

Java ile Programlamaya Giriş Eğitimi



2.HAFTA

8 Kasım 2020



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

Geçen Haftanın Özeti

- **Java ekosistemi**
- **Java kodunun derlenip çalıştırılması**
- **main metodu**
- **Veri tipleri**
 - Tam sayı
 - Ondalıkli sayı
 - Mantıksal tipler
 - Karakter tipler
 - String (dizgeler)
- **Aritmetik Operatörler**
- **İlişkisel Operatörler**
- **Değişken Tanımlama ve İklendirme**
- **Ekrana yazı yama ve kullanıcıdan girdi alma**

Bu Haftanın Konuları

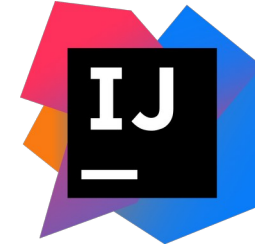
- **IDE kullanımı**
- **Akış kontrolü**
- **Mantıksal operatörler**
- **if-else karar yapısı**
- **while döngüsü**
- **for döngüsü**
- **continue ve break kelimeleri**
- **Diziler**
- **Bir sürü örnek...**

IDE Seçenekleri

- **IDE:** Integrated Development Environment
- Tümüleşik Geliştirme Ortamı
- **Faydaları:**
 - Hızlı kod geliştirme
 - Otomatik tamamlama, sentaks renklendirme
 - Test, build, kod analizi, deployment vb için entegre araçlar
- İlk öğrenme maliyeti yüksek olabilir
- Küçük programlar için hantal kalır
- Genelde fazla sistem kaynağı kullanırlar

IDE Seçenekleri

- **IntelliJ IDEA**
 - Community sürümü ücretsiz
 - Enterprise sürümü için lisans gerekiyor
- **Eclipse**
 - Ücretsiz, açık kaynak
 - Eclipse vakfı (aynı zamanda JakartaEE yi geliştiriyor)
 - Çok yetkin ancak karmaşık gelebilir
- **NetBeans**
 - Original Sun Microsystems / Oracle
 - Apache vakfına devredildi
 - Ücretsiz, açık kaynak
- **VSCode**
 - Microsoft ürünü
 - Ücretsiz, açık kaynak
 - Diğerleri kadar popüler ve yetkin değil ancak gelişmeye devam ediyor.
 - Daha hafif ve hızlı
- **Online alternatifler**
 - repl.it
 - gitpod
 - AWS Cloud9



Visual Studio Code



IDE Seçenekleri

- “Merhaba” programı
 - IntelliJ demosu
 - repl.it demosu
- Not:
 - IntelliJ kurulmadan önce JDK kurulması gerekir.
 - Daha önce kurulmadıysa, IntelliJ içerisinden de JDK eklenebilir.

Örnek: Basit Toplama Programı

- **Planlama adımları:**

1. Hoş geldin mesajını ekrana yaz
2. Birinci sayıyı kullanıcıdan al
3. Kullanıcıdan alınan sayıyı ilkSayi isimli String değişkene kaydet
4. İkinci sayıyı kullanıcıdan al
5. Kullanıcıdan alınan sayıyı ikinciSayi isimli String değişkene kaydet
6. İlkSayi ve ikinciSayi değişkenlerini topla ve sonucu ekrana yaz.

Örnek: Basit Toplama Programı

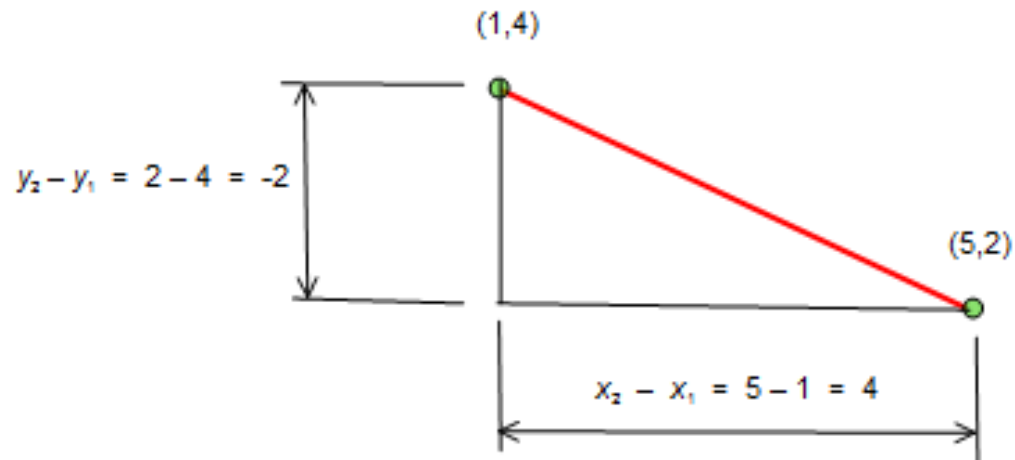
- **Tekrar Düzenlenen Planlama adımları:**

1. Hoş geldin mesajını ekrana yaz
2. Birinci sayıyı kullanıcıdan al
3. Kullanıcıdan alınan sayıyı ilkSayi isimli String değişkene kaydet
- 4. İlk sayıyı integer tipine çevi (`Integer.parseInt()`)**
5. İkinci sayıyı kullanıcıdan al
6. Kullanıcıdan alınan sayıyı ikinciSayi isimli String değişkene kaydet
- 7. İkinci sayıyı integer tipine çevir.**
8. İlkSayi ve ikinciSayi değişkenlerini topla ve sonucu ekrana yaz.

Örnek: İki Nokta arasındaki Mesafe

1. İki nokta arasındaki mesafeyi hesaplayan programı yazın.

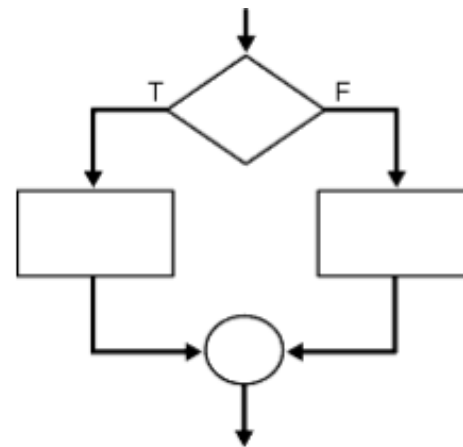
- Not: Pisagor teoremi. Karekök alma için `Math.sqrt()` metoduna bakın.



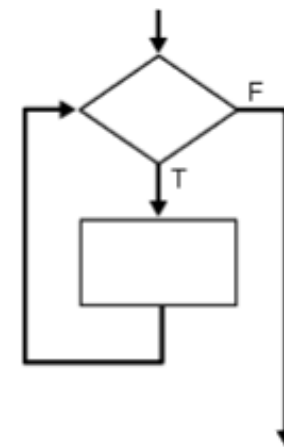
$$\text{Distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Akış Kontrolü

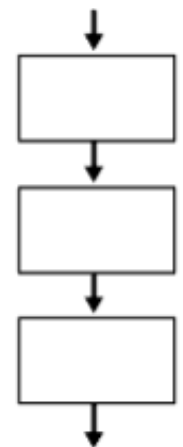
- Program komutları yukarıdan aşağıya doğru lineer olarak çalışır.
- Bu akış, karar ve döngü yapıları ile değiştirilebilir.
 - Temel karar yapıları
 - if-else
 - switch-case
 - Temel döngü yapıları
 - while
 - do-while
 - for
 - for each



Selection



Iteration



Sequence

Karar Yapılarına Giriş (if-else)

- Normalde program akışı yukarıdan aşağıya sırayla gider.
- Program akışını bazı şartlarda değiştirmek için if-else yapısını kullanabiliriz.

```
if (şart) {  
    // işlemler  
}
```

* tek bir işlem satırı varsa blok açmaya gerek yok.

```
if (şart) {  
    // işlemler  
} else {  
    // işlemler  
}
```

* şart true ise üstteki blok

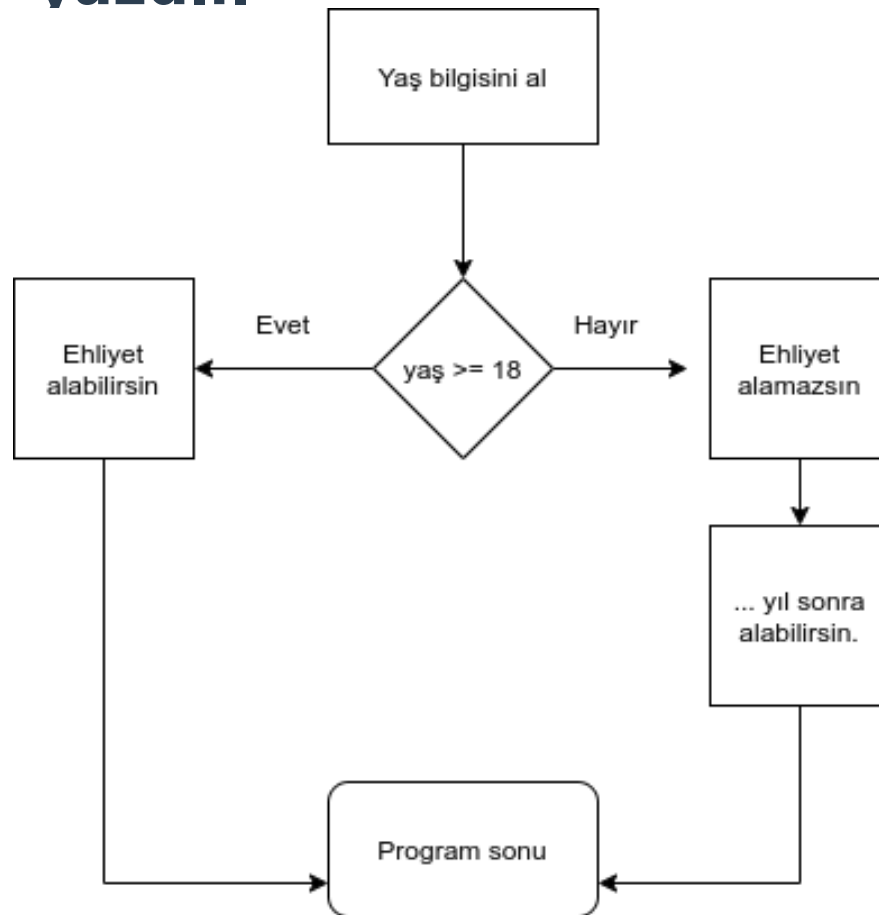
* false ise alttaki bloğu çalıştır

Örnek: Tek mi Çift mi

- **Kullanıcıdan bir sayı alarak, girilen sayının tek mi çift mi olduğunu ekrana yazan program.**
 - Sayıyı kullanıcıdan al
 - Sayı çift mi? (buradaki karar if-else yapısı ile verilir)
 - Evet -> Ekrana “Girdiğiniz sayı çift” yaz
 - Hayır -> Ekrana “Tekrar deneyiniz” yaz

Örnek: Yaş Kontrolü

- Kullanıcıya yaşını sor. Yaşı 18'den küçükse “Ehliyet alabilirsin” vazdır.



Mantıksal Operatörler

- **&&** -> “and” operatörü
- **||** -> “or” operatörü
- **!** -> “not” operatörü

&& (and)	true	false
true	true	false
false	false	false

 (or)	true	false
true	true	true
false	true	false

! (not)	
true	false
false	true

Örnek: Vücut Kitle İndeksi

Kişinin boyu ve kilosunu alarak vücut kitle indeksini hesaplayan ve çıkan sonuca göre kişinin kategorisini ekrana yazan bir program yazın.

- < 18.5 : zayıf
- >= 18.5 - < 25 : normal
- >=25 - < 30: fazla kilolu
- >=30: obezite

$$VKİ = \frac{\text{vücut ağırlığı(kg)}}{(\text{boy(m)})^2}$$

Artırma ve Azaltma Operatörleri

- **Pre-increment, pre-decrement**

- Önce artır veya azalt, sonra işlemi yap

`++x`

`--x`

- **Post-increment, post-decrement**

- Önce işlemi yap, sonra artır veya azalt

`x++`

`x--`

Döngüler (while döngüsü)

- while (koşul) {



}

- Koşul true değer döndürdükçe blok içerisindeki kodlar çalışmaya devam eder.

Döngüler: Örnekler

1. ilk 10 sayıyı yazdır
2. sayı 5 ise yazdırma
3. sayı 5 ve 7 ise yazdırma
4. sadece 2'nin katlarını yazdır
5. sadece 2 ve 3 ün katlarını yazdır

Döngüler (for döngüsü)

for (iniklendirme; koşul; son işlem) {

.....	blok
----------------	------

}

- **İniklendirme** -> sadece ilk döngünün başında çalışır
- **Koşul** -> her döngü başında kontrol edilir.
true döndüğü sürece döngü çalışır
- **Son işlem** -> her döngü sonunda çalışır

Döngü Kontrolü (break, continue)

- **break -> döngü sonlandırılır.**
 - > akış kontrolü döngü sonrasına geçer
- **continue -> mevcut döngü pas geçer**
 - > akış kontrolü döngü başına geçer
 - > mevcut döngü için kalan satırlar çalışmaz

Örnek: Yazar Kasa

- * Teker teker fiyatı girilen ürünlerin toplam fiyatını hesapla. Ürün fiyatı boş geçilene kadar toplamaya devam et. En son toplamı ekrana yazdır.**
- * Geliştirme: Satın alınan ürün adedini de yazdır.**
- * Geliştirme: Kullanıcıya ne kadar parası olduğunu sor. Eğer mevcut toplam parayı geçerse, toplama işlemini durdur ve ekrana “paranız yetersiz” yaz.**

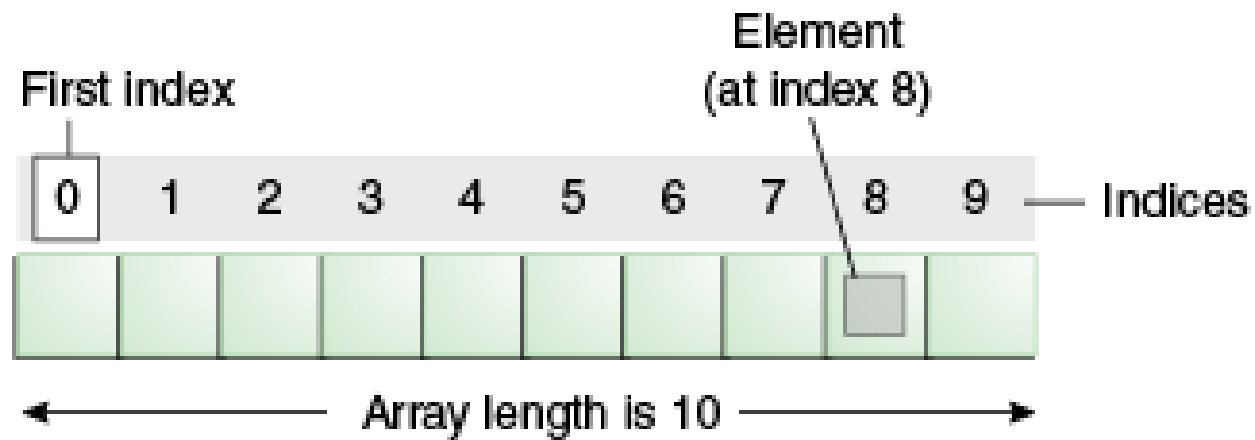
Diziler

- **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}`

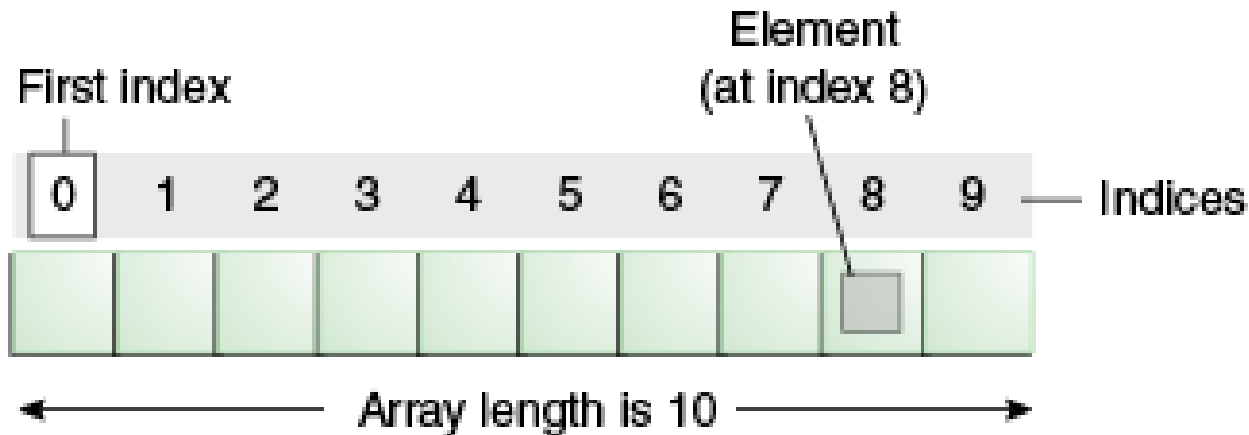
Diziler

- **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.**

Haftaya...

- **switch-case karar yapısı**
- **2 boyutlu diziler**
- **İç içe döngüler**
- **String sınıfı ve temel metotları**
- **Math sınıfı ve temel metotları**
- **Ternary operator (?:)**
- **Java sınıf ve nesnelere giriş**
- **metot yazımı**
- **import ve package kavramları**

Ödevler

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.