

Bölüm 2: Programlamaya Giriş



Motivasyon

Önceki bölümde, bir Java programını nasıl oluşturacağınızı, derleyeceğinizi ve çalıştıracağınızı öğrendiniz. Bu bölümden başlayarak, pratik problemlerin programlama mantığıyla nasıl çözüleceğini öğreneceksiniz. Bu problemlerle; Java ilkel (primitive) veri tiplerini, değişkenlerini, sabitlerini, veri tiplerini, operatörlerini, ifadelerini, girdi ve çıktı ile ilgili konuları öğreneceksiniz.



Amaçlar

- Basit hesaplamaları gerçekleştirmek için Java programları yazmak (§2.2).
- Scanner sınıfını kullanarak konsoldan girdi almak (§2.3).
- Değişkenleri, sabitleri, yöntemleri ve sınıfları adlandırmak için tanımlayıcıları (identifiers) kullanmak (§2.4).
- Verileri depolamak için değişken kullanmak (§§2.5-2.6).
- Atama (Assignment) deyimi (statement) ve ifadeleri (expression) ile programlamak (§2.6).
- Kalıcı verileri depolamak için sabitleri kullanmak (§2.7).
- Java ilkel veri türlerini öğrenmek: byte, short, int, long, float, double, and char (§§2.8.1).
- Sayısal ifadeleri yazmak için java operatörlerini kullanmak (§§2.8.2–2.8.3).
- Geçerli saati göstermek (§2.9).
- Short hand operatörlerini kullanmak (§2.10).
- Bir türün değerini başka bir türe atamak (§2.11).
- Kredi ödemesini hesaplamak (§2.12).
- Karakterleri temsil etmek için Char veri türünü kullanarak (§2.13).
- Parasal değişiklikleri hesaplamak (§2.14).
- Bir dizgeyi (String) temsil etmek için String veri türünü kullanmak (§2.15).
- Java dokümanlarına, programlama stiline ve adlandırma kurallarına aşinalık kazanmak (§2.16).
- Yazım (syntax) hatalarını, çalışma zamanı (runtime) hatalarını, mantık (logic) hatalarını ve hata ayıklama (debug) hatalarını ayırt etmek (§2.17).
- (GUI) JOptionPane giriş iletişim kutularını (input dialog boxes) kullanarak giriş elde etmek (§2.18).

Bir Örnek ile Programlamaya Giriş

Örnek 2.1 Bir dairenin alanının hesaplanması

Bu program bir dairenin alanını
hesaplar

ComputeArea

Run

IMPORTANT NOTE: To enable the buttons, you must download the entire slide file *slide.zip* and unzip the files into a directory (e.g., c:\slide) .

Programın Yürütülmesinin İzlenmesi

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```

radius değişkeni için bellekten
yer ayrılması (allocate)

radius

no value



Programın Yürütülmesinin İzlenmesi

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```

Bellek (memory)

radius no value

area no value

area değişkeni için bellekten
yer ayrılması (allocate)



Programın Yürütülmesinin İzlenmesi

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```

20 değerine Radius değişkenine ata

radius

20

area

no value



Programın Yürütülmesinin İzlenmesi

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```

memory

radius	20
area	1256.636

Alanı hesapla ve area değişkenine ata.



Programın Yürütülmesinin İzlenmesi

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double radius;  
        double area;  
  
        // Assign a radius  
        radius = 20;  
  
        // Compute area  
        area = radius * radius * 3.14159;  
  
        // Display results  
        System.out.println("The area for the circle of radius " +  
            radius + " is " + area);  
    }  
}
```

memory

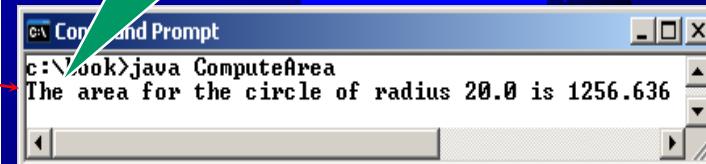
radius

20

area

1256.636

Konsolda (Console)
mesajı yaz.



Konsoldan (Console) Girdi (Input) Okunması

1. Scanner nesnesi (object) oluşturulur.

```
Scanner input = new Scanner(System.in);
```

2. Metotlardan biri kullanılır; next(), nextByte(), nextShort(), nextInt(), nextLong(), nextFloat(), nextDouble(), or nextBoolean() to obtain to a string, byte, short, int, long, float, double, or boolean value.
Örneğin,

```
System.out.print("Bir double değer giriniz:");  
Scanner input = new Scanner(System.in);
```

ComputeAreaWithConsoleInput

Run

ComputeAverage

Run

Tanımlayıcı (Identifiers)

- Tanımlayıcı harf, rakam, alt çizgi (_) ve dolar işaretlerinden (\$) oluşan bir karakter dizisidir.
- Bir tanımlayıcı, bir harf, alt çizgi (_) veya dolar işareti (\$) ile başlamalıdır. Rakamla başlayamaz.
- Bir tanımlayıcı, ayrılmış bir kelime olamaz. (Ayrılmış kelimelerin bir listesi için Ek A, “Java Anahtar Kelimeleri” ne bakınız).
- Bir tanımlayıcı `true`, `false`, or `null` olamaz.
- Bir tanımlayıcı herhangi bir uzunlukta olabilir.



Değişkenler (Variables)

```
// Compute the first area  
radius = 1.0;  
area = radius * radius * 3.14159;  
System.out.println("The area is " +  
    area + " for radius "+radius);
```

```
// Compute the second area  
radius = 2.0;  
area = radius * radius * 3.14159;  
System.out.println("The area is " +  
    area + " for radius "+radius);
```



Değişkenleri Bildirmek (Declaring Variables)

```
int x;           // x' in bir tamsayı  
                // değişkeni olduğunu  
                // bildirme;
```

```
double radius;  // radius' un bir double  
                // değişkeni olduğunu  
                // bildirme;
```

```
char a;         // a' nın bir char  
                // değişkeni olduğunu  
                // bildirme;
```



Atama İfadeleri

(Assignment Statements)

```
x = 1;           // 1 değerini x' e ata;
```

```
radius = 1.0;    // 1.0 değerini radius  
                // değişkenine ata;
```

```
a = 'A';         // 'A' değerini a  
                //değişkenine ata;
```



Bildirme ve Atama (Declaring and Initializing) İşlemlerini Tek Adımda Gerçekleştirme

```
□ int x = 1;
```

```
□ double d = 1.4;
```



Sabitler (Constants)

```
final datatype CONSTANTNAME = VALUE;
```

```
final double PI = 3.14159;
```

```
final int SIZE = 3;
```



Sayısal (Nümerik) Veri Tipleri

(Numerical Data Types)

Name	Range	Storage Size
byte	-2^7 (-128) to 2^7-1 (127)	8-bit signed
short	-2^{15} (-32768) to $2^{15}-1$ (32767)	16-bit signed
int	-2^{31} (-2147483648) to $2^{31}-1$ (2147483647)	32-bit signed
long	-2^{63} to $2^{63}-1$ (i.e., -9223372036854775808 to 9223372036854775807)	64-bit signed
float	Negative range: -3.4028235E+38 to -1.4E-45 Positive range: 1.4E-45 to 3.4028235E+38	32-bit IEEE 754
double	Negative range: -1.7976931348623157E+308 to -4.9E-324 Positive range: 4.9E-324 to 1.7976931348623157E+308	64-bit IEEE 754

Sayısal İşleçler (Numeric Operators)

Name	Meaning	Example	Result
+	Addition	34 + 1	35
-	Subtraction	34.0 - 0.1	33.9
*	Multiplication	300 * 30	9000
/	Division	1.0 / 2.0	0.5
%	Remainder	20 % 3	2

Tamsayı Bölümü (Integer Division)

+, -, *, /, ve %

$5 / 2 \rightarrow$ Bir tamsayı (2) sonuç verir.

$5.0 / 2 \rightarrow$ Bir ondalık değer (2,5) sonucu verir.

$5 \% 2 \rightarrow$ 1 sonucunu verir.

(bölme işlemi sonucunda kalanı verir).



Kalan İşleci (Remainder Operator)

Kalan işleci programlamada çok kullanışlıdır. Örneğin, % 2 işlemi çift sayılar için her zaman 0, tek sayılar için ise her zaman 1 sonucunu verir. Böylece, bir sayının çift mi yoksa tek mi olduğunu belirlemek için bu işleci kullanabilirsiniz. Bugün cumartesi olduğunu ve arkadaşlarınızla 10 gün içinde buluşacağınızı varsayalım. 10 gün sonrası hangi gün olacaktır? Aşağıdaki ifadeyi kullanarak o günün Salı olduğunu görebilirsiniz:

Saturday is the 6th day in a week



A week has 7 days

(6 + 10) % 7 is 2

The 2nd day in a week is Tuesday

After 10 days

Problem: Zamanı Gösterme

Saniye değerinden saat ve dakikayı elde eden programı yazınız.

DisplayTime

Run



NOT

Kayan noktalı (Floating-point) sayıları içeren hesaplamalar, bu sayılar tam bir doğrulukla kaydedilemediğinden yaklaştırılmıştır. Örneğin,

```
System.out.println(1.0 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1);
```

0.5'i değil 0.5000000000000000001'i gösterir.

```
System.out.println(1.0 - 0.9);
```

0.1'i değil 0.099999999999999999998'i gösterir.

Tamsayılar tam olarak saklanır. Bu nedenle, tamsayılarla yapılan hesaplamalar kesin bir tamsayı sonucu verir.

Sayı Değişmezleri (Number Literals)

Değişmez (Literal), program içerisinde doğrudan görünen sabit değerdir. Örneğin, 34, 1000000 ve 5.0 aşağıdaki ifadenin değişmezleridir.

```
int i = 34;
```

```
long x = 1000000;
```

```
double d = 5.0;
```



Tam Sayı Değişmezi (Integer Literals)

Bir tam sayı (integer) değişkenine, değişken boyutunun izin verdiği sınırlar içerisinde bir tam sayı değişmezi atanabilir. Değişmez, değişkenin tutabileceği boyuttan büyükse derleme hatası meydana gelir. Örneğin, *byte b = 1000* ifadesi bir derleme hatasına neden olacaktır, çünkü 1000 byte tipinde bir değişkende saklanamaz.

Bir tamsayı (integer) değişmezinin değeri integer değişkeninin sınırları olan -2^{31} (-2147483648) ile $2^{31}-1$ (2147483647) arasında olabilir. Long tipinde bir değişmezi tanımlamak L veya l harfi kullanılabilir. l harfi 1 rakamı ile karıştırılabildiği için L (Büyük Harf) harfi tercih edilir.

Kayan Noktalı Değişmez (Floating-Point Literals)

Kayan nokta değişmezleri ondalık basamakla yazılmıştır. Varsayılan olarak, kayan nokta değişmezi double tipinde değer olarak kabul edilir. Örneğin, 5.0 bir kayan (float) değer değil, double değer olarak kabul edilir. F veya f harfini ekleyerek bir sayıyı float yapabilir, aynı şekilde d veya D harfini de ekleyerek bir sayıya double yapabilirsiniz. Örneğin, bir kayan sayı için 100.2f veya 100.2F ifadelerini, bir double sayı için ise 100.2d veya 100.2 ifadelerini kullanabilirsiniz.

Bilimsel Gösterim (Scientific Notation)

Kayan nokta değişmezleri bilimsel gösterimde de belirtilebilir, örneğin $1.23456e2$ ile aynı olan $1.23456e + 2$, 123.456 'ya eşittir ve $1.23456e-2$, 0.0123456 'ya eşdeğerdir. E (veya e) bir üssünü temsil eder ve küçük veya büyük harf olabilir.

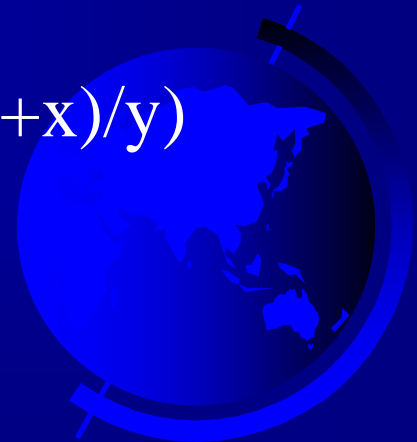


Aritmetik İfadeler (Arithmetic Expressions)

$$\frac{3+4x}{5} - \frac{10(y-5)(a+b+c)}{x} + 9\left(\frac{4}{x} + \frac{9+x}{y}\right)$$

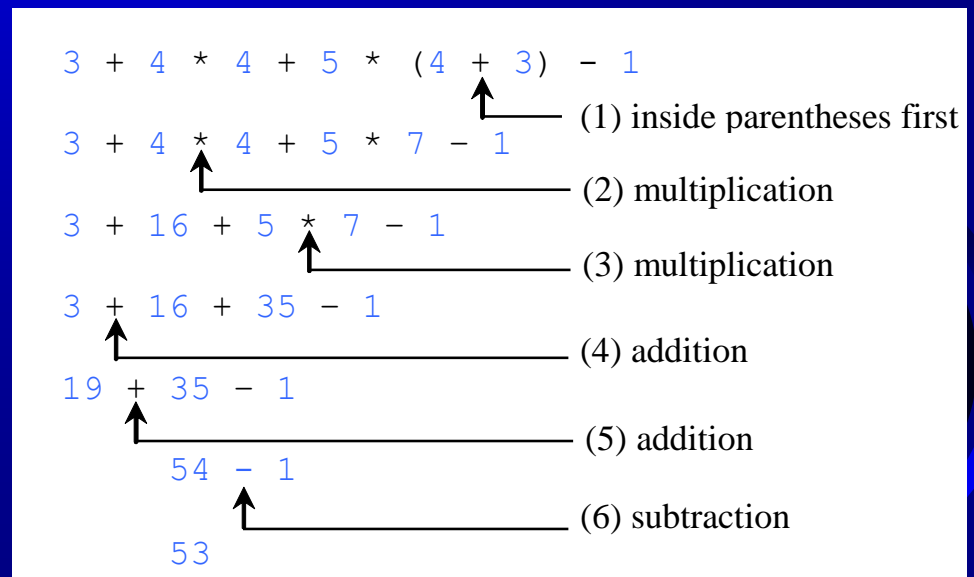
Yukarıdaki ifadenin programlama diline çevrilmiş hali aşağıdaki gibidir;

$$(3+4*x)/5 - 10*(y-5)*(a+b+c)/x + 9*(4/x + (9+x)/y)$$



Bir İfade Nasıl Değerlendirilir

Java, bir aritmetik ifadeyi değerlendirmek için kendi yoluna sahip olsa da, bir Java ifadesinin sonucu ve ona karşılık gelen aritmetik ifadesinin sonucu aynıdır. Bu nedenle, bir Java ifadesini değerlendirmek için aritmetik kuralı güvenle uygulayabilirsiniz.



Problem: Sıcaklıkları Dönüştürmek

Aşağıdaki formülü kullanarak Fahrenheit değerini Celsius değerine çeviren programı yazınız:

$$celsius = (\frac{5}{9})(fahrenheit - 32)$$

FahrenheitToCelsius

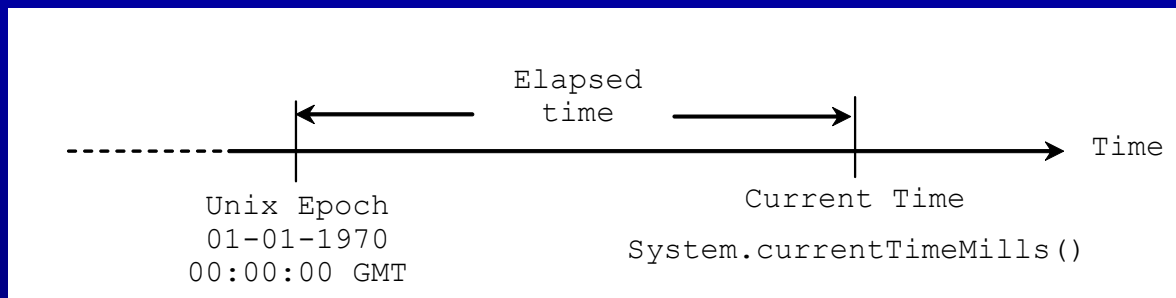
Run



Problem: Güncel Saati Gösterme

GMT'de şu anki zamanı saat: dakika: saniye, olacak şekilde yazan programı yazınız (Örneğin: 1:45:19 gibi).

currentTimeMillis, System sınıfındaki bir yöntemdir, geçerli zamanı 1 Ocak 1970 GMT gece yarısından bu yana milisaniye cinsinden verir. (1970, Unix işletim sisteminin resmen tanıtıldığı yıldır). Geçerli saati elde etmek için bu yöntemi kullanabilir ve ardından geçerli saniye, dakika ve saati aşağıdaki gibi hesaplayabilirsiniz.



ShowCurrentTime

Run

Kısayol Atama Operatörleri

<i>İşleç</i>	<i>Örnek</i>	<i>Eş değeri</i>
<code>+=</code>	<code>i += 8</code>	<code>i = i + 8</code>
<code>-=</code>	<code>f -= 8.0</code>	<code>f = f - 8.0</code>
<code>*=</code>	<code>i *= 8</code>	<code>i = i * 8</code>
<code>/=</code>	<code>i /= 8</code>	<code>i = i / 8</code>
<code>%=</code>	<code>i %= 8</code>	<code>i = i % 8</code>



Azaltma ve Arttırma İşleçleri

İşleç	Ad	Tanım
<u>++var</u>	preincrement	The expression (++var) increments <u>var</u> by 1 and evaluates to the <i>new</i> value in <u>var</u> <i>after</i> the increment.
<u>var++</u>	postincrement	The expression (var++) evaluates to the <i>original</i> value in <u>var</u> and increments <u>var</u> by 1.
<u>--var</u>	predecrement	The expression (--var) decrements <u>var</u> by 1 and evaluates to the <i>new</i> value in <u>var</u> <i>after</i> the decrement.
<u>var--</u>	postdecrement	The expression (var--) evaluates to the <i>original</i> value in <u>var</u> and decrements <u>var</u> by 1.



Azaltma ve Arttırma İşleçleri

```
int i = 10;
```

```
int newNum = 10 * i++;
```

Same effect as

```
int newNum = 10 * i;  
i = i + 1;
```

```
int i = 10;
```

```
int newNum = 10 * (++i);
```

Same effect as

```
i = i + 1;  
int newNum = 10 * i;
```

Azaltma ve Arttırma İşleçleri

Arttırma ve azaltma işleçlerini kullanmak ifadeleri kısaltır, ancak bu işleçler ifadeleri karmaşık ve okunması zor bir hale getirebilir.

Bu işleçleri, birden fazla değişkeni değiştiren ifadelerde veya bunun gibi birden çok kez aynı değişkeni kullanmaktan kaçının: `int k = ++ i + i.`



Atama İfadeleri ve Atama Yapıları

Java 2'den önce, tüm ifadeler statement olarak kullanılabilirken, Java 2'den bu yana, yalnızca aşağıdaki ifadeler statement olarak kullanılabilir:

`variable op= expression; // Where op is +, -, *, /, or %`

`++variable;`

`variable++;`

`--variable;`

`variable--;`



Sayısal Tip Dönüşümü (Numeric Type Conversion)

Aşağıdaki ifadeleri inceleyiniz:

```
byte i = 100;
```

```
long k = i * 3 + 4;
```

```
double d = i * 3.1 + k / 2;
```



Dönüşüm Kuralları (Conversion Rules)

Farklı tipte 2 operand içeren binary (ikili) bir işlem gerçekleştirirken, Java aşağıdaki kurallara göre operand' ı otomatik olarak çevirir.

1. Operand' lardan biri double ise diğeri de double' a çevrilir.
2. Operand' lardan biri float ise diğeri de float' a çevrilir.
3. Operand' lardan biri long ise diğeri de long' a çevrilir.
4. Aksi takdirde iki operand' da int ' e çevrilir



Tip (Type) Dönüşümü

Implicit casting

```
double d = 3; (type widening)
```

Explicit casting

```
int i = (int)3.0; (type narrowing)
```

```
int i = (int)3.9; (Fraction part is truncated)
```

What is wrong? `int x = 5 / 2.0;`

range increases

byte, short, int, long, float, double

Problem: Ondalık Noktadan Sonra İki Rakam Tutmak

Satış vergisini, virgülden sonra iki basamaklı olarak gösteren bir program yazınız.

SalesTax

Run



Problem:

Bilgi İşlem Kredisİ Ödemeleri

Bu program, kullanıcının faiz oranını, yıl sayısını ve kredi tutarını girmesini sağlar ve aylık ödemeyi ve toplam ödemeyi hesaplar.

$$\text{monthlyPayment} = \frac{\text{loanAmount} \times \text{monthlyInterestRate}}{1 - \frac{1}{(1 + \text{monthlyInterestRate})^{\text{numberOfYears} \times 12}}}$$

ComputeLoan

Run

Karakter Veri Tipi (Character Data Type)

```
char letter = 'A'; (ASCII)
```

```
char numChar = '4'; (ASCII)
```

```
char letter = '\u0041'; (Unicode)
```

```
char numChar = '\u0034'; (Unicode)
```

Four hexadecimal digits.

NOT: Artırma ve azaltma operatörleri, bir sonraki veya bir önceki Unicode karakterini elde etmek için char değişkenlerinde de kullanılabilir. Örneğin, aşağıdaki ifadeler b karakterini gösterir.

```
char ch = 'a';
```

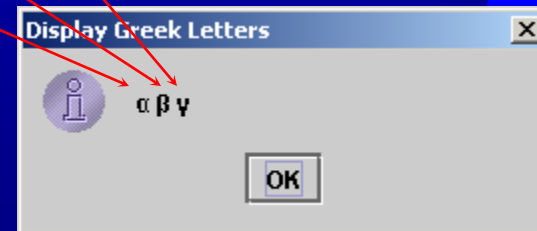
```
System.out.println(++ch);
```



Unicode Biçimi (Format)

Java karakterleri, Unicode Konsorsiyumu tarafından dünyanın çeşitli dillerinde yazılı metinlerin değişimini, işlenmesini ve görüntülenmesini desteklemek için kurulan 16 bitlik bir kodlama şeması olan Unicode'u kullanır. Unicode, '\ u0000' ile '\ uFFFF' arasında çalışan dört onaltılık sayılarla ifade edilen \ u ile gelen iki baytı alır. Böylece, Unicode $65535 + 1$ karakter ile temsil edebilir.

Üç Yunan harfi için, Unicode \u03b1 \u03b2 \u03b3



Özel karakterler için Çıkış Tuşu Dizisi

<i>Tanım</i>	<i>Çıkış Tuşu</i>	<i>Unicode</i>
Backspace	\b	\u0008
Tab	\t	\u0009
Linefeed	\n	\u000A
Carriage return	\r	\u000D
Backslash	\\	\u005C
Single Quote	\'	\u0027
Double Quote	\"	\u0022



Ek B: ASCII Karakter Kümesi

ASCII Karakter Kümesi, Unicode'un bir alt kümesidir;

\ u0000 ile \ u007f aras1

TABLE B.1 ASCII Character Set in the Decimal Index

	0	1	2	3	4	5	6	7	8	9
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	nl	vt	ff	cr	so	si	dle	dcl	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	sp	!	"	#	\$	%	&	'
4	()	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[\]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{		}	~	del		

ASCII Karakter Kümesi

ASCII Karakter Kümesi, Unicode'un bir alt kümesidir;

\ u0000 ile \ u007f aras1

TABLE B.2 ASCII Character Set in the Hexadecimal Index

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht	nl	vt	ff	cr	so	si
1	dle	dcl	dc2	dc3	dc4	nak	syn	etb	can	em	sub	esc	fs	gs	rs	us
2	sp	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	del

Char ve Sayısal (Nümerik) Tipler Arasında Dönüşüm (Casting)

```
int i = 'a'; // Same as int i = (int) 'a';
```

```
char c = 97; // Same as char c = (char) 97;
```



Problem: Parasal Birimler

Bu program kullanıcının miktarı ve kurları temsil eden ondalık sayıyı girmesini ve parasal eşdeğeri tek dolar, çeyrek, dime (10 cent), nickel (5 cent) ve penny (1 cent) cinsinden listeleyen bir rapor çıkarmasını sağlar.

Programınız, bu sırayla maksimum dolar sayısını, ardından maksimum çeyrek sayısını vb. Bildirmelidir.

ComputeChange

Run

Hesaplama Değişimini İzleyin

Amount değerini 11.56 olarak kabul edin.

```
int remainingAmount = (int)(amount * 100);
```

```
// Find the number of one dollars
```

```
int numberOfOneDollars = remainingAmount / 100;
```

```
remainingAmount = remainingAmount % 100;
```

```
// Find the number of quarters in the remaining amount
```

```
int numberOfQuarters = remainingAmount / 25;
```

```
remainingAmount = remainingAmount % 25;
```

```
// Find the number of dimes in the remaining amount
```

```
int numberOfDimes = remainingAmount / 10;
```

```
remainingAmount = remainingAmount % 10;
```

```
// Find the number of nickels in the remaining amount
```

```
int numberOfNickels = remainingAmount / 5;
```

```
remainingAmount = remainingAmount % 5;
```

```
// Find the number of pennies in the remaining amount
```

```
int numberOfPennies = remainingAmount;
```

remainingAmount

1156

remainingAmount
Atanmış ilk değer



Hesaplama Değişimini İzleyin

Amount değerini 11.56 olarak kabul edin.

```
int remainingAmount = (int)(amount * 100);
```

```
// Find the number of one dollars
```

```
int numberOfOneDollars = remainingAmount / 100;
```

```
remainingAmount = remainingAmount % 100;
```

```
// Find the number of quarters in the remaining amount
```

```
int numberOfQuarters = remainingAmount / 25;
```

```
remainingAmount = remainingAmount % 25;
```

```
// Find the number of dimes in the remaining amount
```

```
int numberOfDimes = remainingAmount / 10;
```

```
remainingAmount = remainingAmount % 10;
```

```
// Find the number of nickels in the remaining amount
```

```
int numberOfNickels = remainingAmount / 5;
```

```
remainingAmount = remainingAmount % 5;
```

```
// Find the number of pennies in the remaining amount
```

```
int numberOfPennies = remainingAmount;
```

remainingAmount

1156

numberOfOneDollars

11

numberOfOneDollars
Atanmış Değer



Hesaplama Değişimini İzleyin

Amount değerini 11.56 olarak kabul edin.

```
int remainingAmount = (int)(amount * 100);
```

```
// Find the number of one dollars
```

```
int numberOfOneDollars = remainingAmount / 100;
```

```
remainingAmount = remainingAmount % 100;
```

```
// Find the number of quarters in the remaining amount
```

```
int numberOfQuarters = remainingAmount / 25;
```

```
remainingAmount = remainingAmount % 25;
```

```
// Find the number of dimes in the remaining amount
```

```
int numberOfDimes = remainingAmount / 10;
```

```
remainingAmount = remainingAmount % 10;
```

```
// Find the number of nickels in the remaining amount
```

```
int numberOfNickels = remainingAmount / 5;
```

```
remainingAmount = remainingAmount % 5;
```

```
// Find the number of pennies in the remaining amount
```

```
int numberOfPennies = remainingAmount;
```

remainingAmount

56

numberOfOneDollars

11

remainingAmount
Güncellendi



Hesaplama Değişimini İzleyin

Amount değerini 11.56 olarak kabul edin.

```
int remainingAmount = (int)(amount * 100);
```

```
// Find the number of one dollars
```

```
int numberOfOneDollars = remainingAmount / 100;  
remainingAmount = remainingAmount % 100;
```

```
// Find the number of quarters in the remaining amount
```

```
int numberOfQuarters = remainingAmount / 25;  
remainingAmount = remainingAmount % 25;
```

```
// Find the number of dimes in the remaining amount
```

```
int numberOfDimes = remainingAmount / 10;  
remainingAmount = remainingAmount % 10;
```

```
// Find the number of nickels in the remaining amount
```

```
int numberOfNickels = remainingAmount / 5;  
remainingAmount = remainingAmount % 5;
```

```
// Find the number of pennies in the remaining amount
```

```
int numberOfPennies = remainingAmount;
```

remainingAmount 56

numberOfOneDollars 11

numberOfOneQuarters 2

numberOfOneQuarters
Atanmış Değer



Hesaplama Değişimini İzleyin

Amount değerini 11.56 olarak kabul edin.

```
int remainingAmount = (int)(amount * 100);
```

```
// Find the number of one dollars
```

```
int numberOfOneDollars = remainingAmount / 100;  
remainingAmount = remainingAmount % 100;
```

```
// Find the number of quarters in the remaining amount
```

```
int numberOfQuarters = remainingAmount / 25;  
remainingAmount = remainingAmount % 25;
```

```
// Find the number of dimes in the remaining amount
```

```
int numberOfDimes = remainingAmount / 10;  
remainingAmount = remainingAmount % 10;
```

```
// Find the number of nickels in the remaining amount
```

```
int numberOfNickels = remainingAmount / 5;  
remainingAmount = remainingAmount % 5;
```

```
// Find the number of pennies in the remaining amount
```

```
int numberOfPennies = remainingAmount;
```

remainingAmount

6

numberOfOneDollars

11

numberOfQuarters

2

remainingAmount
Güncellendi



Dizge (String) Tipi

Char türü yalnızca bir karakteri temsil eder. Bir karakter dizesini temsil etmek için, String adlı veri türü kullanılır. Örneğin, String message = "Welcome to Java";

Dizge aslında Java kütüphanesinde tıpkı System sınıfı ve JOptionPane sınıfı gibi önceden tanımlanmış bir sınıftır. Dizge türü ilkel bir tür değildir. Referans türü olarak bilinir. Herhangi bir Java sınıfı, bir değişken için referans türü olarak kullanılabilir. Referans veri tipleri, “Nesneler ve Sınıflar” başlıklı 7. Bölümde ayrıntılı olarak ele alınacaktır. Şimdilik, bir String değişkeninin nasıl bildirileceğini (declare), değişkene bir stringin nasıl atanacağını ve dizgelerin nasıl birleştirileceğini bilmeniz yeterlidir.

StringBirleştirme (Concatenation)

// Üç adet String birleştirildi.

String message = "Welcome " + "to " + "Java";

// String Chapter «2» sayısı ile birleştirildi.

String s = "Chapter" + 2; // s becomes Chapter2

// String Supplement 'B' karakteri ile birleştirildi.

String s1 = "Supplement" + 'B'; // s1 becomes SupplementB



Programlama Stili ve Belgeler

- Uygun Yorumlar
- Adlandırma Kuralları
- Doğru Girintileme ve Boşluk Satırları
- Blok Stilleri



Uygun Yorumlar

Programın ne işe yaradığını, temel özelliklerini, destekleyici veri yapılarını ve kullandığı benzersiz teknikleri açıklamak için programın başına bir özet ekleyin.

Programın başında adınızı, sınıf bölümünüzü, eğitmeninizi, tarihinizi ve kısa bir açıklamayı ekleyin.



Adlandırma Kuralları

- Anlamlı ve açıklayıcı isimler seçin.
- Değişkenler ve Yöntem Adları
 - Küçük harf kullanın. Eğer isim birkaç kelimeden oluşuyorsa, hepsini bir arada birleştirin, ilk kelime için küçük harf kullanın ve sonraki her kelimenin ilk harfini büyük harfle yazınız. Örneğin, radius ve area değişkenleri ve computeArea yöntemi gibi.



Adlandırma Kuralları

□ Sınıf (Class) İsimleri:

- Sınıf ismindeki her kelimenin ilk harfini büyük harfle yazın.
- Örneğin, `ComputeArea` sınıf adı gibi.

□ Sabitler:

- Sabitlerde, tüm harfleri büyük yazın ve kelimeleri birleştirmek için alt çizgi (`_`) kullanın. Örneğin; `PI` ve `MAX_VALUE` sabitleri gibi.



Doğru Girintileme ve Boşluk Satırları

- Girintili Yazma
 - 2 boşluk girinti bırakın.
- Aralık (Spacing)
 - Kodun bölümlerini ayırmak için boş satır kullanın..



Blok Stilleri

Parantezler için satır sonu (End-of-line) stilini kullanın.

*Next-line
style*

```
public class Test
{
    public static void main(String[] args)
    {
        System.out.println("Block Styles");
    }
}
```

*End-of-line
style*

```
public class Test {
    public static void main(String[] args) {
        System.out.println("Block Styles");
    }
}
```

Programlama Hataları (Programming Errors)

- Yazım (Syntax) Hataları
 - Derleyici (Compiler) tarafından algılanır.
- Çalışma Zamanı (Runtime) Hataları
 - Programın iptal edilmesine neden olurlar.
- Mantık (Logic) Hataları
 - Yanlış sonuç üretirler.



Yazım (Syntax) Hataları

```
public class ShowSyntaxErrors {  
    public static void main(String[] args) {  
        i = 30;  
        System.out.println(i + 4);  
    }  
}
```



Çalışma Zamanı (Runtime) Hataları

```
public class ShowRuntimeErrors {  
    public static void main(String[] args) {  
        int i = 1 / 0;  
    }  
}
```



Mantık (Logic) Hataları

```
public class ShowLogicErrors {  
    // Determine if a number is between 1 and 100 inclusively  
    public static void main(String[] args) {  
        // Prompt the user to enter a number  
        String input = JOptionPane.showInputDialog(null,  
            "Please enter an integer:",  
            "ShowLogicErrors", JOptionPane.QUESTION_MESSAGE);  
        int number = Integer.parseInt(input);  
  
        // Display the result  
        System.out.println("The number is between 1 and 100, " +  
            "inclusively? " + ((1 < number) && (number < 100)));  
  
        System.exit(0);  
    }  
}
```


Hata Ayıklama (Debugging)

Mantık hatalarına böcek (bug) denilmektedir. Hataları bulma ve düzeltme işlemine ise hata ayıklama (debugging) adı verilir. Hata ayıklamaya ortak bir yaklaşım, hatanın bulunduğu programı daraltmak için bir yöntem kombinasyonunun kullanılmasıdır. Programı elle izleyebilir (yani, programı okuyarak hataları yakalayabilirsiniz) veya değişkenlerin değerlerini veya programın yürütme akışını göstermek için basılı ifadeler (print statement) ekleyebilirsiniz. Bu yaklaşım kısa, basit bir program için işe yarayabilir. Ancak büyük, karmaşık bir program için, hata ayıklama için en etkili yaklaşım bir hata ayıklayıcı yardımcı programını kullanmaktır.

Hata Ayıklayıcı (Debugger)

Hata ayıklayıcı, hata ayıklamayı kolaylaştıran bir programdır. Bir hata ayıklayıcıyı aşağıdaki gibi kullanabilirsiniz;

- Bir seferde tek bir ifade yürütün.
- Bir yönteme adım atın ya da bir yönteme geçin.
- Kesme noktalarını ayarlayın.
- Değişkenleri göster.
- Çağrı yığınını göster.
- Değişkenleri değiştirin.



Dizgeleri (Strings) Tam sayılara (Integers) Dönüştürme

Giriş iletişim kutusundan döndürülen giriş bir dizgedir. 123 gibi bir sayısal değer girerseniz, “123” değerini döndürür. Girişi sayı olarak almak için bir dizgeyi sayıya dönüştürmeniz gerekir.

Bir dizgeyi (String) int değerine dönüştürmek için, Integer sınıfındaki statik parseInt yöntemini aşağıdaki gibi kullanabilirsiniz:

```
int intValue = Integer.parseInt(intString);
```

where intString is a numeric string such as “123”.



Dizgeleri (Strings) Double' a Dönüştürme

Bir dizgyi (String) double değere dönüştürmek için, Double sınıfındaki statik `parseDouble` yöntemini aşağıdaki gibi kullanabilirsiniz:

```
double doubleValue = Double.parseDouble(doubleString);
```

where doubleString is a numeric string such as “123.45”.



Problem: Giriş İletişim Kutularını Kullanarak Kredi Ödemelerini Hesaplama

Kredi ödemelerinin hesaplanması için önceki programdakiyle aynıdır, ancak girdi iletişim kutularından girdi girilmesi ve çıktının çıktı iletişim kutusunda görüntülenmesi gerekmektedir.

$$1 - \frac{\text{loanAmount} \times \text{monthlyInterestRate}}{(1 + \text{monthlyInterestRate})^{\text{numberOfYears} \times 12}}$$

ComputeLoanUsingInputDialog

Run