

Problem Çözme ve Algoritmalar



Problem Çözme Kavramları ve Yaklaşımlar

Programlama nedir?

- Belirli bir amaca ulaşmak için oluşturulan algoritmanın herhangi bir programlama dilinde yazılarak bilgisayar tarafından işlenmesine programlama denir.
- Bir bilgisayar bilimcisi için en önemli beceri problem çözme becerisidir.
- Bilgisayar bilimcileri genel olarak matematiksel sembolleri, işlemleri ve formülleri kullanır, mühendisler gibi tasarım yaparak farklı sistemler oluşturur ve bilim insanları gibi deney yaparak teknoloji desteği ile çözüm üretir.
- Programlama;
Bilgisayarın donanıma nasıl davranacağını anlatan, bilgisayara yön veren komutlar ve işlemler bütünüdür.
- Programlama, hem problem çözme becerisi hem de bilgi işlemsel düşünme becerisine sahip olmayı gerektirir.
- Programlama, günlük hayattaki genel veya özel problemlerin makinelere tanıtılması, öğretilmesi ve öğretilen çözüm yolları ile sonuca ulaştırılmasını sağlamak amacıyla kullanılan teknik bir terimdir.
- **Programlama, bir bilgisayara ya da cihaza nasıl davranacağını, hangi durumlarda ne tepki vereceğini ve bu tepkiler sonucunda nelerin etkileneceğini öğretme işlemidir.**
- Söz konusu cihazın davranışlarını önceden belirleyip, o komutlara göre çalışacak yazılımı yapmak, bu yazılımı cihazın anlayacağı dile çevirmek ve cihaza yükleyerek cihazın nasıl çalışacağını ona bildirmek programlamadır. Nasıl ki beynimiz hiç karşılaşmadığı bir durumla karşılaşınca nasıl davranacağını bilmez, cihazlarda öyledir. İşte durumlar karşısında cihazın davranışlarını belirleme işine programlama diyoruz.

Programlama işlemi 6 ana temelden oluşur:

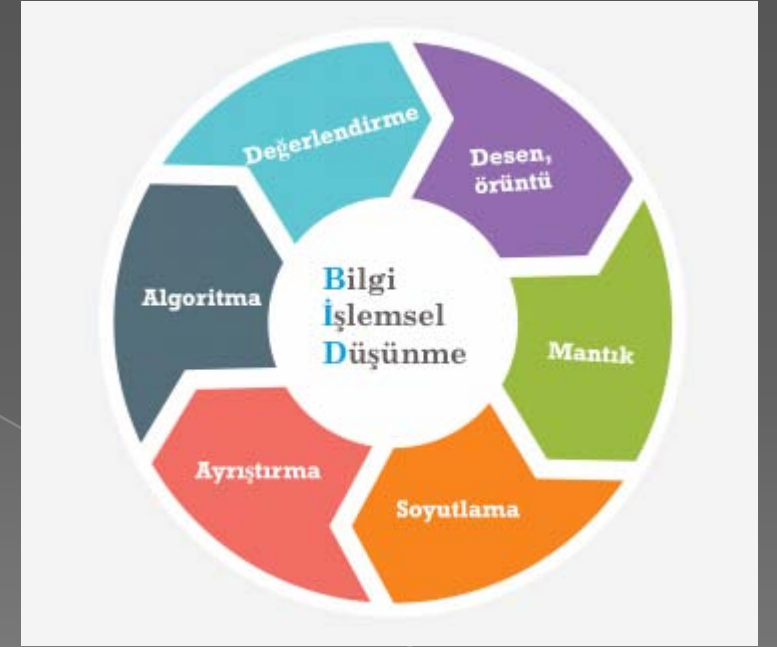
- **1- Problemin belirlenmesi;** Bir program yazmak için öncelikle ortada bir problem olması gerekir. Bu problemin ne olduğunu iyi tanımak ve probleme neden olan etkenleri iyi belirlemek lazım gelir.
- **2- Çözüm yollarının belirlenmesi;** Probleme neden olan etkenleri ortadan kaldırmak için bir çözüm yolu geliştirilmeli ve bu çözüm yolunun problemi en doğru şekilde çözeceğine emin olunmalı.
- **3- Algoritmanın geliştirilmesi;** Algoritma bir problemin çözümü için gerekli basamakları sıralayıp çözüme ulaştıracak merdiveni tamamlama biçimidir. Programlamada çok önemli yere sahiptir. Mantıksal ve sözel olarak çözüme ulaşmak için hangi sırayı takip etmesi gerektiğini belirtmek için kullanılır.
- **4- Akış şeması oluşturulması;** Akış şeması algoritmanın görselleştirilmiş halidir diyebiliriz. Algoritma ve akış şeması programı en az hata ile en küçük ayrıntıları dahi atlamadan yazmamız için büyük kolaylık sağlayacaktır. Algoritma ve akış şemasını daha geniş kapsamlı olarak ele alacağımız bir ders hazırlayacağız.
- **5- Uygun programlama dilinin seçilmesi;** Programlama dili insan ile programın yazılacağı bilgisayar arasında iletişimi sağlayan, programın nasıl çalışacağı ve çalışırken hangi durumlarda nasıl bir tepki vereceğini standart bir şekilde bilgisayara ve derleyiciye açıklayan sözdizimi şeklidir. Bir örnekle bu konuyu daha iyi açıklayalım; Mesela arkadaşından su getirmesini isteyeceksin, Bunu arkadaşının anlayacağı dilde ona açıklamalısın. Örneğin arkadaşın Türkçe biliyorsa ona "Şu getirir misin" demelisin. Arkadaşına su getirmesini söylediğin bu dile programlama dilidir diyebiliriz.

Programlama işlemi 6 ana temelden oluşur:

- **6-Programın yazılması, derlenmesi ve test edilmesi;** Gerekli olan programın bilgisayar yardımıyla yazılması ve derlenmesidir. Derleme kavramı bilgisayar ile yazdığımız programın derleyici yardımıyla programın çalıştırılacağı cihazın anlayacağı dile (Makine Diline) dönüştürülmesidir. Biraz önce su isteme örneğinde arkadaşın anlayacağı dil ile ondan su istemiştik. Burada da aynı örnek ile devam edelim. Arkadaşımız bizim su istediğimizi algılar ve beyinde suyun ne olduğunu, nerde bulabileceğini, neyle ve nasıl taşınacağını, çözümler. Bu işlemi derlemeye benzetebiliriz. Test aşaması ise programın düzgün çalışıp çalışmadığını, komutlara gerekli tepkiyi gösterip göstermediğini, olası ve olağan dışı durumlarda nasıl tepki verdiğinin testini yapmaktır.

Bilgi işlemsel düşünme

Bilgisayar biliminin kavramlarından yararlanarak problem çözüme, sistem tasarlama ve insan davranışlarını anlama olarak tanımlanabilir.



Bilgi işlemsel düşünme

Bilgi İşlemsel Düşünme, insan zihninin iyi yaptığı şeyleri insan zihniyle, bilgisayarların iyi yaptığı şeyleri de bilgisayarla yapmaktır. Bilgi işlemsel düşünme bir problem çözme sürecidir. Bazı karakteristik özellikleri:

- ❖ En verimli ve etkili çözüme ulaşmak için muhtemel çözümlerin tanımı, analizi ve uygulanması, gerekli kaynak ve adımların belirlenmesi.
- ❖ Problem çözme becerilerinin genelleştirilmesi ve farklı konulara transfer edilmesi.
- ❖ Problemleri bilgisayarlar ya da başka araçlar kullanacak şekilde formüle etme.
- ❖ Bilginin mantıklı bir şekilde organizasyonu ve analizi.
- ❖ Bilginin modeller ve simülasyonlar ile temsil edilmesi.
- ❖ Algoritmik düşünce ile problem çözümlerinin otomasyonu.

Bilgi işlemsel düşünmenin önemi

- Akademik bilgi ile gerçek dünya arasında bağlantılar kurulmasını , öğrenilen bilgiyi daha anlamlı bir bütün haline getireceklerdir.
- Bilgi işlemsel düşünme öğrencilerin başarı seviyelerini yükseltir.
- Bilgi işlemsel düşünme öğrenmek için büyük miktarlarda paralara ihtiyaç yoktur.
- öğrencileri küresel rekabete hazırlar.
- eleştirel düşünce becerilerini geliştirir.
- Öğrencileri gelecek için hazırlar.
- Bilgi işlemsel düşünme öğrencileri üniversite ve kariyerlerine hazırlar
- Bilgi işlemsel düşünme değişik konuları kapsar.

Program Nedir?

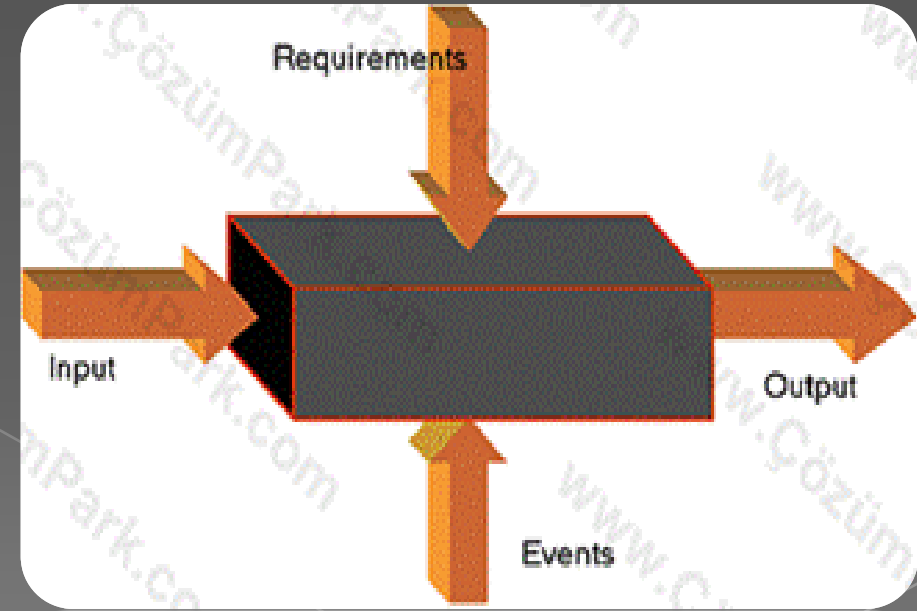
Yapılacak bir işlemi ya da hesaplamayı gerçekleştirmek için birbirini izleyen komut ya da yönergelerden oluşan yapıdır.

Girdi: Klavyeden, dosyadan veya başka bir ayardan veri almaktır.

Çıktı: Ekranda veriyi görüntüleme veya veriyi dosyaya veya başka bir ayarda göndermedir.

Koşullu yürütme: Belirli durumları sınamak ve komutları uygun bir sıraya göre çalıştırmaktır.

Tekrarlama: Bazı eylemleri genellikle ufak tefek değişikliklerle yineleme işlemidir.



Algoritma nedir?

Bir problemin çözümünde izlenecek yol anlamına gelir ve problemin çözümünün adımlar halinde yazılmasıyla oluşturulur. Genellikle matematikte ve programlamada bir işi yapmak için tanımlanan, belli bir başlangıcı ve sonu olan, açıkça belirlenmiş basamaklardır

Değişkenler

Dışarıdan girilen ve bizim oluşturduğumuz değerleri tutan elemanlardır.

Algoritma sözcüğü nereden gelir?

Algoritma sözcüğü Türkistanlı bir âlimden gelir. Türkistanlı âlim 9. yüzyılda Cebir adında bir kitap oluşturarak algoritmik çalışmalarını sergilemiştir. Kısaca algoritmayı bulan kişi kim diye sorulduğunda Ebu Abdullah Muhammed bin Musa el Harezmi diyebiliriz. Bu kitabın matematiğe çok büyük katkıları olmuştur. Bu kitap farklı dillere çevrilmiştir. Avrupa da çok ilgi gören kitabın ismi ve âlimin ismi Avrupalılar tarafından telaffuz edilemediği için “algorizm” sözcüğünü kullanmışlardır. Algorizm “Arap sayıları kullanarak problemleri çözmek” anlamına gelir. Zamanla algorizm algoritmaya dönüşmüştür .

START

forward 2

turn right ▾

right

left

back

90

180

270





```
Kod Modu  
1 forward(2)  
2 turn(right)  
3 forward(3)  
4 turn(left)  
5 forward(2)
```

▶ Çalıştır

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

START

forward 2

turn right ▾

right

left

back

90

180

270





```
Kod Modu
```

```
1 forward(3)  
2 turn(left)  
3 forward(2)  
4 turn(right)  
5 forward(4)
```



 kodla
büyü

forward










getCarrot

turn right ▼



```
1 forward(4)
2 turn(left)
3 forward(4)
4 getCarrot()
5
```


Akış Şemaları

Simge	İşlev
	Başla/Bitir
	Giriş
	Atama/İşlem
	Denetim (Karar)
	Çıkış
	Döngü
	Akış Yönü
	Bağlaç
	Önceden Tanımlı İşlem/Fonksiyon

Şekil 1.10: Akış şeması sembolleri



Bağlantı Sembolü

Akış yönü ve semboller arasındaki bağlantıyı göstermek için kullanılır.



Başlangıç / Bitiş Sembolü

Sonlandırıcı, işleminizin nerede başladığını veya bittiğini gösterir. İşleri daha açık hale getirmek için sonlandırıcı şeklinin içinde 'Başlat', 'Başla', 'Son' gibi kelimeleri kullanabilirsiniz.



Değer atama ve dört işlem Sembolü

Bir işlemi, süreci veya aktiviteyi ifade eder



Giriş / Çıkış Simgesi

sisteme giren veya sistemden gönderilen materyal veya bilgileri temsil eder. Klavyeden değer girilmesi, mouse ile yapılan seçimler, ekranda gösterilmesi, yazıcıdan kağıda basılması gibi



Karar Sembolü

Bir karar veya dallanma noktasını temsil eder. Bir kararın alınması gerektiğini ve bu karardan dolayı sürecin belirli bir yönü izleyeceğini göstermektedir.



Döngü Sembolü ve Bağlayıcı

Program içinde bir işlemin birden çok defa yapılması gerektiğinde kullanılır. Bazı durumlarda bu tekrarlamaya işlemi bir koşula bağlı olarak gerçekleştirir. Akışın, eşleşen bir sembolün (aynı harfi içeren) yerleştirildiği yerde devam ettiğini gösterir.



Veya Sembolü

Proses akışının ikiden fazla dalda devam ettiğini gösterir.

Kullanıcıdan aldığı iki sayının çarpımını bulup ekrana yazan program

Adım 1-Başla

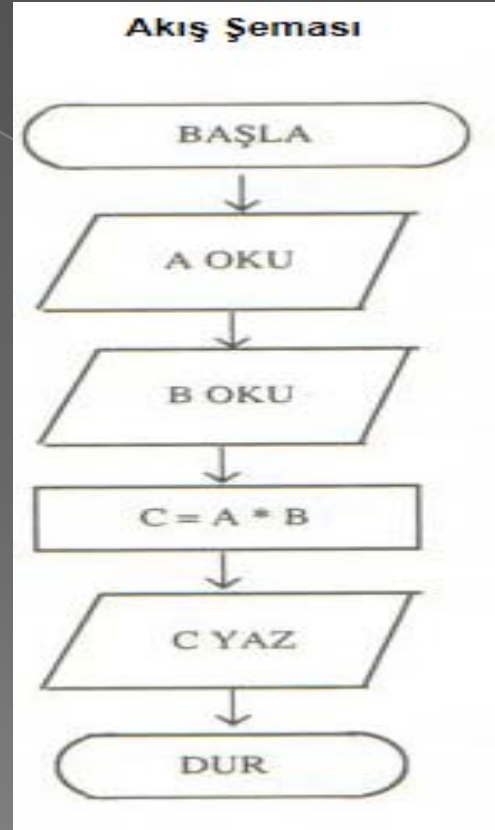
Adım 2-A'yı oku

Adım 3-B'yi oku

Adım 4-C=A*B yi hesapla

Adım 5-C'yi yaz

Adım 6-Dur



```
A = input('1. Sayı : ')
```

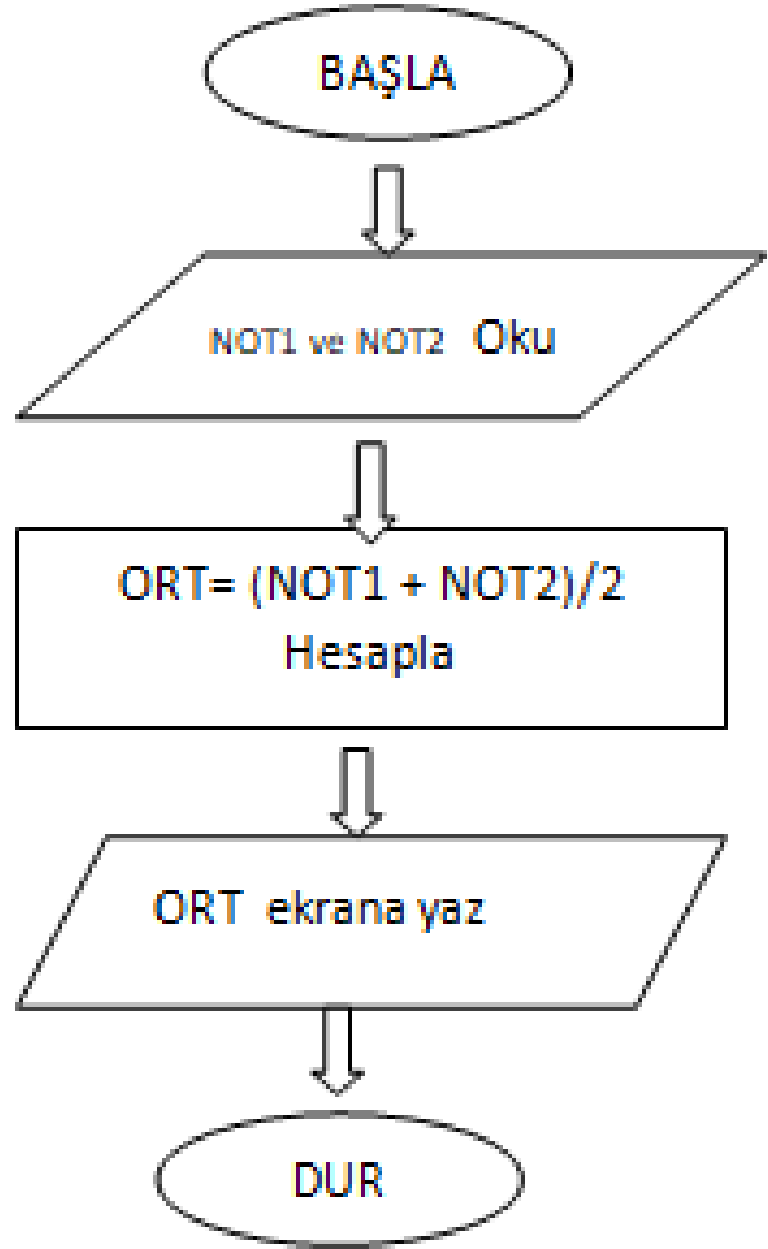
```
B = input('2. Sayı : ')
```

```
C=float(A)*float(B)
```

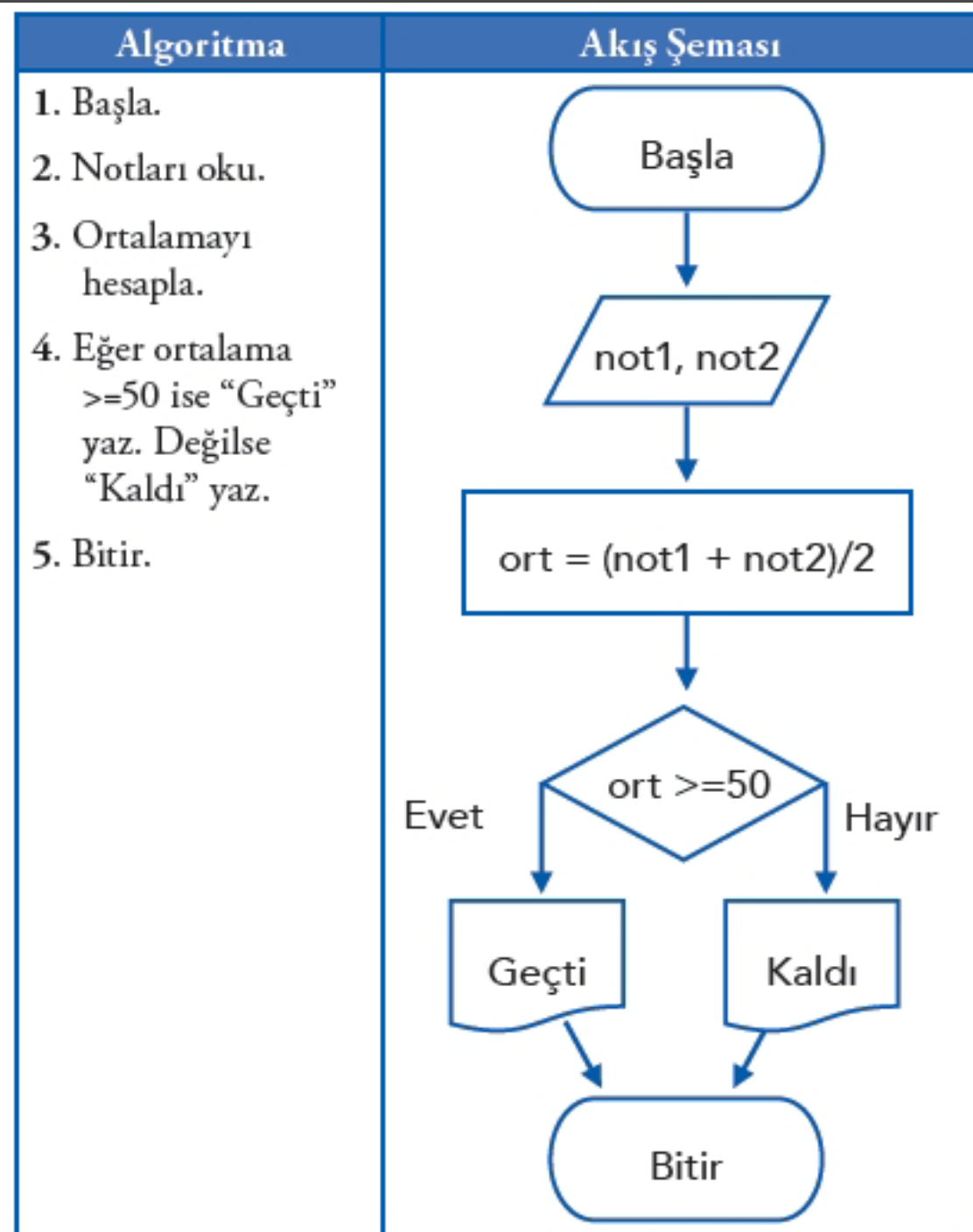
```
print(C)
```

Doğrusal mantık yapısı ile problem çözme, işlemleri sıra ile çalıştırır. Yanda, klavyeden girilen iki sınav puanının aritmetik ortalamasını hesaplayan yapı görülmektedir:

1. BASLA
2. NOT1 VE NOT2 OKU
3. $ORT = (NOT1 + NOT2) / 2$
HESAPLA
4. ORT EKRANA YAZ
5. DUR



Karar yapıları ile problem çözme, iki olasılıktan birini seçmek ve ona göre devam etmek için kullanılır. Yanda, klavyeden girilen iki sınav puanının ortalamasını bularak öğrencinin dersten geçip geçmediğini kontrol eden yapı görülmektedir.



○ Birden fazla karar içeren algoritmaları yazmak için kullanılacak üç tür karar yapısı vardır:

1. Düz mantık,
2. Pozitif Mantık
3. Negatif Mantık

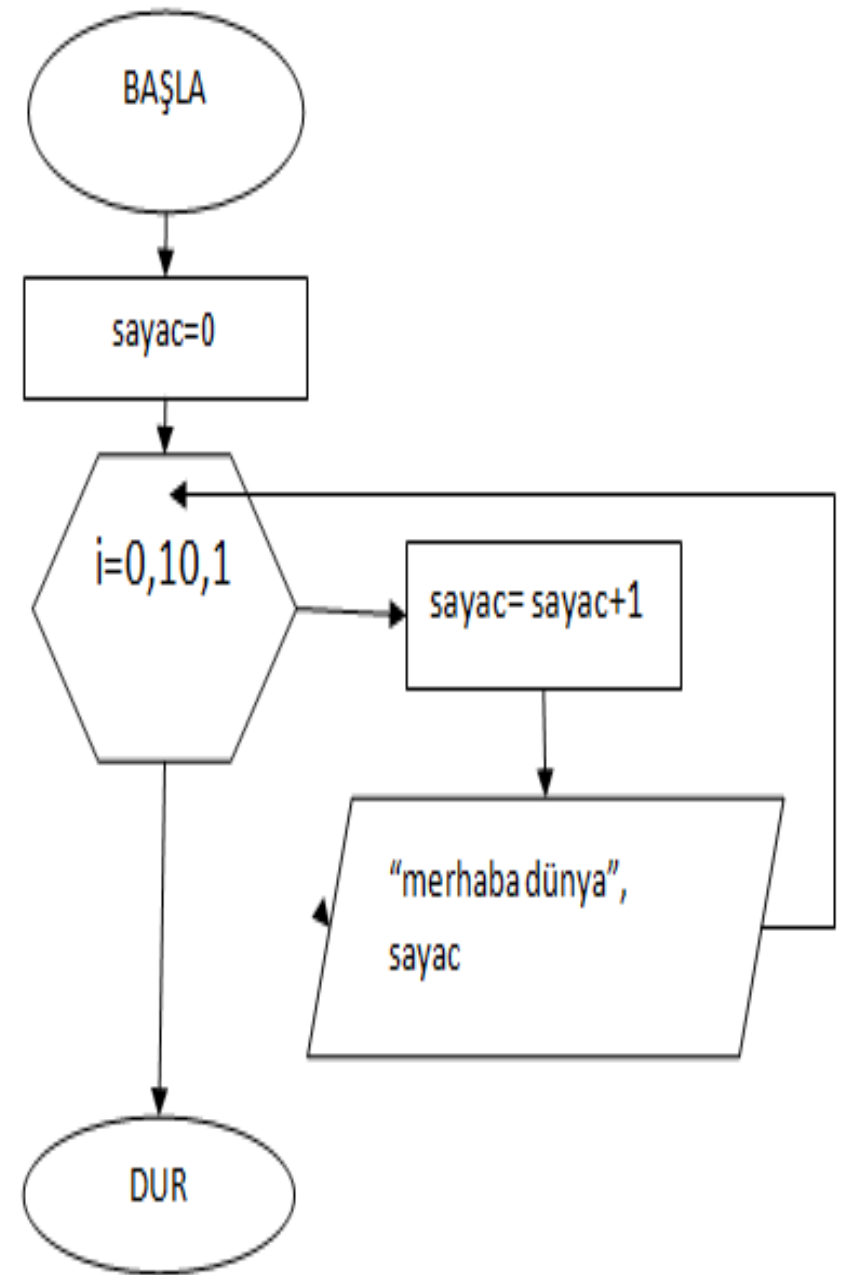
Döngü yapıları ile problem çözme, bir dizi işlemi tekrarlamak için kullanılır.

1. BAŞLA
2. $sayac=0$
3. DÖNGÜ $i=0, 10, 1$
4. $sayac=sayac+1$ HESAPLA
5. "merhaba dünya", sayac YAZ
6. "PROGRAM SONLANDI" YAZ
7. DUR

DÖNGÜ $i=0, 10, 1$: sıfırdan başlayarak 10'a kadar 1'er arttırarak sayar ve her sayma işleminde döngü içindeki işlemleri gerçekleştirir.

0 1 2 3 4 5 6 7 8 9 02468 0369 g a z i a n t e p

$sayac \leq 10$: Koşula bağlı döngüler koşul doğru olduğu sürece döngü içindeki işlemleri gerçekleştirir.



Hata Ayıklama Nedir?

Programlama hatalarını bulma ve düzeltme işlemine **hata ayıklama (debugging)** denilir.

Bir programda üç tür hata oluşabilir: söz dizimsel hatalar, çalışma zamanı hataları ve anlam bilimsel hatalar.



Çalışma zamanı hataları

- Bu hatalar ancak program çalıştırıldıktan sonra ortaya çıkar.
- Hesaplanması mümkün olmayan işlemler (sıfıra bölünme) ya da hiç gerçekleşmeyecek koşulların ($5 < 3$) yürütülmesi gibi durumlarda ortaya çıkar.

```
ilk sayı: 23
ikinci sayı: fdsfd
Traceback (most recent call last):
  File "deneme.py", line 5, in <module>
    ikinci_sayı = int(ikinci_sayı)
ValueError: invalid literal for int() with base 10: 'fdsfd'
```



Anlam bilimsel hatalar

- Program, genellikle hata mesajı vermeden çalışır ancak çoğu zaman beklenen sonucu üretmez. En zor hata ayıklama türüdür.
- Bu yüzden programı satır satır çalıştırarak, farklı adımlardaki çıktıları gözlemleyerek nerede mantık hatası yapıldığını bularak program doğru biçimde çalışana kadar bu hataları ayıklamak gerekir.

