

**Problem Nedir?**

Problem; çözülmesi gereken mesele, soru, sorun veya aşılması gereken engel olarak tanımlanır.

**Problem Çözme Nedir?**

Problemleri formüle edebilme, farklı ve yaratıcı çözüm yolları önerebilme, çözümü kesin ve doğru biçimde ifade edebilme becerisidir.

**Program Nedir?**

Program, yapılacak bir işlemi ya da hesaplamayı gerçekleştirmek için birbirini izleyen komut ya da yönergelerden oluşan yapıdır.

**Temel Program Yapıları:**

- ✚ Girdi
- ✚ Çıktı
- ✚ İşlem
- ✚ Koşullu Yürütme
- ✚ Tekrarlama

**Programlama Nedir?**

Programlama; bilgisayar donanımına nasıl davranacağını anlatan, bilgisayara yön veren komutlar ve işlemler bütünüdür.

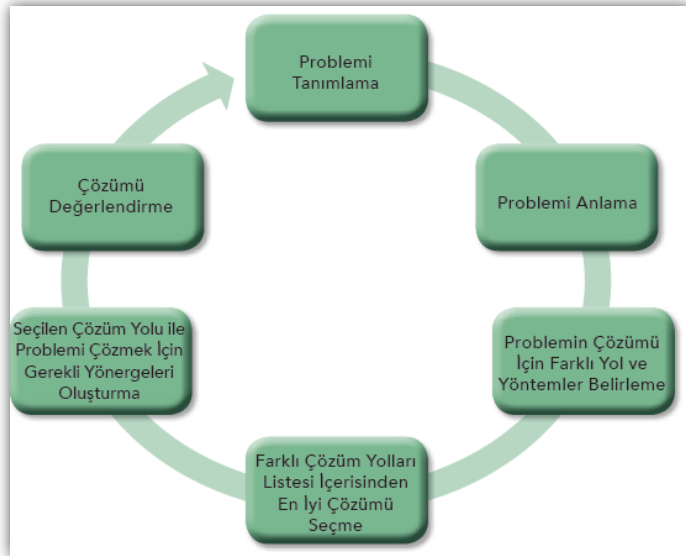
Programlama; hem **problem çözme** becerisi hem de **bilgi işlemel düşünme** becerisine sahip olmayı gerektirir.

**Bilgi İşlemel Düşünme Nedir?**

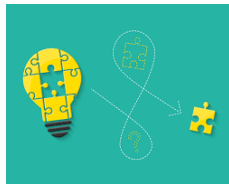
Bilgisayar biliminin kavramlarından yararlanarak problem çözme, sistem tasarlama ve insan davranışlarını anlama olarak tanımlanır.

**Problem Çözme Adımları**

Bir problemi çözerken, aşağıda gösterilen işlem adımları sırayla uygulanmalıdır.

**Problem Çözmede Genel Kurallar**

- ✚ Her Zaman Bir Planınız Olsun
- ✚ Problemi Tekrar İfade Edin
- ✚ Problemi Küçük Parçalara Ayırın
- ✚ Önce Bildiklerinizden Yola Çıkın
- ✚ Benzerlikleri Arayın
- ✚ Deneme Yapın
- ✚ Asla Vazgeçmeyin

**Çözüm Türleri:**

**Algoritmik Çözüm:** Adım adım yönergelere dayalı olan çözümlere denir. Kek yapmak gibi.

**Keşfe Dayalı Çözümler:** Doğrudan işlem adımları ile ulaşılamayan, bilgi ve deneyim gerektiren, bir dizi deneme ve yanılma süreci sonucunda ulaşılan çözümlere denir. Doğru yatırım yapmak gibi.

**SABİT VE DEĞİŞKENLER**

Bilgisayar programlarında problemleri çözme süreci boyunca sabit ve değişken olarak adlandırılan yapılar kullanılır.

**Sabit:**

- ✚ **“Sabit”** olarak tanımlanan veriler problemin çözüm süreci boyunca **asla değişmeyen** değerlerdir.
- ✚ Sabit değerler sayısal, karakter ya da özel semboller olabilir.
- ✚ Program çalıştığı sürece bu değer kendisine verilen **isim** ile çağrılır ve değeri asla değiştirilemez.
- ✚ Örneğin, pi değeri değişmeyen bir değer olacağı için sabit olarak tanımlanmalıdır.

**Değişken:**

- ✚ **“Değişken”** olarak tanımlanan yapıların **değeri, program çalıştığı sürece değişebilir.**
- ✚ Değişkene yeni bir değer atandığında, **eski değer silinir ve yerine yeni değer konulur.**
- ✚ Değişkenlere taşıdığı değerleri ifade eden isimler verilir, bu şekilde belirleyici özellikleri de oluşur.
- ✚ Programcılar çözüm sürecinde ihtiyaç duyulan her bir değişkene ayrı bir isim vermelidir.
- ✚ Örneğin, **fiyat** isimli bir değişkenin içerisinde 50 değeri atanmış olabilir, program çalıştığı süre içerisinde bu değer değişebilir ancak değişkenin ismi değişmez.

**Değişken İsimlendirme Kuralları:**

- Değişkenlere isim verirken boşluk kullanılmaz.
- Değişkenlere isim verirken bir alt tire ( ) veya bir harf karakter ile başlanır. (Rakamla başlanmaz.)
- Bazı programlar desteklemediği için Türkçe karakter kullanımı tavsiye edilmez. (Ç,ç,Ş,ş,Ğ,ğ,Ü,ü,Ö,ö,ı,İ)
- Özel karakterler değişken isimlerinde kullanılamaz. (\*,/, -,+, #,%,&,(= vb.).
- Programlama dillerinde kullanılan komut isimleri değişken olarak kullanılamaz. if, for, while, else, do, int vb.

Yanlış	Doğru
1 sayı	sayi1
Okul No.	okulNo
Soru?	soru

**Değişkenleri İsimlendirme Önerileri**

- ✚ Değişkene içerdiği değer ile tutarlı isimler verilmelidir.
- ✚ İki kelimededen oluşan değişken isimlendirmelerinde boşluk karakteri yerine alt çizgi ( \_ ) karakteri kullanılabilir. Buna “Snake Case” kullanımı denir.

**Örnek:** tc\_kimlik\_no

- ✚ İki kelimededen oluşan değişken isimlendirmede diğer bir yöntem ise; küçük harfle başlanır ve ikinci bir kelimeye, ilk kelimenin hemen ardından büyük harfle devam edilir. Buna “Camel Case” kullanımı denir.

**Örnek:** tcKimlikNo

**BİLGİSAYAR VERİYİ NASIL SAKLAR?**

- ✚ Bilgisayar veriyi hafızada saklar. **Her bir değişken için hafızada belirli bir alan ayrılır** ve bu alan her seferinde tek bir değer saklayabilir.
- ✚ Kullanıcı, var olan değer yerine yeni bir değer atadığında eski değer silinir. Hafızada bu konumlar geçicidir.
- ✚ Programın çalışması bittiğinde ya da bilgisayar kapatıldığında bu veriler silinir.
- ✚ Verilerin daha sonra tekrar kullanılması gerekiyorsa sabit disk gibi kalıcı bir konuma kaydedilmeleri gerekir. Bu şekilde kaydedilen verilere “dosya” adı verilir. Temel anlamda program dosyaları ve veri dosyaları olmak üzere iki dosya türü vardır.

**VERİ TÜRLERİ**

- Çevremizdeki kavram ve nesnelere farklı şekillerde anlamlandırmak için farklı veri türleri kullanırız.
- Çözümler üretebilmek için bilgisayarlar "veri"ye gereksinim duyar.
- Program içinde kullanılan değişkenlerin hangi veri türüyle çalışıyor olduğu mutlaka belirtilmelidir. Bir programda farklı veri türleriyle işlem yapılabilir.

**1. Sayısal Veri**

- Sayısal veriler tüm sayı tiplerini içerir. Pozitif ya da negatif tam sayılar ve reel sayılar kullanılabilir.
- Sayısal veri, hesaplama işlemlerinde kullanılabilen tek veri türüdür.**
- Sayısal veriler; açılar, uzaklık, nüfus, ücret, yarıçap gibi hesaplama sürecinde gerekli değerler için tanımlanır.
- Banka hesap numarası ya da posta kodu gibi sayısal ama hesaplama için kullanılmayan veriler de vardır. Bu tür veriler sayısal olarak tanımlanmaz.

Veri Türü	Veri Seti	Örnek
Sayısal: Tam sayı	Tüm sayılar	66578 -2356
Sayısal: Reel sayı	Tüm reel sayılar ve ondalık sayılar	-56.23 8695.235 0.005

**2. Karakter ve Karakter Dizisi Veri**

- Karakter veri; tüm **tek haneli** sayılar ("0".. "9"), harfler ("a".."z", "A".."Z") ve özel karakterleri ("#", "&", "\*", ..) kapsar.
- Bu veri setinden oluşturulan değer, tırnak içinde belirtilir.
- Büyük ve küçük harf duyarlıdır yani "a" ile "A" farklı algılanır.
- Birden fazla karakter bir araya getirilirse, bu veri "**karakter dizisi**" olarak adlandırılır.
- Karakter ve Karakter Dizisi verileri sadece sayıdan oluşa bile hesaplama işlemlerinde kullanılamazlar.**
- Karakter ve Karakter Dizisi verileri karşılaştırılabilir ve alfabetik sıraya göre sıralanabilir.

Veri Türü	Veri Seti	Örnek
Karakter	Tüm rakamlar, harfler ve özel semboller	"A", "Y", "k", "I", "6", "0", "+", "06"
Dizi	Birden fazla karakterden oluşan kombinasyon	"Bilgisayar", "532-5556633"

**3. Mantıksal Veri**

- Mantıksal veri, veri setinde yalnızca iki kelime barındırır: doğru ve yanlış.
- Bu veri evet ya da hayır şeklindeki karar verme süreçlerinde kullanılır.
- Örneğin elde edilen değer, beklenen değer mi, evli mi, arabası var mı, öğrenci lise mezunu mu gibi sonucu kesin doğru ya da yanlış olan durumlarda mantıksal veri tanımlaması yapılır.

Veri Türü	Veri Seti	Örnek
Mantıksal	Doğru Yanlış True False	Doğru Yanlış True False

**OPERATÖRLER**

Bilgisayar programlarında matematiksel ve mantıksal işlemlerin yapılmasını sağlayan sembollerdir. Bunlar;

- Matematiksel Operatörler
- Mantıksal Operatörler
- İlişkisel Operatörler

olmak üzere gruplanabilir.

**1. Matematiksel Operatörler:**

Operatör	Bilgisayar Sembolü	Örnek	Sonuç
Toplama	+	6.7 + 2	8.7
Çıkarma	-	5.6-3.4	2.2
Çarpma	*	3*4	12
Bölme	/	40/8	5
Mod	%	10%3	1

**2. Mantıksal Operatörler**

Operatör	Bilgisayar Sembolü	Örnek	Sonuç
Değil	NOT	NOT Erkek*	Kız
Ve	AND	Erkek AND 18(yaş)*	Hem Erkek hem de yaşı 18
Veya	OR	Erkek OR Beşiktaş (Tuttuğu Takım)*	- Erkek olabilir - Beşiktaşlı olabilir, - Hem Erkek hem de Beşiktaşlı olabilir.

**3. İlişkisel(Karşılaştırma) Operatörler**

Operatör	Bilgisayar Sembolü	Örnek	Sonuç
Eşit	== (Tek eşit "=" sembolü değişkene değer atamak için kullanılır!)	6 == 8	False
Küçüktür	<	6 < 8	True
Büyüktür	>	6 > 8	False
Küçük ya da eşittir	<=	6 <= 8	True
Büyük ya da eşittir	>=	6 >= 8	False
Eşit değildir	<>	6 <> 8	True

**İşlem Önceliği:**

İşlemler, işlem önceliğine göre yapılmazsa sonuç beklendiği gibi çıkmayabilir. İşlem önceliği şu sırayla gerçekleşir;

- Parantezler () - İç içe parantez varsa en içtekinden başlanır.
- Üslü Sayılar  $x^n$
- Çarpma, Bölme
- Toplama, Çıkarma +,-
- Aynı işlem önceliğine sahip işlemler varsa bunlar soldan sağa doğru yapılır.

**İFADE ve EŞİTLİKLER**

Şu ana kadar gördüğümüz tüm bileşenler, ifade ya da eşitlik biçiminde kullanılmadığı sürece bir anlam ifade etmez.

Çözölmeye çalışılan problem vergi ya da maaş hesaplama, değerleri sıralama, en büyük değeri bulma gibi farklı işlemlerden oluşabilir.

**Örnek: alan=uzunluk \* genişlik**

Bu durumda uzunluk ve genişlik değerlerinin çarpım sonucu hafızada "alan" olarak ayrılan yerde korunur.

**FONKSİYONLAR**

- Fonksiyonlar, belirli işlemleri yürüten ve sonuçları döndüren bir işlem kümesidir.
- Fonksiyonlar, tekrarlanan işlemler için kullanılır ve böylece programcının, hem problemi daha hızlı çözmesini hem de programın daha anlaşılır olmasını sağlar.
- Fonksiyona gönderilen verilere "parametre" denir.
- Örneğin, karekök fonksiyonuna (Sqrt(N)) gönderilen parametre (N değeri) için karekök değeri hesaplanır. Sqrt fonksiyon ismi, N işlem yapılacak parametredir.
- Hazır olarak kullanılan çeşitli fonksiyonlar bulunmaktadır.