

Java ile Programlamaya Giriş Eğitimi



3.HAFTA

15 Kasım 2020



<https://github.com/milikkan/acm-hacettepe-java>

Geçen Haftanın Özeti

- **IDE kullanımı**
- **Mantıksal operatörler**
- **if-else karar yapısı**
- **while döngüsü**
- **for döngüsü**
- **continue ve break kelimeleri**

Bu Haftanın Konuları

- **switch-case yapısı**
- **Ternary operator (?:)**
- **do-while döngüsü**
- **Tek ve 2 boyutlu diziler**
- **İç içe döngüler**
- **Java sınıf ve nesnelere giriş**
- **import ve package kavramları**
- **Java dokümantasyonunun kullanılması**
- **String sınıfı ve temel metotları**
- **ArrayList sınıfı ve temel metotları**
- **for each döngüsü**

Ödevlerin Çözümleri

1. Vücut kitle indeksi programını geliştirin. Eğer “normal” sonucu çıkmasa kişinin kaç kilo alması veya kaç kilo vermesi gerektiğini hesaplayıp ekrana yazdırın.

2. Bir sayının faktörünü hesaplayan programı yazın.

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

3. Sayı tahmin oyunu yazın. Bilgisayar 1 ile 100 arasında rastgele bir sayı tutsun. Program kullanıcıyı “in”, “çık” şeklinde yönlendirerek sayıyı bulana kadar devam etsin. En son kaç denemede bulunduğunu ekrana yazsın.

Rastgele sayı üretimi için `Math.random()` metodunu inceleyin.

Üçlü if (ternary if)

- (ifade) ? (true ise yap) : (false ise yap)

* **ifade**: herhangi bir boolean değer döndüren ifade

* **eşidi**:

```
if (ifade) {  
    // true ise yap  
} else {  
    // false is yap  
}
```

switch-case

- Birçok alternatif arasından seçim yapmamızı sağlar.

switch (ifade) {

case değer1:

// işlemler

break;

case değer2:

// işlemler

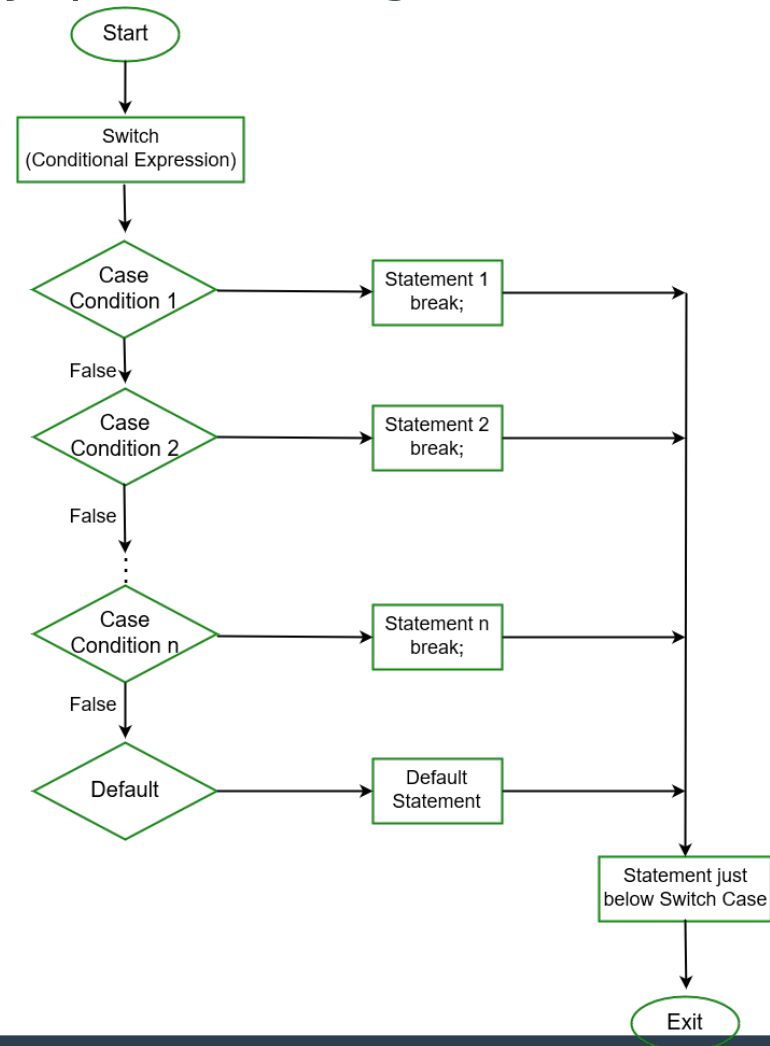
break;

default:

// işlemler

// break

}



do-while

- Döngüde işlemi en az bir kez tekrar etmemiz gerekiyorsa faydalı olur.

```
do {  
    // işlemler  
    // işlemler  
} while (ifade);
```

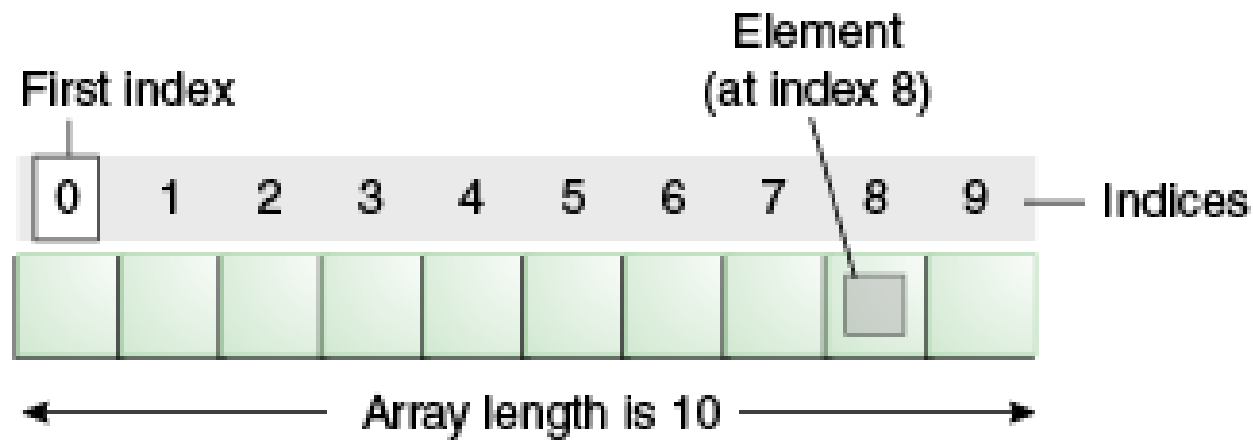
Diziler

- **Belirli bir tipteki veriden istenilen sayıda saklamak için bir veri yapısıdır.**
- **Veri tipinin yanına [] eklenerek tanımlanır.**
 - ◆ `int[] sayilar;`
- **New operatörü ile oluşturulması gerekir. Oluşturulurken dizinin kapasitesi belirtilmek zorundadır.**
 - ◆ `sayilar = new int[2]`
- **İlk değerleri tanımlama aşamasında da verilebilir. Bu durumda kapasite belirtilemez.**
 - ◆ `sayilar = new int[]{1, 2, 3}`

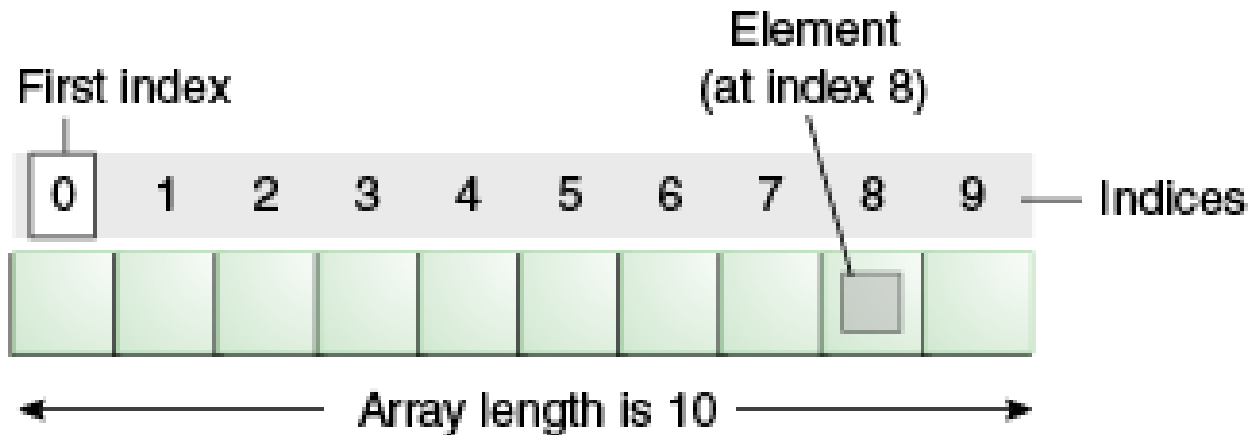
Diziler

- **Dizi elemanlarına [] ile erişilir.**

- renkler[0]
- renkler[1]



Diziler



- **Dizi elemanlarına yeni değer atanabilir.**
 - ◆ `renkler[1] = "mavi";`
- **Dizideki eleman sayısı "length" özelliği ile öğrenilebilir.**
- **Olmayan bir indise erişim sağlanmaya çalışılırsa hata verir.**

Örnek: Not Ortalaması

- * 5 adet not değeri girip ortalamasını hesaplayan bir program yazınız.**
- * geliştirme: Program girilen notların en yüksekini de bulsun.**

for each Döngüsü

* Diziler veya koleksiyonlar üzerinde hızlıca döngü oluşturmak için kullanılır.

```
for (tip değişken : dizi) {  
    // işlemler  
    //  
}
```

2 Boyutlu Diziler

* Dizi içindeki elemanlar da dizi ise 2 boyutlu bir dizi elde etmiş oluruz. Matris oluşturmak için kullanılır.

* Tanımlama:

```
int[][] 2dArray = new int[sayı][]
```

- sadece ilk dizinin boyutunu belirtmek zorunludur.
- 3 X 3 lük bir 2 boyutlu dizi örneği

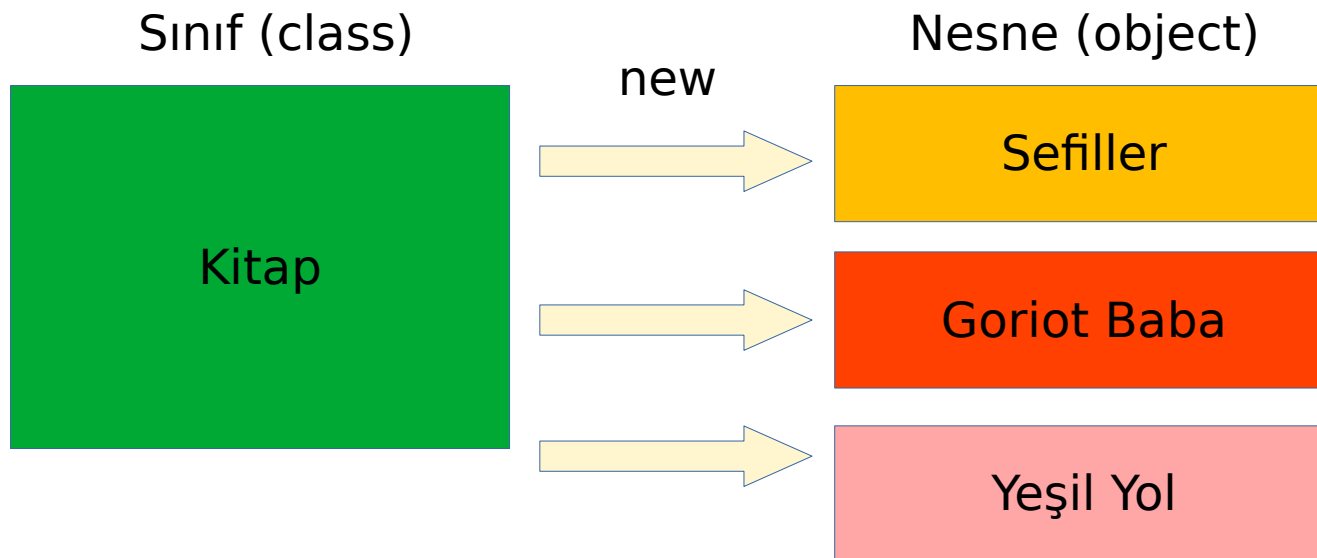


Örnek: 2 Boyutlu Diziler

- * Öğrencilerin notlarını saklayan ve ekrana yazan bir program
- * 2 boyutlu dizi içerisinde gezinmek için 2 adet iç içe geçmiş döngü kullanmak gerekiyor.

Sınıf ve Nesne nedir?

- * Nesneler sınıfların birer örneğidir.
- * Bu anlamda, sınıf bir şablon görevi görür.



import ve package

- * Java sınıf ve arayüz dosyaları paketler içerisinde bulunur.
- * Java standart kütüphanesindeki herhangi bir sınıfı kullanmak için import etmek gerekir.
- * **import** kodun en başında **package** ifadesinden sonra yer almalıdır.
- * **import** kullanılmazsa sınıfın uzun ismi kullanılmak zorundadır.
- * **java.lang.*** paketi otomatik olarak import edilmiş durumdadır.

String Sınıfı

- **String sınıfı immutable'dır. Başlıca metotları:**

- * length()
- * toUpperCase(), toLowerCase()
- * charAt(int index)
- * equals(Object o)
- * endsWith(String suffix)
- * indexOf(int n)
- * isEmpty()
- * replace(char old, char new)
- * substring(int begin, int end)
- * strip()

StringBuilder Sınıfı

- * **immutable değildir**
- * **string birleştirme işlemlerini daha performanslı ve etkin bir şekilde yerine getirir.**
- * **append(), delete(), insert() gibi metotlar sağlar.**

ArrayList Sınıfı

- * **Sabit uzunlukta olmayan dinamik diziler oluşturmak için kullanılır.**

- * **Tanımlanması:**

```
ArrayList<Tip> list = new ArrayList<>();
```

Haftaya...

- **Kendi sınıflarımızı yazma**
- **Metotlar ve sınıf değişkenleri**
- **static kelimesi**
- **Constructors (kurucular)**
- **Inheritance (kalıtım)**
- **Encapsulation (kapsülleme)**
- **Erişim kontrolü (public, private, protected)**
- **Inheritance (kalıtım)**
- **Arayüzler (interface)**
- **Metot yükleme (overloading)**
- **Metot ezme (overriding)**