

Yordam Çağırmada This Kullanımı

Döngü kullanmadan Aynı fonksiyonu çağırarak değerlerin artmasını sağlayan fonksiyon (*return this*)

```
public class Araba
{
    int ToplamArabaSayisi = ; //classın içerisindeki tüm fonksiyonların kullanabileceği değişken

    Araba ArabaYap() //fonksiyona tip olarak ana classın ismi veriliyor.
    {
        ToplamArabaSayisi++;
        return this; //Fonksiyonun bulunduğu class'a geri dönüş
    }

    void goster()
    {
        System.out.println("Toplam Araba Sayısı = " + ToplamArabaSayisi);
    }

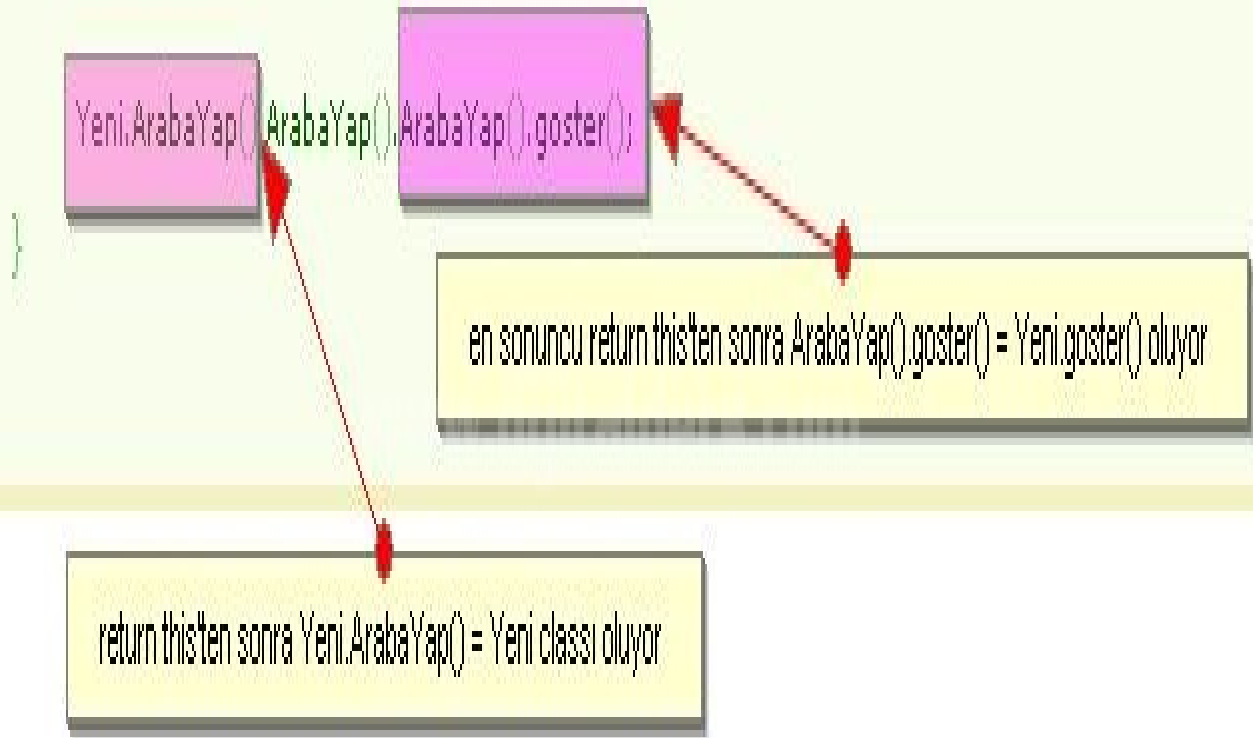
    public static void main(String[] args)
    {
        Araba Yeni = new Araba(); //Araba class'ından türeyen Yeni classı

        Yeni.ArabayaYap().ArabayaYap().ArabayaYap().goster();
        //Yeni.ArabayaYap().ArabayaYap().goster();
        //Yeni.ArabayaYap().goster();
        //Yeni..goster();
    }
}
```

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

```
{
```

```
Araba Yeni = new Araba(); //Araba class'ından türeyen Yeni classı
```



Döngü kullanmadan Aynı fonksiyonu çağırarak fonksiyonların çalışmasını sağlayan başka bir uygulama

```
import java.io.*;
```

```
class Uretici
```

```
{
```

```
int ToplamKisiSayisi=1;
```

```
String GirilenIsimler="";
```

```
String Mesaj;
```

```
Uretici IsimSoran() throws IOException
```

```
{
```

```
String Giris;
```

```
BufferedReader GirilenIsim = new BufferedReader(new
```

```
InputStreamReader(System.in));
```

```
Mesaj = ToplamKisiSayisi + ". isim ";
```

```
System.out.println(Mesaj);
```

```

        Giris = GirilenIsim.readLine();

        this.GirilenIsimler += this.Mesaj + Giris + "\n";

        ToplamKisiSayisi++;

        return this;
    }

    void Yazdirici()
    {
        System.out.println(this.GirilenIsimler);
    }
}

public class Cagri
{

    public static void main(String[] args) throws IOException
    {
        Uretici Ureticiden = new Uretici();

        Ureticiden.IsimSoran().IsimSoran().IsimSoran().Yazdirici();

        //Ureticiden nesnesine ait IsimSoran fonksiyonu her çalıştırıldıktan sonra
        kendini barındıran class'a geri dönüyor
    }
}

```

Bir Yapılandırıcıdan Diğer Bir Yapılandırıcıyı Çağırma

```

public class Tost
{
    int sayi;
    String malzeme;

    public Tost() {
        this(5);
        // this(5,"sucuklu"); !Hata!-iki this kullanılamaz
        System.out.println("parametresiz yapılandırıcı");
    }

    public Tost(int sayi){
        this(sayi, "Sucuklu");
        this.sayi = sayi;
        System.out.println("Tost(int sayi) ");
    }

    public Tost(int sayi, String malzeme){
        this.sayi = sayi;
        this.malzeme = malzeme;
        System.out.println("Tost(int sayi ,String malzeme) ");
    }
}

```

```

public void siparisGoster(){

    // this(5,"Kasarlı"); !Hata!-sadece yapılandırıcılarda kullanılır
    System.out.println("Tost sayisi=" + sayi + "malzeme =" + malzeme);
}

public static void main(String[] args){

    Tost YeniTost = new Tost(); // YeniTost tanımlanır tanımlanmaz ilk olarak
                                //aynı isimdeki içindeki fonksiyonu çağırır.

    YeniTost.siparisGoster();
}
}

```

İşlem adımları :

```

public class Tost
{
    int sayi;
    String malzeme;
    public Tost() {
        this(5);
        // this(5,"sucuklu"); !Hata!-iki this kullanılamaz
        System.out.println("parametresiz yapılandırıcı");
    }
    public Tost(int sayi){
        this(sayi, "Sucuklu");
        this.sayi = sayi;
        System.out.println("Tost(int sayi) ");
    }
    public Tost(int sayi, String malzeme){
        this.sayi = sayi;
        this.malzeme = malzeme;
        System.out.println("Tost(int sayi ,String malzeme) ");
    }
    public void siparisGoster() {
        // this(5,"Kasarlı"); !Hata!-sadece yapılandırıcılarda kullanılır
        System.out.println("Tost sayisi=" + sayi + "malzeme =" + malzeme);
    }
    public static void main(String[] args){
        Tost YeniTost = new Tost(); // YeniTost tanımlanır tanımlanmaz ilk olarak
                                    //aynı isimdeki içindeki fonksiyonu çağırır.
        YeniTost.siparisGoster();
    }
}

```

Ekran Çıktısı

```

Test<int sayi ,String malzeme>
Test<int sayi>
parametresiz yapilandirici
Test sayisi=5malzeme =Sucuklu

```

Bir başka örnek

```

class tugla
{

    int kamyon, ton;
    String tip;

    public tugla()
    {

        this(10);
        System.out.println("Tuglaciya 10 tugla istegi yolladi");

    }

    public tugla(int sayi)
    {

        this(sayi, "kirmizi");
        System.out.println("Tuglaciya kirmizi tugla istegi yolladi");

    }

    public tugla(int sayi, String tip)
    {

        this.kamyon = sayi;
        this.tip = tip;
        System.out.println("Tugla istegi tamamlandi!");

    }

    void kamyonDolusu()
    {

        System.out.println(this.tip + " tipinde tuglalardan " + this.kamyon + " kamyon, kamyon basina " + this.ton + " ton tasıyacak size...nToplam " + (this.kamyon*this.ton)+" ton tugla gelecek");

    }

    tugla tuglaEkle(int ton)
    {

        this.ton += ton;
        return this;

    }

}

class ornek
{

    public static void main(String args[])
    {

```

```
tugla YeniTugla = new tugla();  
YeniTugla.tuglaEkle(10).tuglaEkle(5);  
YeniTugla.kamyonDolusu();  
YeniTugla.tuglaEkle(7);  
YeniTugla.kamyonDolusu();
```

```
}  
}
```

```
C:\Documents and Settings\aidata\Desktop>javac ornek.java
```

```
C:\Documents and Settings\aidata\Desktop>java ornek
```

```
Tugla istegi tamamlandı!
```

```
Tuglaciya kırmızı tugla istegi yolladı
```

```
Tuglaciya 10 tugla istegi yolladı
```

```
kırmızı tipinde tuğlalardan 10 kanyon, kanyon başına 15 ton taşıyacak size...
```

```
Toplam 150 ton tuğla gelecek
```

```
kırmızı tipinde tuğlalardan 10 kanyon, kanyon başına 22 ton taşıyacak size...
```

```
Toplam 220 ton tuğla gelecek
```