

YMH112

ALGORİTMA ve PROGRAMLAMA

II

LAB. KISMI

1. CANLI DERS GÜNDÜZ (A)GRUBU

25 MART 2021

SAAT =11.00-12:00

ARŞ. GÖR. ALEV KAYA

Metotlar

- Metodlar java programlarının **ana parçalarıdır**.
- Metodlar **sınıfların(class) içinde** yer alan küçük program parçacıklarıdır.
- Metodların çoğunda **değişken parametreler metotlar ve sınıflar** arasında **iletişimi** sağlarlar.
- Ayrıca her metodun kendine özgü değişkenleri de vardır.
- Metod yapısının ana sebebi programları **modüler** hale getirmektir.
- Aynı zamanda aynı program parçacığının **tekrarlanmasını önlemeyi de** sağlar.
- Her metod çağrıldığı program parçacığına belli bir değişkeni aktarabilir.
- Metodların tanımlarında aktardıkları değişken türü tanımlanır.
- Eğer metod hiçbir değişken aktarmıyorsa **void** sözcüğüyle tanımlanır.

Metodların özellikleri

- **Class içerisinde** tanımlanmalıdır.
- Modifier bulunmalıdır (private,public,protected yada belirtmezsek default).
- **return type** herhangi bir veri tipi yada **void** olabilir.
- Metod ismi belirtilmelidir.
- Metod parametreleri parantez içerisinde belirtilmelidir . Eğer parametre almayacaksa parantez açılıp kapatılır.
- Metod içerisinde kod blokları **{ işaretiyle başlar } işaretiyle** sonlandırılır.
- Metod parametrelerini ayırmak için **,** işareti kullanılır.
- **Exception lar** metodların yanında **{ işaretinden** önce belirtilebilir.

Metotlar oluřturulma řekillerine gre ikiye ayrılırlar

1-nceden Tanımlanmıř Metotlar : Java ktphanelerinde bulunan hazır metotlardır.

rneęin : `pow(x,y)` metodu. Math sınıfına ait olan bu metot, x ve y olarak iki deęer alır ve bir deęer dndrr. Grevi ilk deęerin(x), ikinci deęer(y)ssn almaktır.

2-Kullanıcı Tanımlı Metotlar: Bizim yazdığımız ve oluřturduğumuz metotlardır.

Metotlar

- Java programlama dilinde aynı işlemi yapan kod satırlarına **metotlar** denir.
- Program içerisinde oluşturulan bir kod bloğunun birkaç kez kullanılması gerektiği durumlarda metotlardan faydalanılır.
- İki sayının dört aritmetik işlemi hesaplama formülünü yazmamız istensin.
- x ve y sayıları temsil ettiğini varsayarak girilen her x ve her y değeri için yeniden **$x*y$, $x+y$, x/y , $x-y$** yapmamız gerekir.
- Fakat oluşturulan bir metot ile bu metoda x ve y değerlerini göndererek cevap bekleyebiliriz.
- **Metotları diğer programlama dillerinde yer alan fonksiyonlar gibi düşünebilir.**
- **Erişim seviyeleri vardır,**
- **Parametre alırlar,**
- **Bir değer döndürürler.**

Metotların genel yapısı

Öncelikle bir **Main metoduna** bakarak hatırlıyalım.

Nasıldı o main metodu :

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

Şimdi bunu genel bir ifadeyle yorumlarsak;

```
erişim_tipi (public,private) static (olabilir, olmayabilir ) dönüş_tipi(void  
,int vb) foksiyon_ismi(parametreler)  
{  
    // Gövde kısmı  
}
```

Erişim Seviyesi

- Erişim Seviyeleri üç alt başlığa ayrılır: **Public, Private, Protected**

Public:

- Public seviyesinde olan bir methoda, aynı sınıf içerisinde veya dışarıdan erişimde bulunmak mümkündür.
- Bu erişimden kastedilen ise örneğin oluşturulan bir sınıf içerisindeki herhangi bir değişkene farklı bir sınıf üzerinden farklı bir değer yüklenebilir veya var olan değer alınabilir.
- public damgası bir değişkeni, metodu ya da sınıfı niteleyebilir.
- Niteledikleri öğeler **herkes tarafından** kullanılabilir.
- Başka pakette olsa bile, program içindeki her kod onlara erişebilir.
- public damgalı bir sınıfın değişkenlerine ve metotlarına kendi alt-sınıfları ve dışarıdaki başka sınıflar kısıtsız erişebilir.
- public damgalı değişkenler ve metotlar için de kısıtsız erişim vardır.
- Uygulama programlarında main() metodunun public damgalı olmasının nedeni budur.

Erişim Seviyesi

Private

- Private seviyesinde olan bir method içerisindeki herhangi bir öğeye dışarıdan erişimde bulunulamaz.
- Örneğin private olarak belirlenen pi değeri, o sınıf dışındaki herhangi bir başka yerde değiştirilemez.
- Bazı değişken, method ya da sınıflara başka sınıftaki kodların erişmesini engellemek isteyebiliriz.
- Bunun için private nitelmesini kullanırız.
- private damgalı öğelere yalnız aynı sınıftaki kodlar erişebilir, başka sınıftaki kodlar erişemez.
- Kendi alt-sınıfları bile erişemez.
- Bir alt-sınıf, atasının public ve ön-tanımlı öğelerine erişebilir, ama private öğelerine erişemez.
- Onlara erişebilmesi için, super class interface-fonksiyonu kullanılır.

Erişim Seviyesi

Protected

- Public ve Private erişim seviyelerinin ortasında bir erişim seviyesine sahiptir.
- Bir sınıf içindeki değişken ve metotlara alt-sınıfların erişebilmesini, ama paket içindeki ya da program içindeki başka kodların erişmesini engellemek isteyebiliriz.
- Bunun için söz konusu öğeyi, protected damgası ile nitelemek yetecektir.



Static Durumu

- **Static anahtar** kelimesi kullanılarak oluşturulan değişkenler nesne değişkeni değil “sınıf değişkeni” olarak adlandırılırlar.
- Bu değişkenler **nesneye ait değil, sınıfa ait bilgileri taşırlar.**
- Sınıf değişkenleri içinde tanımlandığı sınıftan hiçbir nesne oluşturulmamış olsa bile **bellekte yer kaplarlar.**
- Nesne değişkenleri ise ancak bir nesne tanımlandığında bellekte yer kaplarlar.
- Bu iki tür değişkenin ayrıldığı bir başka nokta da sınıf değişkenlerinin sadece bir örneğinin olmasıdır.
- Yani o sınıftan kaç tane nesne oluşturulursa oluşturulsun, bellekte tek bir tane sınıf değişkeni vardır ve ne şekilde erişirsek erişelim, aynı sınıf değişkenine erişiriz.

Dönüş Tipi

- **Void:** Geri Dönüş değeri almaz.
- İnt, String, Float, Double : **Return** değeri almaktadır.

Parametre

- Parametre, malzeme olarak örneklenebilir.
- Siz metoda malzeme verirsiniz ve o da size onu işleyerek geri döndürür veya döndürmeden kendi işlemlerinde kullanır.
- Bazı metotlar dışarıdan parametreye ihtiyaç duyarken, bazıları duymaz.

Metod Çağırma (Calling Methods)

- Metodları ana mainde **iki şekilde** çağırabiliriz ;

1-Eğer metod static değilse :

- **Sınıfismi class = new Sınıfismi();
class.metod();**

2-Eğer metod static ise :

- **metod();**

Metot Oluştururken Dikkat Edilmesi Gerekenler!!!

- Metot isimleri herhangi bir **sayı ile başlayamaz.**
- Metot isimleri **büyük/küçük harf duyarlılığına** sahiptir.
- Metot isimlerinde **boşluk veya bazı karakterler kullanılamaz.**
- **Main metodu** iki kez kullanılamaz.
- Bir metot içerisinde **başka bir metot oluşturulamaz.**

METOT TÜRLERİ

Parametreye Sahip Olmayan Metotlar

Parametreye Sahip Metotlar

Geriye Değer
Döndüren
Metotlar

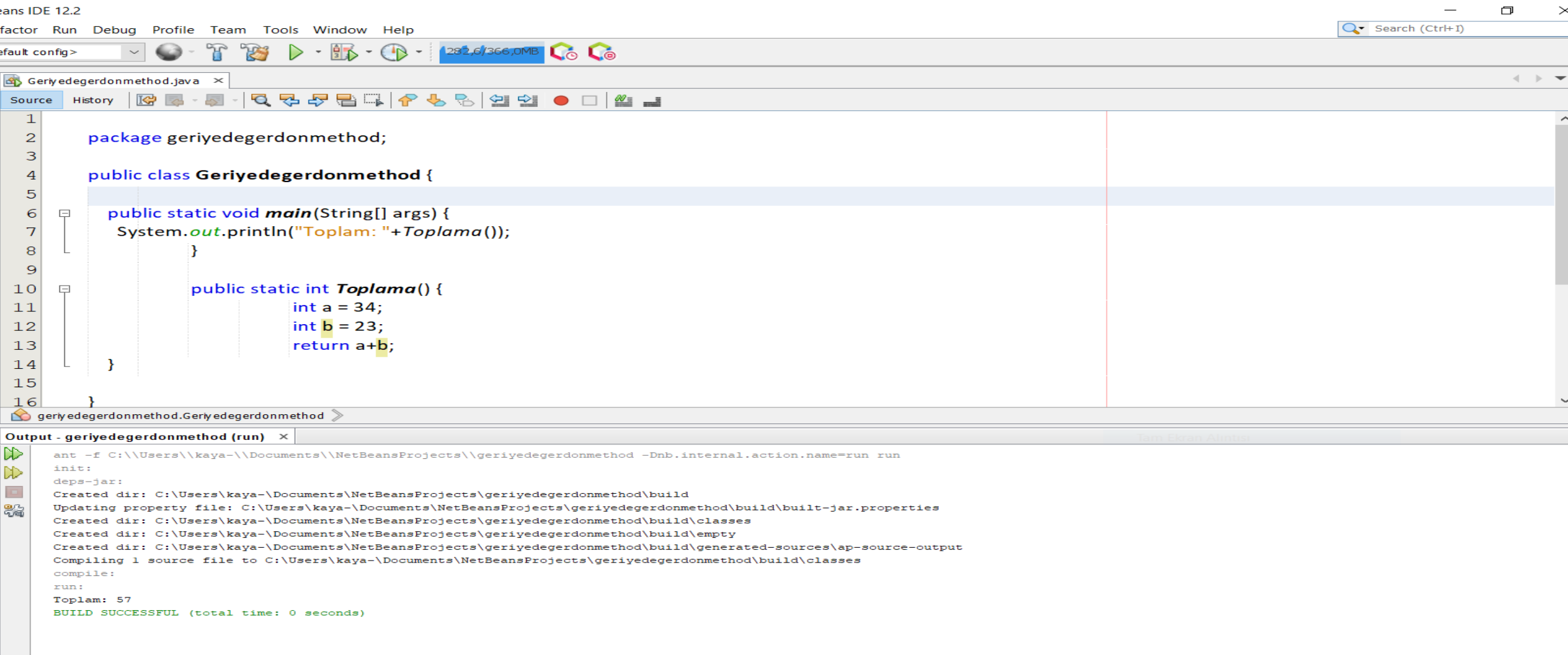
Geriye Değer
Döndürmeyen
Metotlar

Geriye Değer
Döndüren ve
Parametre
Alan Metotlar

Geriye Değer
Döndürmeyen ve
Parametre
Alan Metotlar

A. Parametreye Sahip Olmayan Metotlar

- **A1. Geriye Değer Döndüren Metotlar**
- Çağırıldığı veri tipinde bir değeri geri döndürür. **Return** metotlar olarak bilinirler.



The screenshot displays the NetBeans IDE 12.2 interface. The main editor window shows a Java file named `Geriyedegerdonmethod.java` with the following code:

```
1 package geriyedegerdonmethod;
2
3
4 public class Geriyedegerdonmethod {
5
6     public static void main(String[] args) {
7         System.out.println("Toplam: "+Toplama());
8     }
9
10    public static int Toplama() {
11        int a = 34;
12        int b = 23;
13        return a+b;
14    }
15
16 }
```

The bottom output window, titled `Output - geriyedegerdonmethod (run)`, shows the execution results:

```
ant -f C:\\Users\\kaya-\\Documents\\NetBeansProjects\\geriyedegerdonmethod -Dnb.internal.action.name=run run
init:
deps-jar:
Created dir: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\geriyedegerdonmethod\\build
Updating property file: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\geriyedegerdonmethod\\build\\built-jar.properties
Created dir: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\geriyedegerdonmethod\\build\\classes
Created dir: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\geriyedegerdonmethod\\build\\empty
Created dir: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\geriyedegerdonmethod\\build\\generated-sources\\ap-source-output
Compiling 1 source file to C:\\Users\\kaya-\\Documents\\NetBeansProjects\\geriyedegerdonmethod\\build\\classes
compile:
run:
Toplam: 57
BUILD SUCCESSFUL (total time: 0 seconds)
```

A. Parametreye Sahip Olmayan Metotlar

- **A2. Geriye Değer Döndürmeyen Metotlar**
- Yalnızca bir iş yaptırılır, **void metotlar** olarak da bilinirler.
- **void tanımlı olan metotlar içerisinde return komutu kullanılamaz.**
- Normal bir çıktı mesajı ekrana yazdırılabilir.
- Eğer herhangi bir parametreye sahip değilse parantez içerisi boş bırakılır.

Geriyedegerdondermeyenmeth.java

```
Source History
1 package geriyedegerdondermeyenmeth;
2
3
4 public class Geriyedegerdondermeyenmeth {
5
6     public static void main(String[] args) {
7         ePostam();
8     }
9
10    public static void ePostam() {
11        System.out.println("kaya-alev@outlook.com");
12    }
13
14 }
15
```

Output - parametreyok_geriyedegerdondermeyenmeth (run)

```
ant -f C:\Users\kaya-\Documents\NetBeansProjects\geriyedegerdondermeyenmeth -Dnb.internal.action.name=run run
init:
deps-jar:
Created dir: C:\Users\kaya-\Documents\NetBeansProjects\geriyedegerdondermeyenmeth\build
Updating property file: C:\Users\kaya-\Documents\NetBeansProjects\geriyedegerdondermeyenmeth\build\build-jar.properties
Created dir: C:\Users\kaya-\Documents\NetBeansProjects\geriyedegerdondermeyenmeth\build\classes
Created dir: C:\Users\kaya-\Documents\NetBeansProjects\geriyedegerdondermeyenmeth\build\empty
Created dir: C:\Users\kaya-\Documents\NetBeansProjects\geriyedegerdondermeyenmeth\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Users\kaya-\Documents\NetBeansProjects\geriyedegerdondermeyenmeth\build\classes
compile:
run:
kaya-alev@outlook.com
BUILD SUCCESSFUL (total time: 0 seconds)
```


ÖNEMLİ!!!

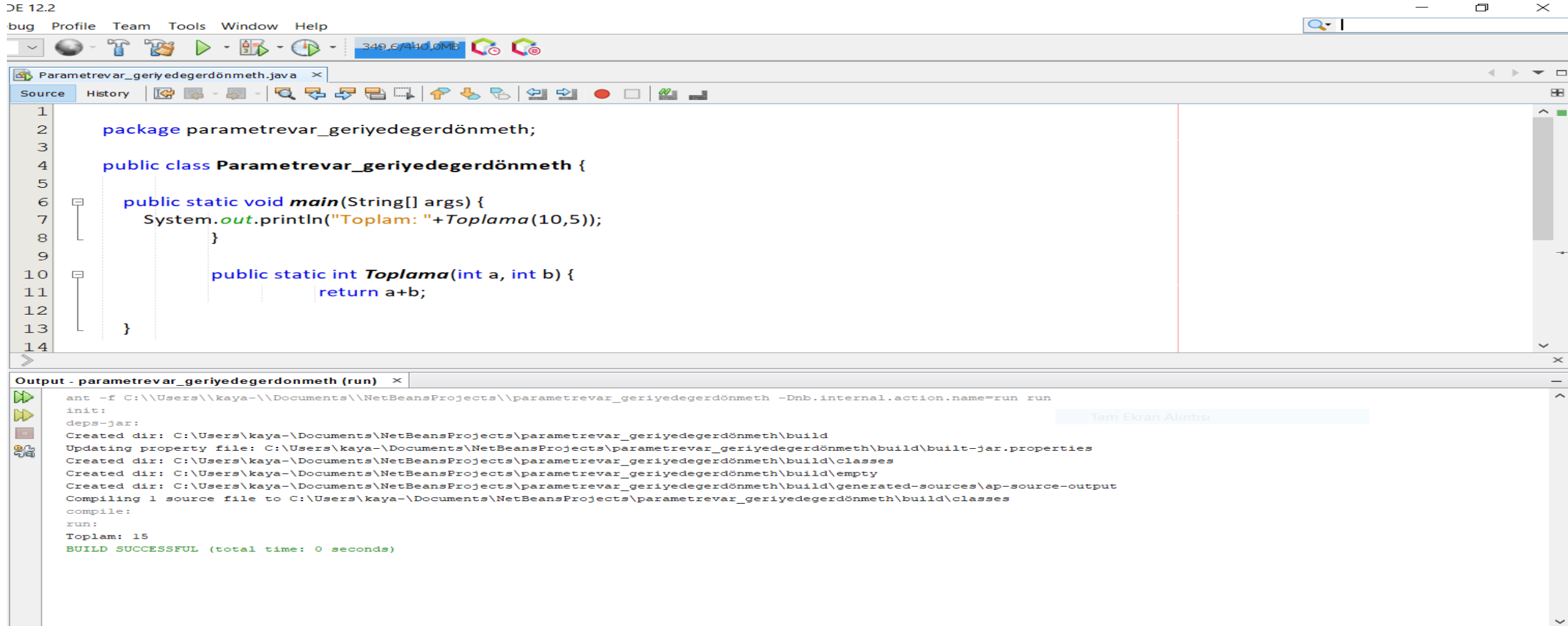
Bir sınıf içerisinde sınırsız sayıda method oluşturulabilir fakat bir method içerisinde farklı bir method oluşturulamaz!



Metot 4,5,6,7 ve 8 ÇALIŞMAZ ve HATA VERİR! Çünkü bir sınıf içerisinde sınırsız method oluşturulabilirken method içerisinde method oluşturulamaz!

B. Parametreye Sahip Metotlar

- Geriye değer döndüren ve değer döndürmeyen metotlar fark etmeden, her metot parametreye veya parametrelere sahip olabilir.
- **B1. Geriye Değer Döndüren ve Parametre Alan Metotlar**



The screenshot displays the NetBeans IDE interface. The top menu bar includes 'bug', 'Profile', 'Team', 'Tools', 'Window', and 'Help'. The toolbar shows various icons for file operations and running. The main editor window is titled 'Parametrevar_geriyedegerdonmeth.java' and contains the following Java code:

```
1 package parametrevar_geriyedegerdonmeth;
2
3
4 public class Parametrevar_geriyedegerdonmeth {
5
6     public static void main(String[] args) {
7         System.out.println("Toplam: "+Toplama(10,5));
8     }
9
10    public static int Toplama(int a, int b) {
11        return a+b;
12    }
13 }
14
```

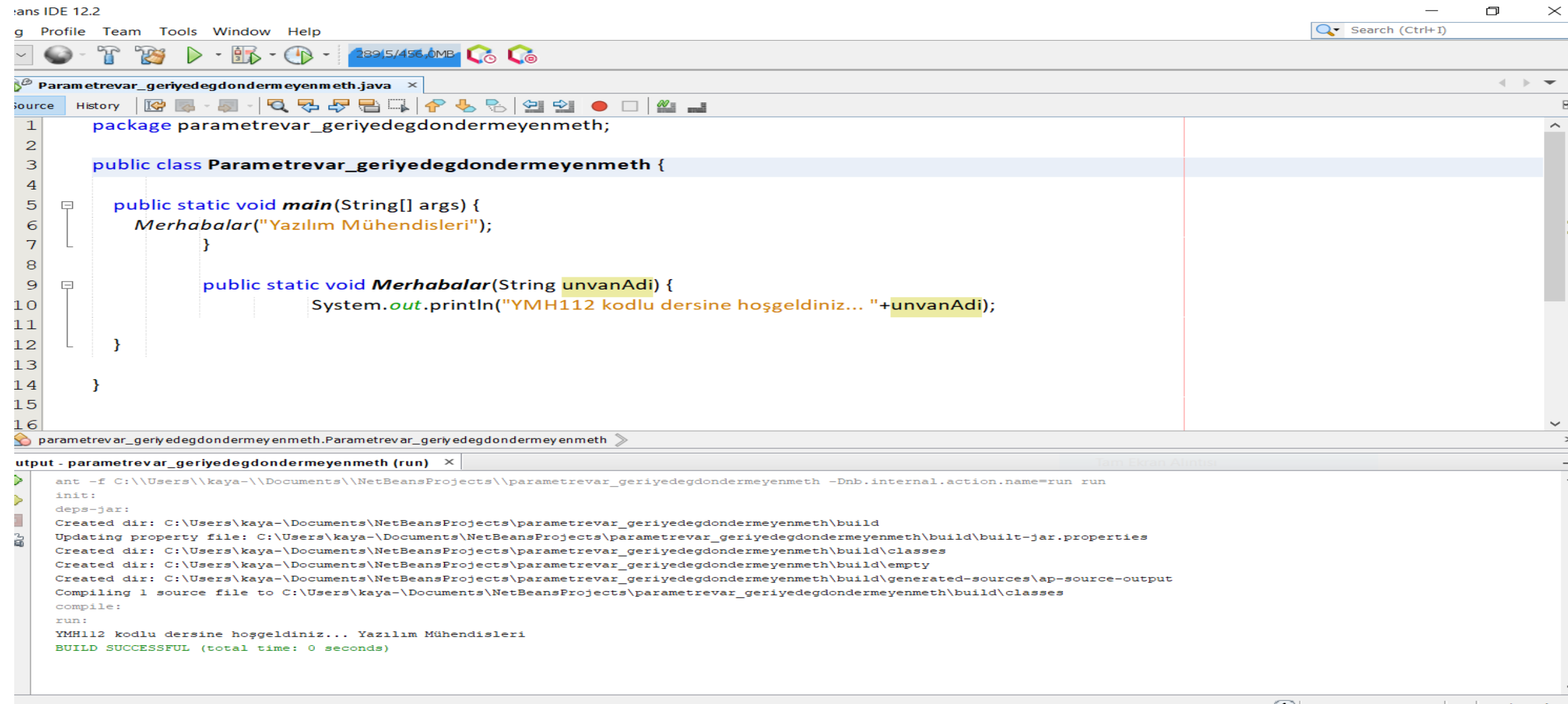
The bottom output window, titled 'Output - parametrevar_geriyedegerdonmeth (run)', shows the execution details:

```
ant -f C:\\Users\\kaya-\\Documents\\NetBeansProjects\\parametrevar_geriyedegerdonmeth -Dnb.internal.action.name=run run
init:
deps-jar:
Created dir: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\parametrevar_geriyedegerdonmeth\\build
Updating property file: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\parametrevar_geriyedegerdonmeth\\build\\built-jar.properties
Created dir: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\parametrevar_geriyedegerdonmeth\\build\\classes
Created dir: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\parametrevar_geriyedegerdonmeth\\build\\empty
Created dir: C:\\Users\\kaya-\\Documents\\NetBeansProjects\\parametrevar_geriyedegerdonmeth\\build\\generated-sources\\ap-source-output
Compiling 1 source file to C:\\Users\\kaya-\\Documents\\NetBeansProjects\\parametrevar_geriyedegerdonmeth\\build\\classes
compile:
run:
Toplam: 15
BUILD SUCCESSFUL (total time: 0 seconds)
```

A button labeled 'Tam Ekran Alıntısı' is visible in the output window.

B. Parametreye Sahip Metotlar

• B2. Geriye Değer Döndürmeyen ve Parametre Alan Metotlar



```
1 package parametrevar_geriyedegdondermeyenmeth;
2
3 public class Parametrevar_geriyedegdondermeyenmeth {
4
5     public static void main(String[] args) {
6         Merhabalar("Yazılım Mühendisleri");
7     }
8
9     public static void Merhabalar(String unvanAdi) {
10        System.out.println("YMH112 kodlu dersine hoşgeldiniz... "+unvanAdi);
11    }
12 }
13
14 }
15
16
```

```
ant -f C:\Users\kaya-\Documents\NetBeansProjects\parametrevar_geriyedegdondermeyenmeth -Dnb.internal.action.name=run run
init:
deps-jar:
Created dir: C:\Users\kaya-\Documents\NetBeansProjects\parametrevar_geriyedegdondermeyenmeth\build
Updating property file: C:\Users\kaya-\Documents\NetBeansProjects\parametrevar_geriyedegdondermeyenmeth\build\build-jar.properties
Created dir: C:\Users\kaya-\Documents\NetBeansProjects\parametrevar_geriyedegdondermeyenmeth\build\classes
Created dir: C:\Users\kaya-\Documents\NetBeansProjects\parametrevar_geriyedegdondermeyenmeth\build\empty
Created dir: C:\Users\kaya-\Documents\NetBeansProjects\parametrevar_geriyedegdondermeyenmeth\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Users\kaya-\Documents\NetBeansProjects\parametrevar_geriyedegdondermeyenmeth\build\classes
compile:
run:
YMH112 kodlu dersine hoşgeldiniz... Yazılım Mühendisleri
BUILD SUCCESSFUL (total time: 0 seconds)
```