

ALGORİTMA VE PROGRAMLAMA I

HAFTA 11 METOTLAR 2

Ders İçeriği: Hafta 11

- Giriş
- Hazır Metotlar
- Matematiksel Metotlar
- Özyinelemeli Metotlar
- Örnekler

Giriş

```
public static void main(String[] args) {
```

```
    int sonuc = toplam(3, 6);
```

```
}
```

1 usage

```
public static int toplam(int sayi1, int sayi2) {
```

```
    int toplam = sayi1 + sayi2;
```

```
    return toplam;
```

```
}
```

Hazır Metotlar

- Nesne Yönelimli Diller çok zengin kütüphanelere sahiptir.
- Bu kütüphanelerde paketler, paketlerin içinde sınıflar ve sınıfların içinde pek çok işi görecektir metotlar vardır.
- Kullanıcı ihtiyaç duyarsa kendi paket, sınıf ve metotlarını yazabilmektedir.
- Bir önceki konuda sınıf içerisinde metot tanımlama örneklerini inceledik.
- Ayrıca, şimdiye kadar kullandığımız `print`, `println` ve `printf` metotları `System` sınıfına ait metotlardır.

Matematiksel Metotlar

➤ Math Sınıfı

- System.Math sınıfında bulunan metotlar matematiksel işlevleri yerine getirmede kullanılırlar.
- Bu sınıfa ait metotlar statik tanımlıdır yani nesne oluşturulmadan doğrudan erişilebilir.
- Metot çağrısı Math.MetotAdi() şeklinde yapılır.
- İlgili metodun işlevine göre bir veya iki parametre ile metot çağrısı gerçekleştirilir.

```
public class Main {  
    public static void main(String[] args) {  
        Math.sqrt(5);  
    }  
}
```

Matematiksel Metotlar

➤ Math Sınıfı Metotları

Method	Description	Return Type
<code>abs(x)</code>	Returns the absolute value of x	double float int long
<code>acos(x)</code>	Returns the arccosine of x, in radians	double
<code>asin(x)</code>	Returns the arcsine of x, in radians	double
<code>atan(x)</code>	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians	double
<code>atan2(y,x)</code>	Returns the angle theta from the conversion of rectangular coordinates (x, y) to polar coordinates (r, theta).	double

Matematiksel Metotlar

➤ Math Sınıfı Metotları

Method	Description	Return Type
<code>cbrt(x)</code>	Returns the cube root of x	double
<code>ceil(x)</code>	Returns the value of x rounded up to its nearest integer	double
<code>copySign(x, y)</code>	Returns the first floating point x with the sign of the second floating point y	double
<code>cos(x)</code>	Returns the cosine of x (x is in radians)	double
<code>cosh(x)</code>	Returns the hyperbolic cosine of a double value	double

Matematiksel Metotlar

➤ Math Sınıfı Metotları

Method	Description	Return Type
<code>exp(x)</code>	Returns the value of E^x	double
<code>expm1(x)</code>	Returns $e^x - 1$	double
<code>floor(x)</code>	Returns the value of x rounded down to its nearest integer	double
<code>getExponent(x)</code>	Returns the unbiased exponent used in x	int
<code>hypot(x, y)</code>	Returns $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow	double

Matematiksel Metotlar

➤ Math Sınıfı Metotları

Method	Description	Return Type
$\log(x)$	Returns the natural logarithm (base E) of x	double
$\log_{10}(x)$	Returns the base 10 logarithm of x	double
$\log_{1p}(x)$	Returns the natural logarithm (base E) of the sum of x and 1	double
$\max(x, y)$	Returns the number with the highest value	double float int long
$\min(x, y)$	Returns the number with the lowest value	double float int long

Matematiksel Metotlar

➤ Math Sınıfı Metotları

Method	Description	Return Type
nextAfter(x, y)	Returns the floating point number adjacent to x in the direction of y	double float
nextUp(x)	Returns the floating point value adjacent to x in the direction of positive infinity	double float
pow(x, y)	Returns the value of x to the power of y	double
round(x)	Returns the value of x rounded to its nearest integer	int
rint(x)	Returns the double value that is closest to x and equal to a mathematical integer	double

Matematiksel Metotlar

➤ Math Sınıfı Metotları

Method	Description	Return Type
signum(x)	Returns the sign of x	double
sin(x)	Returns the sine of x (x is in radians)	double
sinh(x)	Returns the hyperbolic sine of a double value	double
sqrt(x)	Returns the square root of x	double

Matematiksel Metotlar

➤ Math Sınıfı Metotları

Method	Description	Return Type
$\tan(x)$	Returns the tangent of an angle	double
$\tanh(x)$	Returns the hyperbolic tangent of a double value	double
$\text{toDegrees}(x)$	Converts an angle measured in radians to an approx. equivalent angle measured in degrees	double
$\text{toRadians}(x)$	Converts an angle measured in degrees to an approx. angle measured in radians	double
$\text{ulp}(x)$	Returns the size of the unit of least precision (ulp) of x	double float

Matematiksel Metotlar

➤ Rasgele Sayılar

- Programlama sırasında rasgele sayıların bilgisayar tarafından oluşturulması için `Math.random()` metodu kullanılır.
- Metot 0.0 ile 1.0 arasında rasgele sayılar üretir.

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println(Math.random());  
    }  
}
```

0.941973677937289

Matematiksel Metotlar

➤ Rasgele Sayılar

- Farklı aralıklarda sayılar ürettirmek için metot üzerinde işlem yapılması gerekir.
- Örneğin 0 ile 100 arasında sayı üretilmesi isteniyorsa:

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println((Math.random() * 100));  
        System.out.println((int)(Math.random() * 100));  
    }  
}
```

5.816652009272261
14

Özyinelemeli Metotlar

- Recursive (Özyinelemeli) metot kendi kendini çağıran metottur.
- Bu teknik karmaşık problemleri çözmesi daha kolay problemlere ayırmayı sağlar.
- Bilgisayar programcılığında zor problemleri çözen önemli bir algoritma kavramıdır.
- Recursive fonksiyon içinde iki bölüm bulunur:
 - Tekrarlanan (recursive) durum: Tekrar tekrar metot çağrısı yapılır ve bir değer döndürülür. Burada yazılan adımda fonksiyon bir alt basamağı çağırarak şekilde yazılmalıdır.
 - Temel (base) durum: Metot için bir taban değer oluşturulur, yani bu noktaya ulaştıktan sonra metot çağrısını durdurulur ve işlemler hesaplanıp değerlendirilir.

Özyinelemeli Metotlar

- Faktöriyel hesaplama problemini 4 sayısı için inceleyelim.

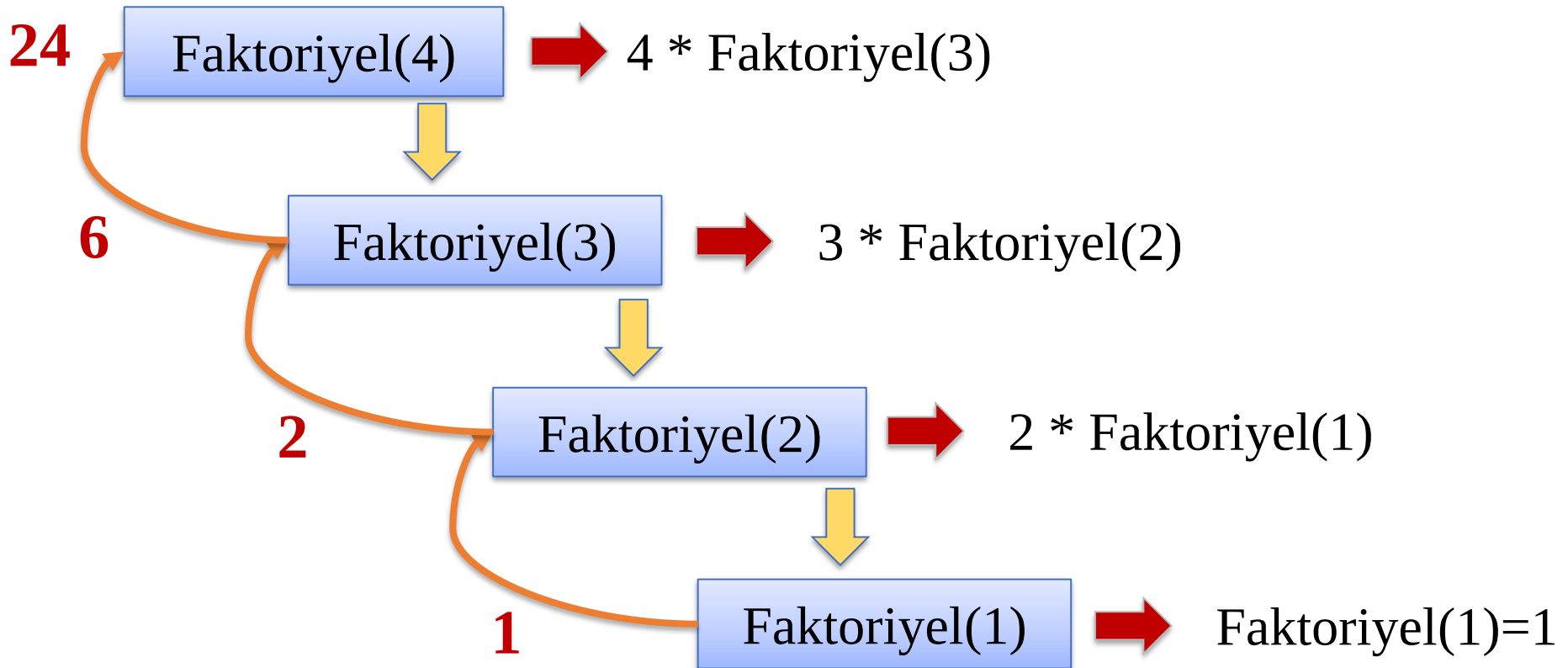


- Faktöriyel hesabını döngü ile yapan örnek fonksiyon:

```
public static long Faktoriyel(int sayi){  
    int faktoriyel=1;  
    for(int i=1;i<=sayi;i++)  
        faktoriyel=faktoriyel*i;  
    return faktoriyel;  
}
```

Özyinelemeli Metotlar

- Faktöriyel hesaplama problemini özyinelemeli yapıda inceleyelim.



Özyinelemeli Metotlar

- Faktöriyel hesabını özyineleme ile yapan örnek fonksiyon:

```
public static long RecursiveFaktoriyel(int sayi){  
    if (sayi<=1)  
        return 1;  
    else  
        return sayi*RecursiveFaktoriyel(sayi-1);  
}
```

Örnek 1

- Klavyeden girilen iki sayının ekok'unu bulan programı yazalım.

```
public static void main(String[] args) {
    Scanner kl=new Scanner(System.in);
    System.out.println("Birinci sayıyı giriniz:");
    int sayi1= kl.nextInt();
    System.out.println("İkinci sayıyı giriniz:");
    int sayi2= kl.nextInt();
    System.out.println("Sayıların ekoku:"+ekokBul(sayi1,sayi2));
}

1 usage
public static int ekokBul(int a, int b){
    for(int i=Math.max(a,b);i<=a*b;i++)
    {
        if(i%a==0 && i%b==0)
            return i;
    }
    return 0;
}
```

Örnek 2

- İlk 15 fibonacci sayısını ekrana yazdıran recursive metodu oluşturalım.

```
public static void main(String[] args) {  
    for(int i=0;i<15;i++){  
        System.out.println(fibonacci(i));  
    }  
}
```

3 usages

```
public static int fibonacci(int sayi){  
    if(sayi==1 || sayi==0)  
        return 1;  
    else  
        return fibonacci(sayi: sayi-1)+fibonacci(sayi: sayi-2);  
}
```

Örnek 3

- Klavyeden girilen iki sayı için üs alma işlemini recursive olarak bulan metodu yazalım ve Math.pow() ile sonucu karşılaştıralım.

```
public static void main(String[] args) {  
    System.out.println(usAlma( taban: 5, us: 4));  
    System.out.println(Math.pow(5,4));  
}
```

2 usages

```
public static double usAlma(int taban, int us){  
    if(us==0)  
        return 1;  
    else  
        return taban*usAlma(taban, us: us-1);  
}
```

Örnek 3

- Klavyeden girilen iki sayı için üs alma işlemini recursive olarak bulan metodu yazalım ve Math.pow() ile sonucu karşılaştıralım.

Execution steps			
Iteration	power()	powerRaised	result
1	power(3, 4)	4	$3 * \text{result}_2$
2	power(3, 3)	3	$3 * 3 * \text{result}_3$
3	power(3, 2)	2	$3 * 3 * 3 * \text{result}_4$
4	power(3, 1)	1	$3 * 3 * 3 * 3 * \text{result}_{\text{final}}$
Final	power(3, 0)	0	$3 * 3 * 3 * 3 * 1 = 81$

Örnek 4

- Klavyeden girilen 10 tabanındaki sayının 2 tabanındaki karşılığını recursive olarak ekrana yazdıran metodu oluşturalım.

```
public static void main(String[] args) {  
    ikiliSayi(23);  
}
```

2 usages

```
public static void ikiliSayi(int sayi){  
    if (sayi>0)  
    {  
        int kalan=sayi%2;  
        sayi=sayi/2;  
        ikiliSayi(sayi);  
        System.out.print(kalan);  
    }  
}
```

Örnek 4

- Klavyeden girilen 10 tabanındaki sayının 2 tabanındaki karşılığını recursive bulan metodu oluşturalım. (Çalışma Mantığı Örneği)

```
23 / 2 is 11 remainder 1
11 / 2 is  5 remainder 1
 5 / 2 is  2 remainder 1
 2 / 2 is  1 remainder 0
 1 / 2 is  0 remainder 1
```

23 in binary is 10111.

Örnek 5

- Klavyeden girilen iki sayının ebob'unu recursive bulan metodu yazalım.

```
public static void main(String[] args) {  
    Scanner kl=new Scanner(System.in);  
    System.out.println("Birinci sayıyı giriniz:");  
    int sayi1= kl.nextInt();  
    System.out.println("İkinci sayıyı giriniz:");  
    int sayi2= kl.nextInt();  
    System.out.println("Sayıların ebobu:"+ebobBul(sayi1,sayi2));  
}
```

2 usages

```
public static int ebobBul(int a, int b){  
    if (b != 0)  
        return ebobBul(b, a%b);  
    else  
        return a;  
}
```

Örnek 5

- Klavyeden girilen iki sayının ebob'unu recursive bulan metodu yazalım. (hcf- ebobBul fonksiyonu)

Execution Steps				
No.	Recursive call	n1	n2	n1 % n2
1	hcf(366, 60)	366	60	6
2	hcf(60, 6)	60	6	0
Final	hcf(6, 0)	6	0	-