



Constructor (Java object'leri nasıl oluşturur ?)

Bana tişört üret

```
public Uret();
```

Bana yeşil tişört üret

```
public Uret("yeşil");
```

Bana yeşil, v yaka tişört üret

```
public Uret("yeşil", "V yaka");
```

Bana yeşil, v yaka, medium tişört üret

```
public Uret("yeşil", "V yaka", "medium");
```

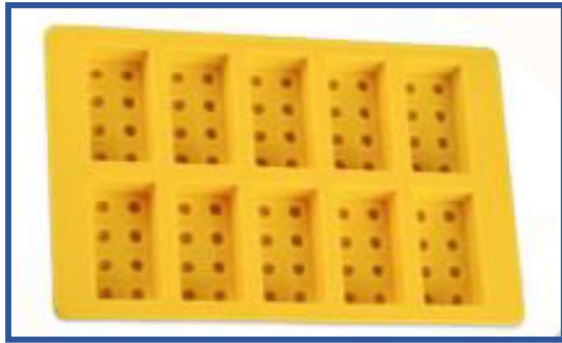
Bana yeşil, v yaka, medium tişörtlerden 100 tane üret

```
public Uret("yeşil", "V yaka", "medium", 100);
```





Constructor (Java object'leri nasıl oluşturur ?)



Class(Object Kalibi)



Object



Birden fazla Obje birleştirilir



Application

Field
(Variables) Method
(Functions)



Constructor (Java object'leri nasıl oluşturur ?)

Java'da Class'lar object üretmek için Constructor kullanılır

Java'da bir Class oluşturduğumuzda, Java object oluşturabilmemiz için default constructor oluşturur.

Default constructor Class içinde görülmez

Kullanıcı yeni bir Constructor oluşturursa Java default constructor'ı siler.

Constructor **nasıl** ve **nerede** oluşturulur ?

- Constructor Class içerisinde oluşturulur.
- Constructor'ın ismi Class ismi aynı olmalıdır, dolayısıyla isim büyük harfle başlar
- Constructor'ların return type'leri olmaz

```
public class MyClass {  
    MyClass() {  
    }  
    public static void main(String args) {  
    }  
}
```



Constructor


Bir Car Class'i oluşturalım.

Arabaları oluştururken özelliklerini yükleyebilmek için variable ve method'lar oluşturabiliriz

Bizim class'imizde

- İlan no
- Marka
- Model
- Yıl
- Fiyat gibi variable'lar ve
- hiz(120) ve yakit(dizel) gibi iki de method olsun.

2020 BMW 3.20i FIRST EDITION M SPORT 900 KM'DE BOYASIZ



647.900 TL

KONYA / KARATAY / FEVZİÇAKMAK MAHALLESİ

| | |
|---------------|----------------------------|
| İlan No: | 16126048 |
| İlan Tarihi: | 24 Şubat 2021 |
| Marka: | BMW |
| Seri: | 3 Serisi |
| Model: | 320i First Edition M Sport |
| Yıl: | 2020 |
| Yakıt Tipi: | Benzin |
| Vites Tipi: | Otomatik |
| Motor Hacmi: | 1597 cc |
| Motor Gücü: | 170 hp |
| Kilometre: | 900 km |
| Boya-değişen: | Tamamı orjinal |
| Takasa Uygun: | Takasa Uygun |
| Kimden: | Galeriden |

Karşılaştırmak Favori Paylaş



Constructor

```
public class Car {  
  
    int ilanNo;  
    String marka;  
    String model;  
    int yil;  
    int fiyat;  
  
    public static void main(String[] args) {  
  
    }  
  
    public void hiz(int maxHiz) {  
        System.out.println("Araba max. " + maxHiz + " km hiz yapar");  
    }  
  
    public void yakit (String yakitTuru) {  
        System.out.println("Araba yakit olarak " + yakitTuru + " kullanir");  
    }  
}
```



Constructor

Baska bir Class'da olusturdugumuz Car class'indan bir object olusturaim ve degerler atayalim.

```
public static void main(String[] args) {  
  
    Car car1=new Car();  
    car1.ilanNo=1001;  
    car1.marka="Toyota";  
    car1.model="Corolla";  
    car1.yil=2010;  
    car1.fiyat=15000;  
  
    System.out.println(car1.ilanNo +"," + car1.marka +"," +car1.model +"," +  
                        car1.yil +"," + car1.fiyat);  
    car1.hiz(150);  
    car1.yakit("Benzin");  
}
```

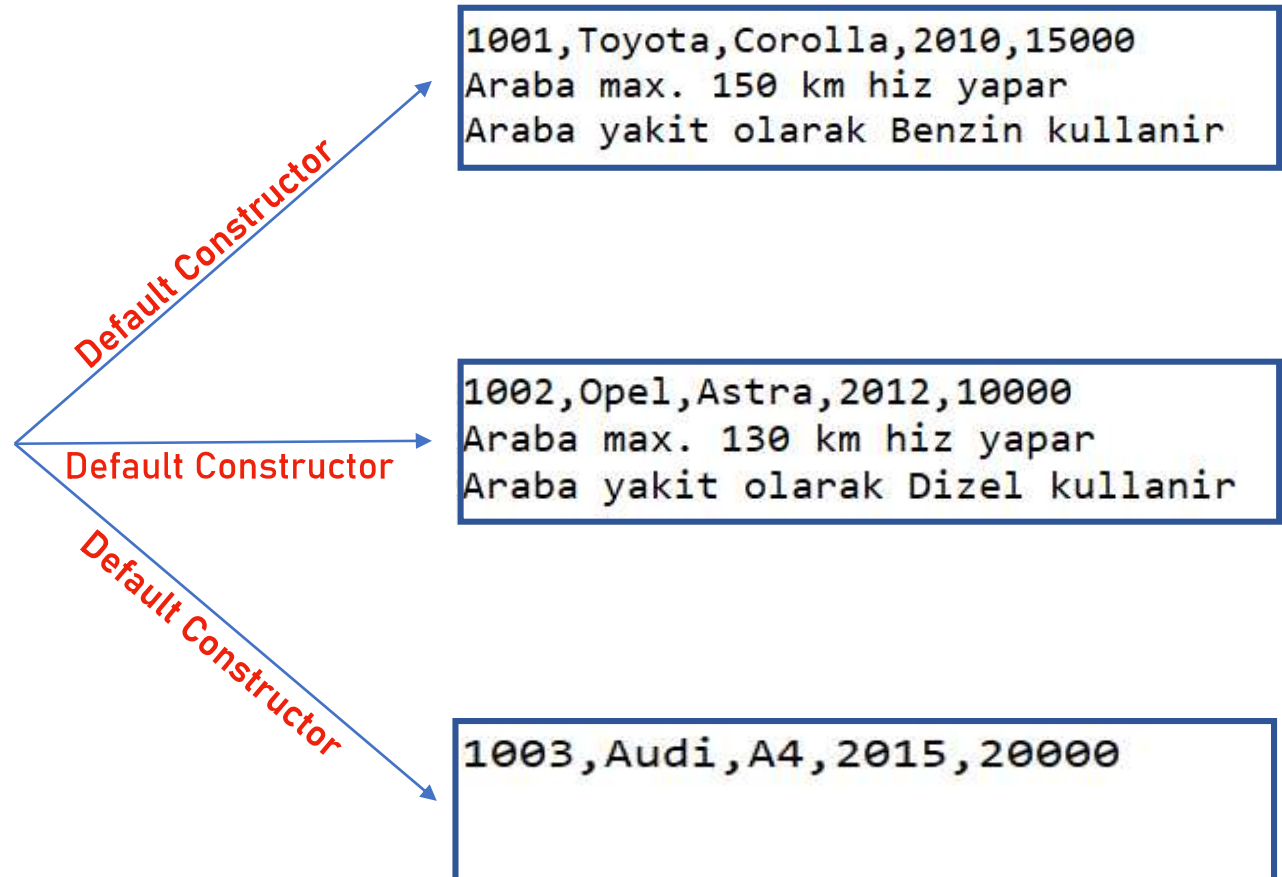
```
1001,Toyota,Corolla,2010,15000  
Araba max. 150 km hiz yapar  
Araba yakit olarak Benzin kullanir
```




Constructor

```
public class Car {  
    int ilanNo;  
    String marka;  
    String model;  
    int yil;  
    int fiyat;  
  
    public static void main(String[] args) {  
  
    }  
  
    public void hiz(int maxHiz) {  
        System.out.println("Araba max. " + maxHiz + " km hiz yapar");  
    }  
  
    public void yakit (String yakitTuru) {  
        System.out.println("Araba yakit olarak " + yakitTuru + " kullanir");  
    }  
}
```

Class icinde gorunur Constructor yok
(Default Constructor) calisiyor





Parametrized (Parametrelili) Constructor

- Default constructor ile olusturdugumuz obje icin default deger disinda deger atamak istersek once objeyi olusturmali, sonra tum ozelliklere tek tek Assignment yapmaliyiz.
- Deger atama islemini obje olustururken yapmak istersek atama yapacagimiz variable'lari iceren constructor olusturabiliriz.
- Bu durumda this ile yolladigimiz degerleri instance variable'lara assign etmeliyiz.

```
public class Car {  
  
    public int ilanNo;  
    public String marka;  
    public String model;  
    public int yil;  
    public int fiyat;  
  
    public Car() {  
    }  
  
    public Car(int ilanNo, String marka, String model, int yil, int fiyat) {  
        this.ilanNo=ilanNo;  
        this.marka=marka;  
        this.model=model;  
        this.yil=yil;  
        this.fiyat=fiyat;  
    }  
  
    public static void main(String[] args) {  
  
    }  
}
```




Constructor

Class icinde birden fazla Constructor olursa ...

Java obje create ederken kullandigimiz argument'lere gore uygun constructor'i kullanir

```
public class Car {  
  
    public int ilanNo;  
    public String marka;  
    public String model;  
    public int yil;  
    public int fiyat;  
  
1  public Car() {  
    }  
2  public Car(int ilanNo) {  
    this.ilanNo=ilanNo;  
    }  
3  public Car(int ilanNo,String marka, String model,  
    int yil, int fiyat) {  
    this.ilanNo=ilanNo;  
    this.marka=marka;  
    this.model=model;  
    this.yil=yil;  
    this.fiyat=fiyat;  
    }  
  
    public static void main(String[] args) {  
  
    }  
}
```

Car **car1** = new Car();
Default degerlere sahip olur

Car **car24** = new Car(1024);
Ilan no 1024 olur,
diger variable'lar Default degerlere sahip olur

Car **car84** = new Car(1834, opel, astra, 2020, 15000);
Argument olarak girilen degerlere sahip olur



Constructor

Asagidaki class calistirildiginda output ne olur ?

```
public class Student {  
  
    String name = "Emily";  
    int age = 20;  
  
    Student(String name, int age) {  
        this.name = name;  
        this.age = 22;  
    }  
  
    public static void main(String args) {  
  
        Student st = new Student("Oliver", 21);  
  
        System.out.print(st.name);  
  
        System.out.print(", " + st.age);  
    }  
}
```



Constructor

Asagidaki class calistirildiginda output ne olur ?

```
public class Student {  
    String name;  
    int age;  
    String phone;  
  
    Student() {  
  
    }  
    Student(String name, int age, String phone) {  
        this.phone = phone;  
        this.name = name;  
    }  
  
    public static void main(String[] args) {  
        Student s1 = new Student();  
  
        Student s2 = new Student("John",25,"029-998877");  
  
        System.out.print(s2.name + ", " + s2.age + ", " + s2.phone);  
    }  
}
```



Constructor Call

- Obje olusturulurken constructor calisir.
- Bazi durumlarda constructor icinden baska bir constructor cagirmamiz gerekebilir
- Bir constructor'un icinden diger bir constructor'i cagirmek icin

this(parametreler); kullanilir.

Soru : Yandaki class calistirildiginda output ne olur ?

this(parametreler); kullanirken

Kural 1 : **this**(parametre); cagrildigi constructor'in ilk satirinda yazilmalidir

Kural 2: Kural 1'den dolayi bir constructor icinde sadece 1 constructor cagrilabilir

```
public class MyConstructor {  
  
    int x =5;  
  
    MyConstructor(){  
        System.out.println("-x" + x );  
    }  
  
    MyConstructor(int x){  
        this();  
  
        System.out.println("-x" + x );  
    }  
  
    public static void main(String[] args) {  
        MyConstructor mc1 = new MyConstructor(4);  
  
        MyConstructor mc2 = new MyConstructor();  
    }  
}
```



Constructor Call

Yandaki class calistirildiginda output ne olur ?

```
public class MyConstructor {  
  
    int x =3;  
    int y =5;  
  
    MyConstructor(){  
        x+=1;  
        System.out.println("-x" + x );  
    }  
  
    MyConstructor(int i){  
        this();  
  
        this.y= i;  
        x += y;  
        System.out.println("-x" + x );  
    }  
    MyConstructor(int i , int i2){  
        this(3);  
  
        this.x -= 4 ;  
        System.out.println("-x" + x );  
    }  
  
    public static void main(String[] args) {  
        MyConstructor mc1 = new MyConstructor(4,3);  
    }  
}
```



Constructor Call

Yandaki class calistirildiginda output ne olur ?

Which are true of the following code? (Choose all that apply)

```
1: public class Rope {  
2:     public static void swing() {  
3:         System.out.print("swing ");  
4:     }  
5:     public void climb() {  
6:         System.out.println("climb ");  
7:     }  
8:     public static void play() {  
9:         swing();  
10:        climb();  
11:    }  
12:    public static void main(String[] args) {  
13:        Rope rope = new Rope();  
14:        rope.play();  
15:        Rope rope2 = null;  
16:        rope2.play();  
17:    }  
18: }
```

- A. The code compiles as is.
- B. There is exactly one compiler error in the code.
- C. There are exactly two compiler errors in the code.
- D. If the lines with compiler errors are removed, the output is climb climb.
- E. If the lines with compiler errors are removed, the output is swing swing.
- F. If the lines with compile errors are removed, the code throws a NullPointerException.



Constructor Call

```
public class MyClass{
    int num1;
    String name = "Ali";

    MyClass(){
        char letter = 'c';
    }

    MyClass (int num1){
        this();
        this.num1 = num1;
    }

    void MyClass(){
        num1++;
    }

    increase(int num1){ name++;
    } }
```

Bosluklari "True" veya "False" olarak doldurun.

- 1) Turquoise'lar instance variable'lardir....
- 2) Orange un-parameterized constructor'dir.
- 3) Pembe parameterized constructor'dir.
- 4) Yesil un-parameterized constructor'dir.
- 5) Mavi parameterized constructor'dir.
- 6) "letter" local variable'dir
- 7) Instance variable'lara mutlaka atama yapılmalıdır
- 8) Verilen code block'da , sadece 1 tane compile time error vardır.
- 9) "this" keyword, instance variable'lar ile iliskilidir.
- 10) this() yesil MyClass()'i cagirir