile başlar.

Sıklıkla Kullanılanlar Belirticiler

Specifier	Output	Example
%b	a boolean value	true or false
%c	a character	'a'
%d	a decimal integer	200
%f	a floating-point number	45.460000
%e	a number in standard scientific notation	4.556000e+01
%s	a string	"Java is cool"

```
int count = 5;
double amount = 45.56;
System.out.printf("count is %d and amount is %f", count, amount);


display              count is 5 and amount is 45.560000
```



Operatör Öncelikleri

Precedence	*Operator*
↓	`var++` and `var--` (Postfix)
	`+`, `-` (Unary plus and minus), `++var` and `--var` (Prefix)
	(type) (Casting)
	`!` (Not)
	`*`, `/`, `%` (Multiplication, division, and remainder)
	`+`, `-` (Binary addition and subtraction)
	`<`, `<=`, `>`, `>=` (Relational)
	`==`, `!=` (Equality)
	`^` (Exclusive OR)
	`&&` (AND)
	`||` (OR)
	`=`, `+=`, `-=`, `*=`, `/=`, `%=` (Assignment operators)

Operatör Önceliği ve İlişkilendirme

3 + 4 * 4 > 5 * (4 + 3) – 1 && (4 – 3 > 5)

Parantez içindeki ifade önce değerlendirilir. (Parantezler iç içe geçebilir, bu durumda iç parantez içindeki ifade önce yürütülür.) Parantezsiz bir ifade değerlendirilirken, işleçler öncelik kuralına ve ilişkilendirme kuralına göre uygulanır.

Operatör Birleştirilmesi

Aynı önceliğe sahip iki operatör değerlendirildiğinde, operatörlerin ilişkililiği değerlendirme sırasını belirler. Atama operatörleri dışındaki tüm ikili operatörler sola dayalıdır :

a – b + c – d ifadesi **((a – b) + c) – d** denktir

Atama operatörleri sağa dayalıdır:

a = b += c = 5 ifadesi **a = (b += (c = 5))** denktir.

Örnek

Operatör önceliği ve ilişkilendirme kuralı uygulanarak, 3 + 4 * 4> 5 * (4 + 3) - 1 ifadesi şu şekilde değerlendirilir: