## Bilgisayar Mühendisliğine giriş

Ders 5: Algoritma ve Akış Şemaları Doç. Dr. Mehmet Dinçer Erbaş Bolu Abant İzzet Baysal Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü

#### **Bazı tanımlar**

- Bilgisayar programlanabilir bir makinedir. Kendisine verilen komut listelerini gerçekleştirebilir.
- Bahsedilen komut listeleri program ismiyle adlandırılır.
- Bir program belli bir problemi çözmek veya bir görevi gerçekleştirmek için hazırlanır.
- Bilgisayar programlama karmaşık problemlerin bilgisayar tarafından çözülmesine olanak vermek için bilgisayar programları dizayn etme ve oluşturma aktivitiesidir.

#### **Bazı tanımlar**

- Bir bilgisayar programcısı:
  - Öncelikle çözmek istediği problemi tanımlar.
  - Daha sonra probleme uygun çözümü dizayn eder.
  - Son olarak önceki aşamada tanımlanmış çözümü bilgisayar tarafından anlaşılabilecek bir programlama dilinde oluşturur.
- Bir bilgisayar tam olarak verilen programı gerçekleştirir.
- Bu sebeple, programın doğru şekilde yazılması bilgisayar programcısının görevidir.
- Yukarıda belirtilen çözümler komut, fonksiyon veya formul listeleri şeklinde oluşturulur.
  - Geliştirilen çözüm **algoritma** şeklinde adlandırılır.

#### Algoritma

- Algoritma, herhangi bir sorunun çözümü için izlenecek yol anlamına gelmektedir.
- Çözüm için yapılması gereken işlemler hiçbir alternatif yoruma izin vermeksizin sözel olarak ifade edilir.
- Diğer bir deyişle algoritma, verilerin, bilgisayara hangi çevre biriminden girileceğinin, problemin nasıl çözüleceğinin, hangi basamaklardan geçirilerek sonuç alınacağının, sonucun nasıl ve nereye yazılacağının sözel olarak ifade edilmesi biçiminde tanımlanabilir.
- Algoritma hazırlanırken, çözüm için yapılması gerekli işlemler, öncelik sıraları gözönünde bulundurularak ayrıntılı bir biçimde tanımlanmalıdırlar.
- Algoritma tanımlandıktan sonra algoritmanın doğru çalıştığı ve bu sayede her türlü girdi için doğru çıktıları oluşturduğu matematiksel olarak ispatlanmalıdır.

#### Algoritma

- Günlük aktivitelerimizde birçok karmaşık veya basit problemi çözüyoruz.
- Örnek: Evden okulda gelme
  - Bu basit görünen aktivitenin çözümü birçoka aktivitenin yapılması ile mümkündür
    - 1. evden çık
    - 2. Otobüse bin
    - 3. Okul durağında in
    - 4. fakülteye git
    - 5. Dersin sınıfını bul
    - 6. Boş bir yere otur
  - Başardın!

# Bilgisayar programlamaya giriş

- Bilgisayarlar belli hesaplamaları veya problem çözümlerini insanlardan çok daha hızlı şekilde yapabilirler.
- Ayrıca, bilgisayara verilen program doğru ise, bilgisayarlar yaptıkları hesaplarda hata yapmazlar.
- Örnek bir hesaplama: 121'e kadar asal sayıları bulunuz.
  - Ne kadar süre alır?
  - Hata yapma ihtimaliniz var mı?
- Çözüm algoritması örneği
  - http://en.wikipedia.org/wiki/File:Sieve\_of\_Eratosthenes\_animation.gif

## Bilgisayar programlamaya airis

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

#### Prime numbers

# Bilgisayar programlamaya giriş

- Bir bilgisayar programı yazarken aşağıdaki adımlar gerçekleştirilmelidir:
  - Ilk olarak programın hesaplaması istenen çıktılar belirlenmelidir.
    - Bunlar program çalıştığında oluşturulacak sonuçlardır.
  - Ikinci olarak istenilen çıktıların oluşturulması için gereken girdiler belirlenmelidir.
    - Bu girdiler çıktıların hesaplanması için gereklidir.
  - Üçüncü olarak, girdileri kullanarak çıktıları hesaplamak için gereken algoritma oluşturulmalıdır.
    - Bu aşamadan sonra programı yazmaya başlayabilirsiniz!
  - Son aşama test ve var olan sorunların giderilmesidir.
    - Bu sayede programın doğru çalıştığını görürsünüz.

- Pseudocode veya yalancı kod, herhangi bir programlama dilinden bağımsız olarak bir algoritmanın çalışma mantığını açıklayan bir komut topluluğudur.
  - Genel yapı olarak bir programlama dilinin görünümünde yazılır ancak bilgisayardan çok insanların algoritmayı okuyup anlayabilmesi için yazılır.
- Akış diyagramı (İng: Flow chart) algoritmanın ne şekilde çalışacağını gösterir.
  - Algoritmanın her adımını farklı türde kutularla gösterir ve bu adımlar arası geçişleri oklar ile gösterir.

- Örnek program: Çarpma işlemi yapan program.
- Yukarıda belirtilen basit programın yalancı kodu şu şekilde oluşturulabilir:

# Başla Oku: A, B Hesapla: C = A \* B Göster: C

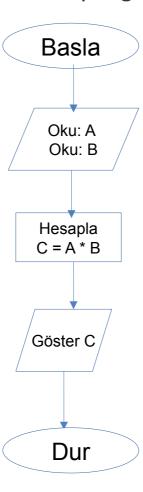
- Örnek program: Çarpma işlemi yapan program.
- Akış diyagramları ile ilgili algoritmayı açıklayabiliriz.

Simge	Simgenin Adi	Simgenin Anlami					
	Elips	Akis diyagraminin baslangiç ve bitis yerlerini gösterir. Baslangiç simgesinden çikis oku vardir. Bitis simgesinde giris oku vardir.					
	Paralel Kenar:	Programa veri girisi ve programdan elde edilen sonuçlarin çikis islemlerini gösterir.					
	Dikdörtgen	Aritmetik islemler ve degisik atama islemlerinin temsil edilmesi için kullanılır.					
	Eskenar Dörtgen	Bir karar verme islemini temsil eder.					
<u>◄</u>	Altigen	Program içinde belirli blokların ard arda tekrar edilecegini gösterir.					
1+	Oklar	Diyagramin akis yönünü gösterir.					

Örnek program: Çarpma işlemi yapan program.

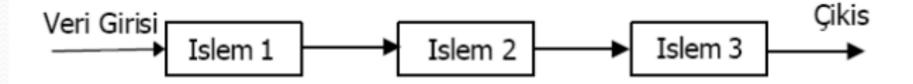
Yukarıda belirtilen basit programın akış diyagramı şu şekilde

oluşturulabilir:



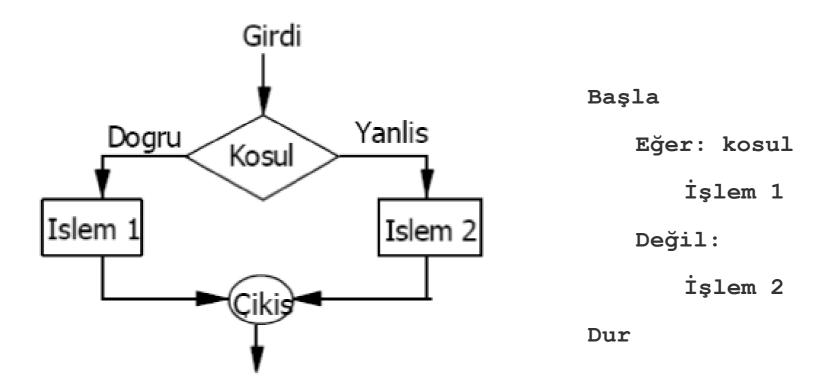
- Algoritma oluşturulurken üç temel parça kullanılır.
  - Doğrusal (Sıralı) operasyonlar
  - Mantıksal (karar verme) operasyonlar
  - Döngüsel operasyonlar

Sıralı yapı

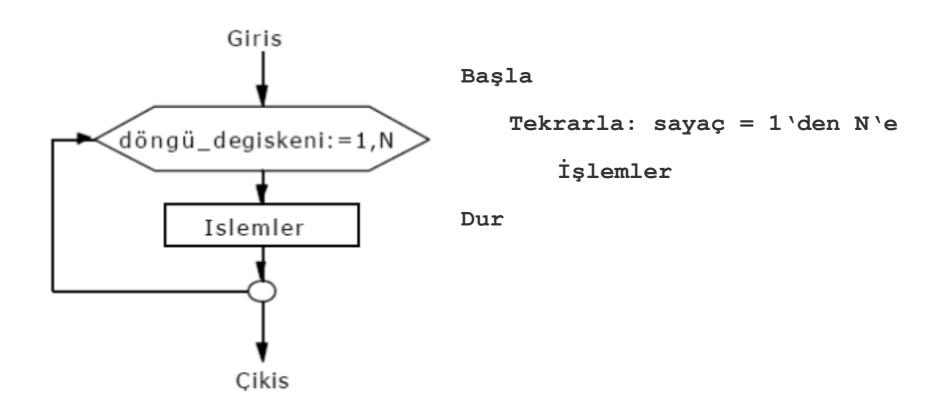


```
Başla
   İşlem 1
   İşlem 2
   İşlem 3
```

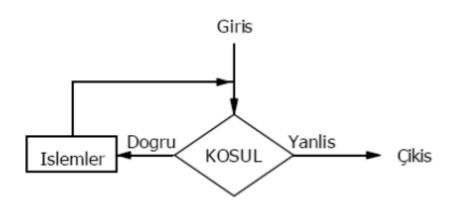
Karar verme yapısı



Döngüsel yapı



Döngüsel yapı



Başla

Doğru ise tekrarla: Koşul

İşlemler

< küçüktür

> büyüktür

<= küçük veya eşittir

>= büyük veya eşittir

== eşittir

!= eşit değildir

#### Örnekler:

x > 5 sayac <= 16 Toplam <= 25

 İki koşulu tek seferde yazamıyoruz ancak "ve" – "veya" kullanarak iki koşulu birleştirebiliriz.

#### Örnekler:

Ogrenci\_sayisi > 10 ve not == 90 Sıcaklık > 25 veya hava == 'Gunesli'

- Dikkat etmemiz gereken konular
  - Ödev, sınav vb. her türlü çalışmayı <u>kendimiz</u> yapıyoruz.
  - Kesinlikle "satıra git" komutu kullanmıyoruz.
    - Derste gördüğümüz komut tipleri ile çözüm üretmeye çalışmalısınız.
  - Belli bir bloğun içine girildiğinde bir tab kadar satırı itmeniz gerekiyor.
    - Örneğin "Koşul" doğru ise yapılacaklar üstteki satırın bir tab daha ilerisinden başlamalı.
  - Koşulların "Değil" kısmına yeni bir koşul yazılmamalı.
  - Bilgisayardan yapmasını isteyebileceğimiz tek satırlık komutlar:
    - Matematiksel işlemler (+,-,/,\*,%) (Hesapla).
    - Ekrana bir şey yazdırma (Göster), ekrandan girdi okuma (Oku).

- Kullanıcıya hoşgeldin diyen bir program yazalım.
- Kenar uzunluğu verilen bir karenin çevresini ve alanını hesaplayan bir program yazalım.
- Verilen iki sayıdan büyük olanını hesaplayan bir program yazalım.
- 1'den 10'a kadar sayıların toplamını hesaplayan bir program yazalım.
- 10 kişilik bir sınıfta bir sınavdan alınan ortalama notu hesaplayan bir program yazalım.

Kullanıcıya hoşgeldin diyen bir program yazalım.

```
Başla

Göster: "Lütfen isminizi yazınız"

Oku: isim

Göster: "Hoşgeldin" isim
```

 Kenar uzunluğu verilen bir karenin çevresini ve alanını hesaplayan bir program yazalım.

#### Başla

```
Oku: KenarUzunluğu

Hesapla: Çevre = KenarUzunluğu * 4

Hesapla: Alan = KenarUzunluğu * KenarUzunluğu

Göster: Çevre

Göster: Alan
```

Verilen iki sayıdan büyük olanını hesaplayan bir program yazalım.

#### Başla Oku: sayi1 Oku: sayi2 Eğer: sayi1 > sayi2 Göster: sayil "büyük" Değil: Eğer: sayi2 > sayi1 Göster: sayi2 " büyük" Değil: Göster: "Sayılarımız eşit"

• 1'den 10'a kadar sayıların toplamını hesaplayan bir program yazalım

```
Başla
  Hesapla: toplam = 0
  Hesapla: sayaç = 1
  Doğru ise tekrarla: sayaç < 11
    Hesapla: toplam = toplam + sayaç
    Hesapla: sayaç = sayaç + 1
  Göster: toplam</pre>
```

1'den 10'a kadar sayıların toplamını hesaplayan bir program yazalım

```
Başla
  Hesapla: toplam = 0
  Tekrarla: sayaç = 1'den 10'a
    Hesapla: toplam = toplam + sayaç
  Göster: toplam
```

 10 kişilik bir sınıfta bir sınavdan alınan ortalama notu hesaplayan bir program yazalım.

```
Başla

Hesapla: toplam = 0

Tekrarla: sayaç = 1'den 10'a

Göster: "Not giriniz"

Oku: not

Hesapla: toplam = toplam + not

Hesapla: ortalama = toplam / 10

Göster: ortalama
```

- Verilen üç tamsayı içindeki en büyük değeri bulan bir program yazalım.
- Verilen 20 tamsayı içindeki en büyük değeri bulan bir program yazalım.
- Farklı sayılarda öğrenci içeren sınıflarda bir sınavdan alınan en büyük, en düşük ve ortalama notu hesaplayan bir program yazalım.