



TECHPROED

PROFESSIONAL TECHNOLOGY EDUCATION

WELCOME TO TECHPROED JAVA TUTORIAL

Testi baslatmak icin asagidaki adimlari takip ediniz

Go to www.socrative.com

Click on *Login*

Click on *Student Login*

Room Name: *ALPTEKIN3523*

Kayıtta kullandığınız ismi tam olarak yazınız

Time: 11 Minutes

Static Block

- 1) Static block static variable'lara deger ataması yapmak için kullanılır.
- 2) Static block, class ilk çalıştırılmaya başlandığında çalışır ve variable'lara değer ataması yapar.

```
class Test {
    static int i;
    int j;
    static {
        i = 10;
        System.out.println("static block called ");
    }
    Test() {
        System.out.println("Constructor called");
    }
}

class Main {
    public static void main(String args[]) {

        // Although we have two objects, static block is executed only once.
        Test t1 = new Test();
        Test t2 = new Test();
    }
}
```

- 3) Static block'lar constructorlardan, tüm methodlardan ve main methodtan önce çalışır.
- 4) Eğer 1'den fazla static block varsa üstteki önce çalışır.

output nedir?

```
class Test1 {  
public  
    static void main(String[] args)  
    {  
        int x = 20;  
        System.out.println(x);  
    }  
    static  
    {  
        int x = 10;  
        System.out.print(x + " ");  
    }  
}
```



output nedir?

```
class Test1 {  
    int x = 10;  
public  
    static void main(String[] args)  
    {  
        System.out.println(x);  
    }  
    static  
    {  
        System.out.print(x + " ");  
    }  
}
```



output nedir?

```
class Test1 {  
    int x = 10;  
public  
    static void main(String[] args)  
    {  
        Test1 t1 = new Test1();  
        System.out.println(t1.x);  
    }  
    static  
    {  
        int x = 20;  
        System.out.print(x + " ");  
    }  
}
```

output nedir?

```
class Test1 {  
    int x = 10;  
public  
    static void main(String[] args)  
    {  
        System.out.println(Test1.x);  
    }  
    static  
    {  
        int x = 20;  
        System.out.print(x + " ");  
    }  
}
```


Arrays

Arrays birden fazla variable depolamak için kullanılabilen object'lerdir.

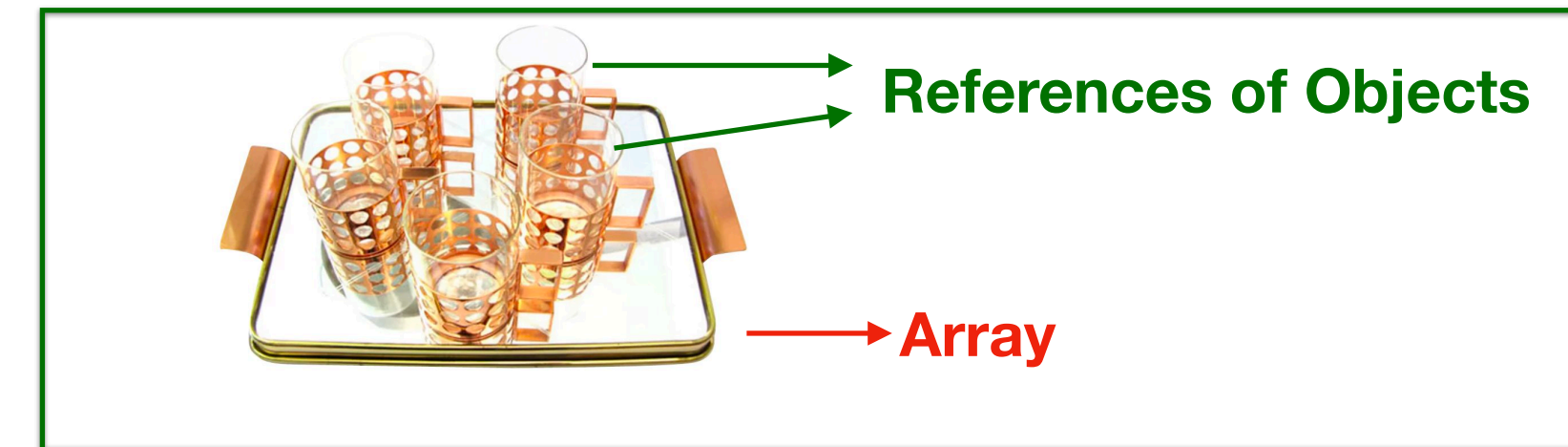
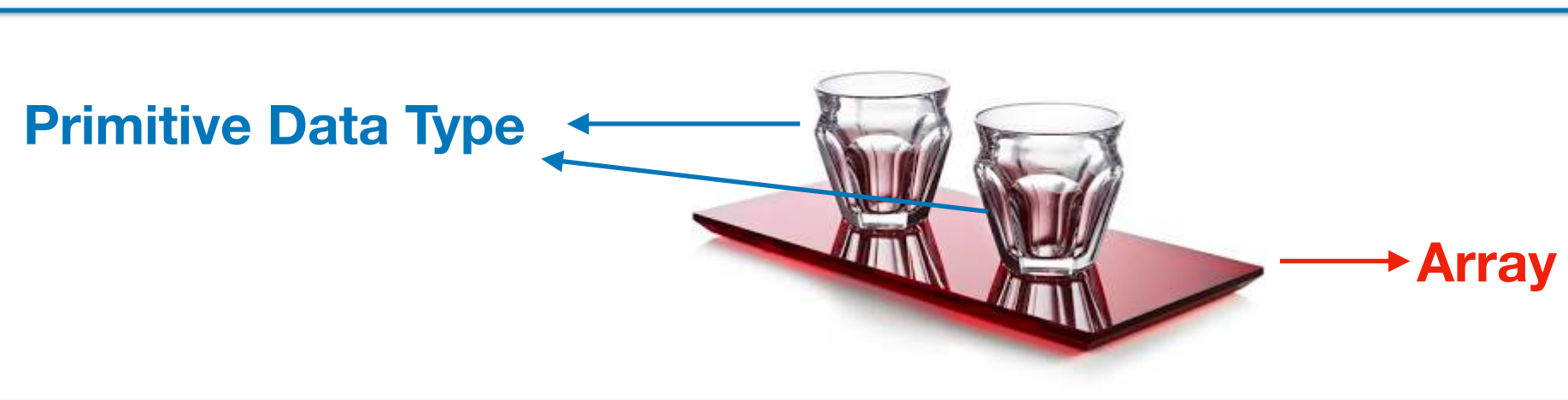
Kural 1:

Arrays sadece

1) Primitive data type

2) Object Reference

içerebilir.



Kural 2:

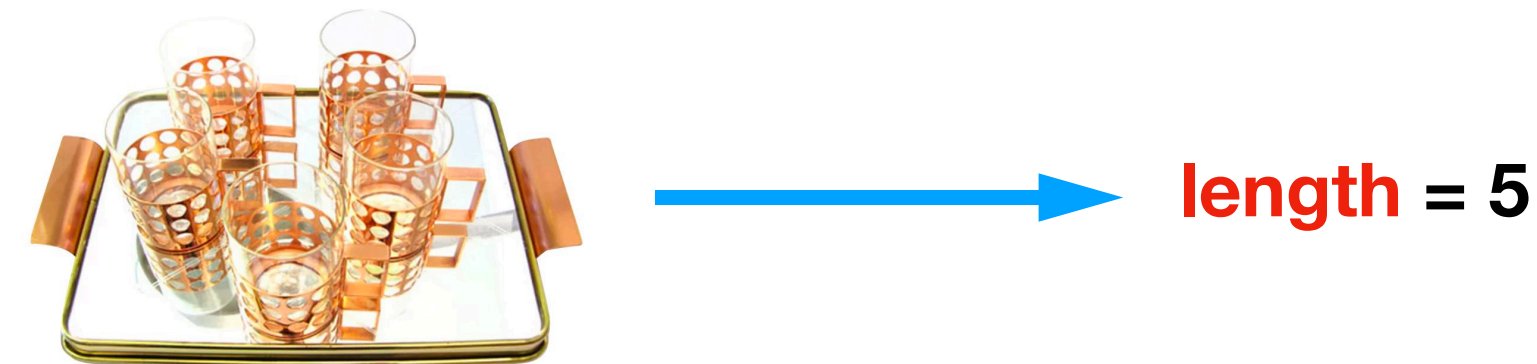
Arrays içindeki tüm variable'lar aynı data type'ında olmalıdır.

Kural 3:

Bir Array oluşturmada önce o Array'in içine kaç variable koyacağımıza karar vermeliyiz.

Kural 4:

Bir Array içine koyabileceğimiz variable sayısına o Array'in "length" i denir. O Array içine length'den fazla variable koyamayız.

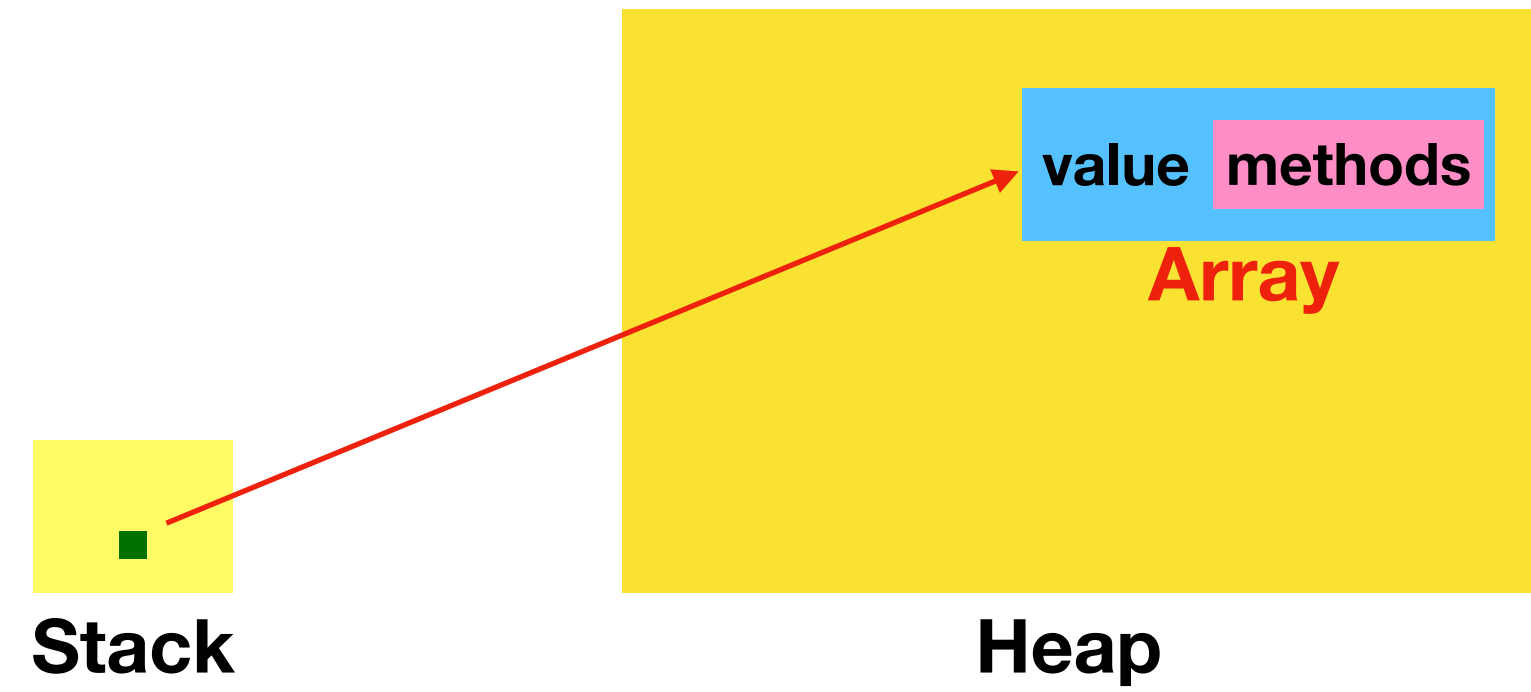


Kural 5:

Array'ler object'tir bu yüzden Heap Memory'de depolanırlar.

Kural 6:

Array'ler object'tir bu yüzden value'ların yanında methodlar da vardır.



Kural 7:

Array'ler object'tir bu yüzden runtime'a oluşturulurlar.

Bir Array nasıl declare edilir?

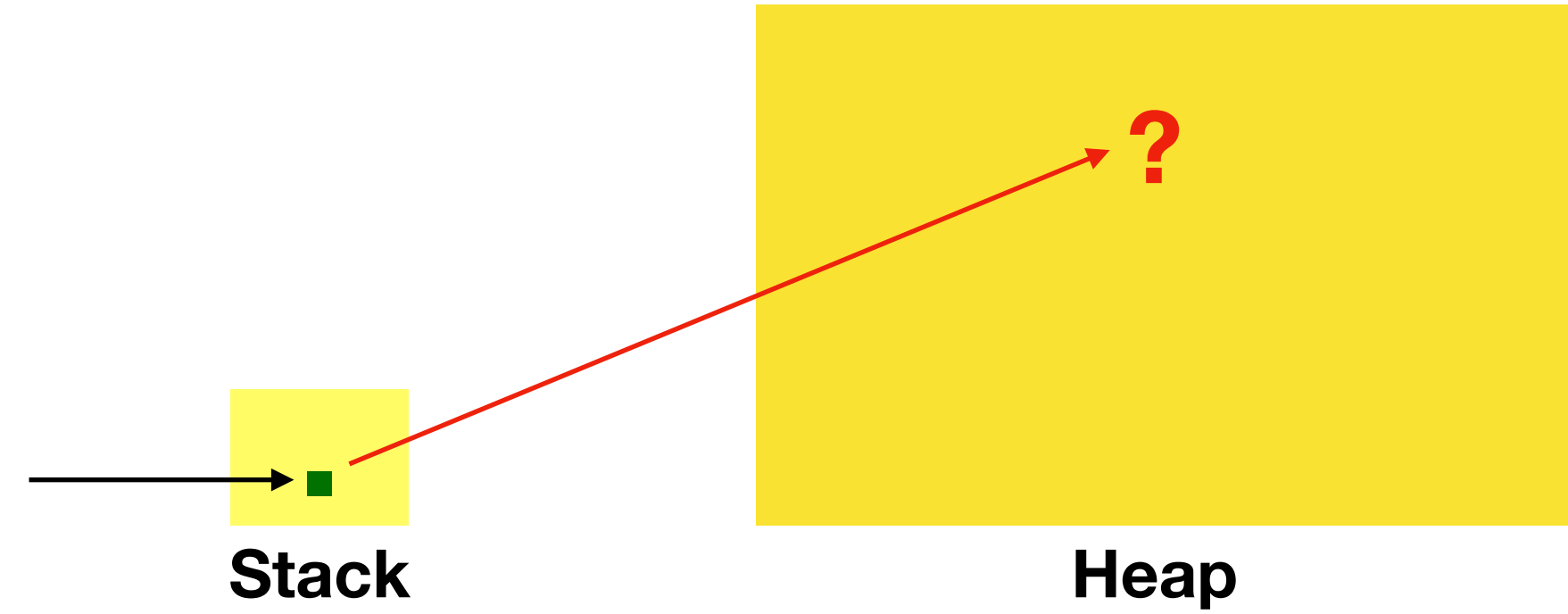
Kural 7:

Array oluşturmak için iki yol vardır :

- 1) `int myArray[];` //Bu daha çok kullanılır
- 2) `int []myArray;`

Kural 8:

Array oluşturulduktan sonra, stack memory'de **reference** oluşturulur; ama Array henüz oluşturulmamıştır.



Bir Array nasıl oluşturulur ?

Kural 9:

```
int myArray[ ] = new int[6];
```

1) Yukarıdaki kod **length**'i 6 olan bir array oluşturur.

2) biz array'e eleman eklemesek Java elemanlar için data type'ına uygun default degerler atar.

Eğer yukarıdaki array'i yazdırırsanız ekranda {0, 0, 0, 0, 0, 0} görürsünüz

Kural 10:

Not: Array oluştururken length'i yazmazsanız compile time error alırsınız.

Bir array'i toString() ile ekrana yazdırmak

```
public static void main(String aargs[]) {  
    int arr[] = {1, 2, 3, 4, 5};  
    System.out.println(Arrays.toString(arr));  
}
```

[1, 2, 3, 4, 5]

output nedir?

```
public static void main(String args[]) {  
    int arr[] = {10, 20, 30, 40, 50};  
    for(int i=0; i < arr.length; i++)  
    {  
        System.out.print(" " + arr[i]);  
    }  
}
```

output nedir?

```
public static void main(String args[]) {  
    int arr[] = new int[2];  
    System.out.println(arr[0]);  
    System.out.println(arr[1]);  
}
```

Bir Array'e Eleman Ekleme

Kural 11:

```
int myArray[] = new int[3];
```

```
myArray[0] = 9;
```

```
myArray[1] = 10;
```

```
myArray[2] = 11;
```

Veya

```
int myArray[] = {9, 10, 11};
```

Bir Array'in elemanlarına Ulaşmak

Kural 12:

```
int myArray[] = {9, 10, 11};
```

Array elemanlarına index'ler kullanılarak ulaşılır.

myArray[0] ==> 9,

myArray[1] ==> 10,

myArray[2] ==> 11,

Kural 13:

“**n**” array'in length'i olmak üzere myArray[**n-1**] son elemanı gösterir

Kural 14:

Bir Array'de olmayan index'i kullanmak isterseniz
“**ArrayIndexOutOfBoundsException**” alırsınız.

Bir Array'in Uzunluğu

Kural 15:

```
int myArray[] = {9, 10, 11};  
int size = myArray.length;
```

Kural 16:

Strings ==> length()
Arrays ==> length

Bir Array'in Tum Elemanlarını Console'a Yazdırmak

Kural 17:

```
int myArray[] = {9, 10, 11};  
int size = myArray.length;  
  
for(int i=0; i<size; i++) {  
    System.out.println(myArray[i]);  
}
```

Output Nedir?

1)

```
public class Main {  
    public static void main(String args[]) {  
        int arr[] = {10, 20, 30, 40, 50};  
        for(int i=0; i < arr.length; i++)  
        {  
            System.out.print(" " + arr[i]);  
        }  
    }  
}
```

2)

```
class Test {  
    public static void main(String args[]) {  
        int arr[2];  
        System.out.println(arr[0]);  
        System.out.println(arr[1]);  
    }  
}
```

3)

```
class Test {  
    public static void main(String args[]) {  
        int arr[] = new int[2];  
        System.out.println(arr[0]);  
        System.out.println(arr[1]);  
    }  
}
```

4)

```
class Test
{
    public static void main (String[] args)
    {
        int arr1[] = {1, 2, 3};
        int arr2[] = {1, 2, 3};
        if (arr1 == arr2)
            System.out.println("Same");
        else
            System.out.println("Not same");
    }
}
```

5)

```
String[] cities = {"London", "Paris", "Rome", "New York"};
System.out.println(cities[cities.length-2]);
```

6) Bir array’de herhangi iki elemanın aynı olup olmadığını kontrol eden bir program yazınız. Aynı eleman varsa “Aynı eleman var” yoksa “Aynı eleman yok” yazdırınız.

Soru:

Kullanıcıya kaç elemanlı bir array gireceğini sorun.

Kullanıcıdan array'in elemanlarını girmesini isteyin.

a) Bu array'in tüm elemanlarını ekrana yazdırın.

b) Bu array'in son elemanını ilk eleman yapın ve tüm elemanlarını ekrana yazdırın.

Mesela; array {1, 2, 3} ise ekrana {3, 1, 2} şeklinde yazdırın.

Soru:

Kullanıcıya kaç elemanlı bir array gireceğini sorun.

Kullanıcıdan array'in elemanlarını girmesini isteyin.

a) Bu array'in içinde herhangi bir elemanın var olup olmadığını kontrol edin.

b) Bu array'in tüm elemanlarını tersten yazdırın.

Mesela; array {1, 2, 3, 4} ise ekrana {4, 3, 2, 1} şeklinde yazdırın.

Bir Array'in Elemanlarını Kucukten Buyuge Sıralamak

```
int[ ] numbers = { 6, 9, 1 };
```

```
Arrays.sort(numbers);
```

```
for (int i = 0; i < numbers.length; i++){  
    System.out.print ( numbers[i] + " " );  
}
```

Output is 1, 6, 9

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

```
    int arr[] = {2,1,7,6};  
    Arrays.sort(arr);  
    for(int i=0; i<arr.length; i++){  
        System.out.print(arr[i] + " ");  
    }
```

?

```
    String str[] = {"Ali", "Ahmet", "Kemal", "Can"};  
    Arrays.sort(str);  
    for(int i=0; i<str.length; i++){  
        System.out.print(str[i] + " ");  
    }
```

?

```
}
```