

# NESNEYE YÖNELİK PROGRAMLAMA

---

*Java Dilinin Syntax'ı*

*Emir Öztürk*

# ÖRNEK JAVA PROGRAMI

---

```
package net.emirozturk;

public class Main {

    /*
     Java main metodu
     Metod, args parametresi ile konsol parametrelerini alır ve geriye değer döndürmez.
     */

    public static void main(String[] args) {
        System.out.print("Deger"); //Burada ekrana Deger yazılıyor.
    }
}
```

*Paket Bilgisi*

*Sınıf Tanımı*

*Açıklama Bloğu*

*main isimli Metot*

*Ekrana yazdırma metodu*

*Açıklama Satırı*

# VERİ TÜRLERİ

---

- Temel (primitif) veri türleri

- Tamsayı

- byte



- short



- int



- long



# VERİ TÜRLERİ

---

## ➤ Primitifler (Stack)

### ➤ Ondalıklı Sayı

➤ float 

➤ double 

### ➤ Karakter

➤ char 

### ➤ Doğru / Yanlış

➤ Boolean 

but its "size" isn't something that's precisely defined.\*

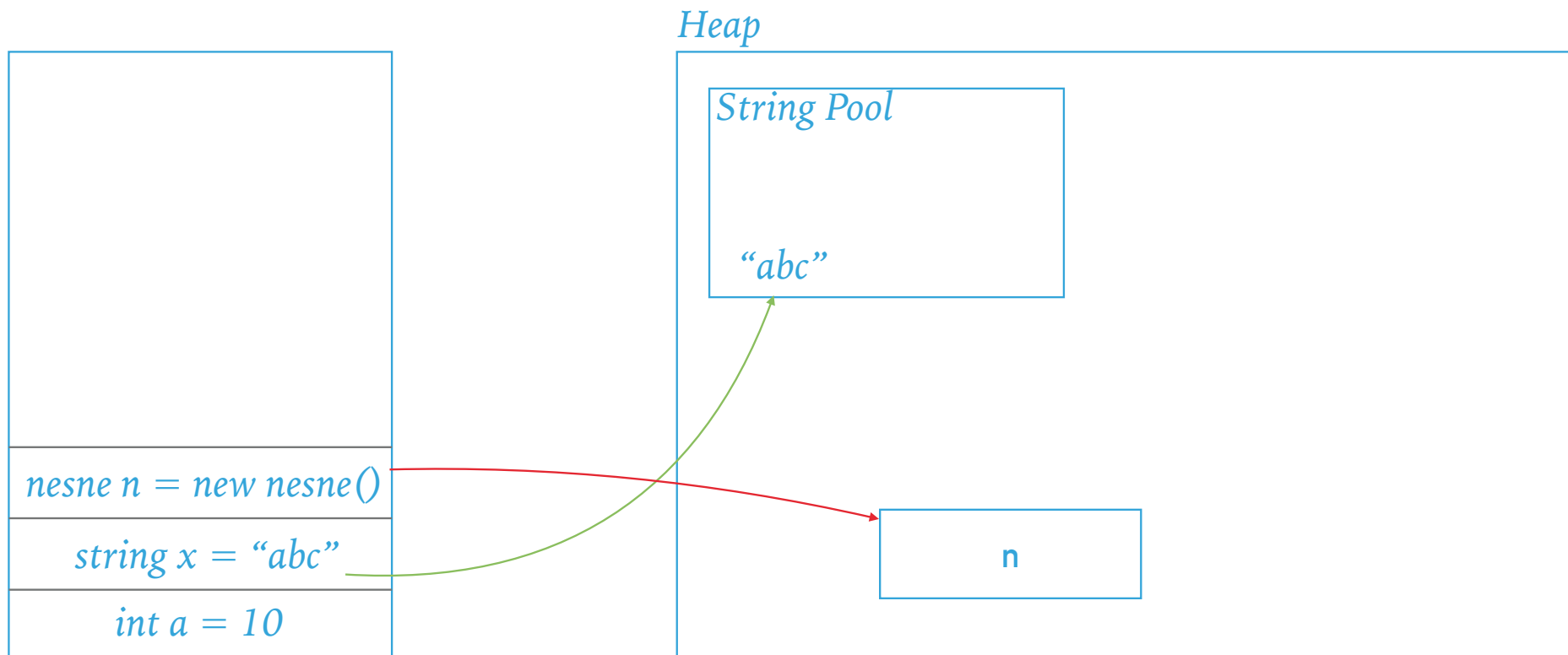
➤ true / false (!= 0/1\*\*\*\*)

\*<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

# VERİ TÜRLERİ

---

## ► Sınıflar (Heap)



# DEĞİŞKEN İSİM KURALLARI

---

- Büyük - küçük harf duyarlı
  - `sayi != SAYI`
- Unicode
- Boşluksuz
  - ~~`degisken-adi`~~
- Karakter, harf, \$ veya \_
- Ayrılmış kelimeler olmamalı
  - ~~`var, int, char, byte, for, while`~~

# DEĞİŞKEN İSİM KURALLARI (SEÇİMLİK)

---

- Tek kelime değişkenler küçük harf
  - `int` degisken;
- Birden fazla kelime içeren değişkenlerde ilk harf küçük, her kelimenin baş harfi büyük.
  - `int` uzunBirDegiskenAdi;
- Sabit bir değer büyük harflerle ve `_` karakteri ile kelimelerin ayrıldığı şekilde.
  - `final static int SABIT_DEGER = 3;`

# OPERATÖRLER

---

Aritmetik	+	-	*	/	%
Karşılaştırma	>	<	>=	<=	== !=
Atama	=	+=	-=	*=	/=
Tekli	++	--			
Mantıksal	&&		!		



# TİP DÖNÜŞÜMLERİ

---

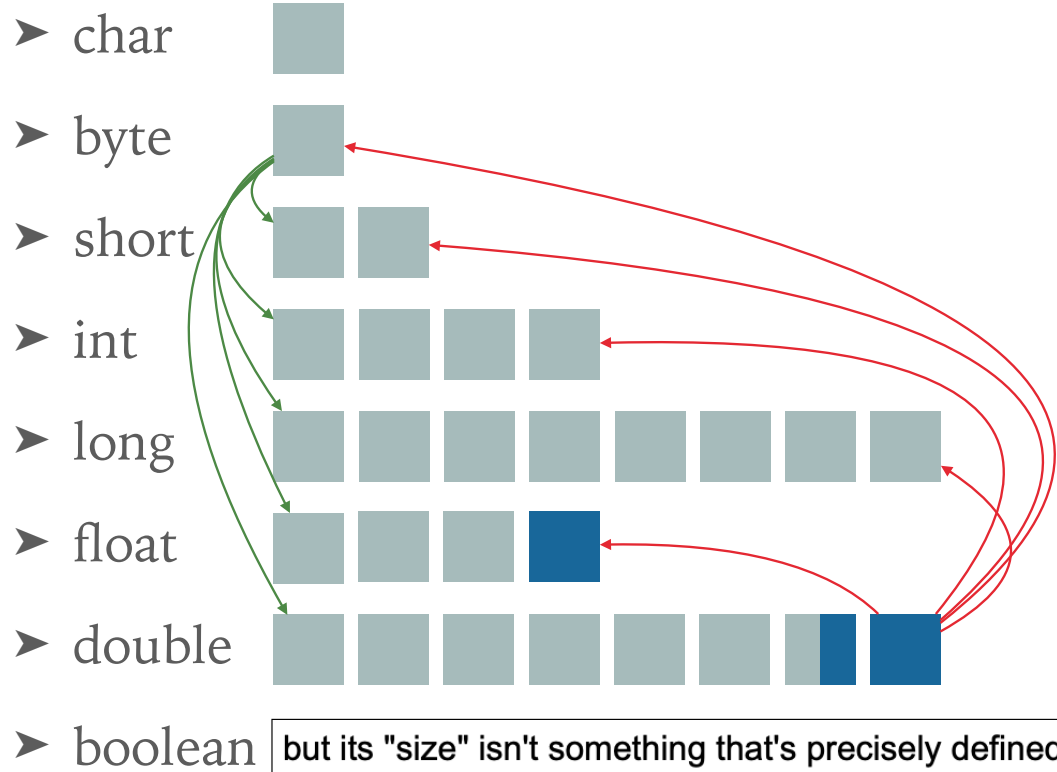
```
package net.emirozturk;
public class Main {
    public static void main(String[] args) {
        String deger = System.console().readLine();
        int sayiDegeri = Integer.parseInt(deger);

        double x = 3.46334;
        int tamsayiDeger = (int)x;
    }
}
```

→ *String'den int'e*

→ *double'dan int'e*

# TİP DÖNÜŞÜMLERİ



Soldan/ Sağa	byte	short	int	long	float	double	char	boolean
byte		O	O	O	O	O	D	X
short	D		O	O	O	O	D	X
int	D	D		O	O*	O	D	X
long	D	D	D		O*	O*	D	X
float	D	D	D	D		O	D	X
double	D	D	D	D	D		D	X
char	D	D	O	O	O	O		X
boolean	X	X	X	X	X	X	X	

*D: Dönüştürme gerekli*

*O: Otomatik*

*X: Dönüştürülemez*

# SYSTEM.OUT.PRINT()/PRINTLN()

---

```
package net.emirozturk;
public class Main {
    public static void main(String[] args) {
        System.out.print("deneme");
        System.out.print(3);
        System.out.print('a');
        System.out.print(3.5);
        System.out.println("deneme");
        //...
    }
}
```

→ Ekrana deneme yazdırır

→ Ekrana 3 yazdırır

→ print farklı parametreler ile kullanılabilir

→ Ekrana 3.5 yazdırır

→ Ekrana deneme yazdırır ve bir alt satıra geçer

→ println de print gibi farklı parametreler ile kullanılabilir

# SCANNER

---

```
package net.emirozturk;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int a = in.nextInt();
        String s = in.nextLine();
        float b = in.nextFloat();
    }
}
```

→ *Scanner nesnesi tanımı*

→ *Sıradaki int değeri okur*

→ *Sıradaki satırı okur*

→ *Sıradaki float değeri okur*

# SYSTEM.CONSOLE.READLINE()

---

```
package net.emirozturk;  
public class Main {  
    public static void main(String[] args) {  
        String deger = System.console().readLine();  
    }  
}
```

*String olarak girilen  
değeri konsoldan okur*

*İzin verilen herhangi bir veri  
tipine dönüştürülebilir.*

# KOŞULLAR

---

```
package net.emirozturk;
public class Main {
    public static void main(String[] args) {
        double x = 3.46334;
        if(x > 3.1){
            //Koşul sağlandığında
        }
        else if(x>5){
            //İlk koşul sağlanmayıp diğeri sağlandığında
        }
        else{
            //Hiçbir koşul sağlanmazsa
        }
    }
}
```

*Koşul parantez içerisinde olmalı*

*else if en az bir koşul içermeli*

*else'ten sonra bir koşul veya parantez bulunmamalı*

# SWITCH İFADESİ – YENİ SWITCH

---

```
package net.emirozturk;
public class Main {
    public static void main(String[] args) {
        int x = Integer.parseInt(System.console().readLine());
        int y = 0;
        switch (x){
            case 1:
                y = 2;
                break;
            case 2:
                y = 2;
                break;
            default:
                y = 8;
                break;
        }
    }
}
```



```
package net.emirozturk;
public class Main {
    public static void main(String[] args) {
        int x = Integer.parseInt(System.console().readLine());
        int y = switch(x){
            case 1,2 -> 2;
            default -> 8;
        };
    }
}
```

# DİZİLER – LİSTELER

```
package net.emirozturk;

import java.util.ArrayList;
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        String[] dizi = {"Eleman1","Eleman2","Eleman3"};
        String[] dizi2 = new String[3];
        dizi2[0] = "Eleman1";
        dizi2[1] = "Eleman2";
        dizi2[2] = "Eleman3";

        ArrayList<String> liste = new ArrayList<String>(Arrays.asList(dizi));

        ArrayList<String> liste2 = new ArrayList<String>();
        liste2.add("Eleman1");
        liste2.add("Eleman2");
        liste2.add("Eleman3");
    }
}
```

→ ArrayList kullanabilmek için  
→ Dizi elemanlarını Arrayliste toplu bir şekilde atabilmek için

→ İlk elemanları verilmiş dizi tanımı  
→ Üstteki tanım ile aynı dizi oluşur. Dizi elemanları tanım sırasında verilmediği için dizi boyutu verilmelidir

→ Tanımlanmış bir diziden liste oluşturma

→ Sıfır elemanlı liste oluşturma

→ Listeye eleman ekleme. Bir üstteki liste ile aynı elemanlara sahip olur



# DÖNGÜLER

---

```
package net.emirozturk;
public class Main {
    public static void main(String[] args) {
        String[] dizi = {"Eleman1","Eleman2","Eleman3"};

        for(int i=0;i<dizi.length;i++){
            System.out.println(dizi[i]);
        }

        for(String eleman : dizi){
            System.out.println(eleman);
        }
    }
}
```

*Bir i değişkeni dizinin uzunluğuna kadar ilerler. Başlangıç;durma koşulu;artış şeklinde tanımlanır.*

*Dizi elemanına erişmek için [i] indisi kullanılır.*

*String türünde bir eleman dizi üzerinde sırayla gezer.*

*Eleman değişkeni dizinin sıradaki değerini aldığı için indis kullanılmaz.*

# DÖNGÜLER

```
package net.emirozturk;
public class Main {
    public static void main(String[] args) {
        String[] dizi = {"Eleman1","Eleman2","Eleman3"};

        int i=0;
        while(i<dizi.length){
            System.out.println(dizi[i]);
            i++;
        }

        i = 0;
        do{
            System.out.println(dizi[i]);
            i++;
        }while(i<dizi.length);
    }
}
```

*while içerisindeki koşul sağlandığı sürece döngü devam eder.*

*Döngü bir indise bağlı ise indis değerinin döngü içerisinde değiştirilmesi gerekmektedir.*

*while döngüsünden farklı olarak do while döngüsü bir defa çalışır ve koşul kontrolü bundan sonra yapılır.*

# FONKSİYONLAR

---

```
package net.emirozturk;

import java.util.ArrayList;
import java.util.Arrays;

public class Main {
    public static String fonk(ArrayList<String> parametre1, int parametre2){
        return parametre1.get(parametre2);
    }

    public static void main(String[] args) {
        String[] dizi = {"Eleman1","Eleman2","Eleman3"};
        ArrayList<String> liste = new ArrayList<String>(Arrays.asList(dizi));
        System.out.println(fonk(liste,2));
    }
}
```

*Fonksiyonda erişim belirleyicisi, dönüş türü isim ve parametreler belirtilmelidir.*

*Parametreler temel türlerden olmak zorunda değildir.*

*Bir fonksiyon değer döndürüyorsa, döndürdüğü değeri parametre alan başka bir fonksiyon içerisinde çağırılabilir.*

- 
- Palindromik metin kontrolü
  - Sayının  $2^n$  olup olmadığının kontrolü
  - Verilen bir stringin tersini döndüren fonksiyon
  - Bir sayının rakamlarının toplamı
  - $n$  değeri alan ve  $n + n^2 + n^3$  değerini döndüren fonksiyon