ALGORİTMA ve PROGRAMLAMAYA GİRİŞ

Giriş-Çıkış İşlemleri

Giriş- Çıkış İşlemleri

- Tüm programlama dillerinde ortak bazı işlemler vardır. Bu işlemlerden biri giriş-çıkış işlemleridir.
- Giriş(input) – klavye vb. girdi araçlarından veri girilmesidir.
- Çıkış(output) girilen veya işlenen verilerin gerektiği durumlarda çıkış birimlerine iletilmesidir.

Unutmayın !

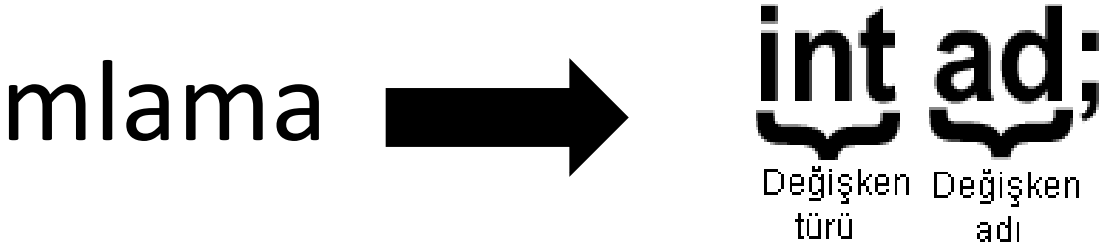
- Kullanıcıdan alınan veriler daha sonra kullanılmak üzere bellekte tutulması gerekir.
- Bellekte bu verilerin tutulduğu yere **değişken** denir.

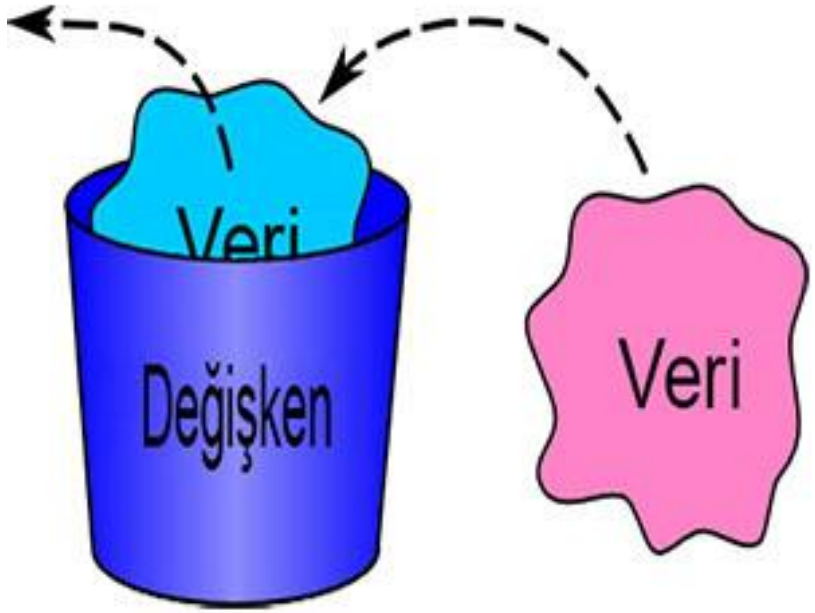
DEĞİŞKENLER ve Veri Tipleri

Değişken Nedir?

- İçinde verileri tuttuğumuz, ismini ve içinde tutulacak veriye göre türünü bizim belirlediğimiz bellek alanlarına **değişken(variable)** denir.
- Değişkenler kullanılmadan önce tanımlanırlar.

Değişken Tanımlama





Değişken tanımlarken, değişkenin içinde tutulacak veriye göre işlemler yapılabileceğinden, değişkenlerin türlerinin olması gerekmektedir.
Örneğin, metinsel tip olarak tanımlanan değişkenle sayısal işlemler yapılamaz. Bu sebeple veri türlerine ihtiyaç duyulmaktadır.

En Sık Kullanılan Veri Tipleri

Integer(int)	Double	String	Boolean
Tam sayı şeklindeki verileri tutmak için kullanılan değişkendir.	Kesirli(ondalıklı) sayı şeklindeki verileri tutmak için kullanılan değişkendir.	İçerisinde metin bulunduran ifadeler string değişkenlerle tanımlanır.	Doğru-Yanlış , 1(var)-0(yok)şeklinde veri tutan değişkenlerdir.
Örnek: int sayi=5000;	Örnek: double sayi=32.5	Örnek: string adi=”ismek”	Örnek: boolean cevap=True

Değişken Tanımlama Kuralları

- İlk karakter mutlaka harf olmalıdır. Rakamlar ilk karakter olarak kullanılamaz fakat ortasında yada sonunda kullanılabilir.

  - Örnek: not, toplam1 (DOĞRU)
    1not, +toplam (YANLIŞ)

- Değişken ismi içerisinde boşluk bulunamaz. Özel karakterler kullanılamaz. Alt çizgi (_) istisna olarak boşluk yerine kullanılabilir

  - Örnek: ad_soyad, adSoyad (DOĞRU)
    ad soyad (YANLIŞ)

Değişken Tanımlama Kuralları

- Değişken isimlerinde Türkçe karakter kullanmamaya özen gösterilmelidir.

  - Örnek: sayi, adi (DOĞRU)
    sayı, adı (YANLIŞ)

- Kullanılan programlama dilinin komutları değişken adı olarak kullanılamaz. Örneğin C# dilinde ekrana yazı yazdırmak için kullanılan writeline komutu kullanılamaz.
- Değişken adları en fazla 255 karakterden oluşmalıdır.

Değer- Değişken-Aktarma

Aktarma : Herhangi bir ifadenin sonucunu bir değişkene aktarma işlemidir.

degisken = ifade

ifade kısmında değer ,değişken veya bir işlemin sonucu olabilir.

= sembolü atama operatörüdür.

Bu operatör, sağda hesaplanan ifadenin değerini degisken içine (sola) iletir.

Tip Dönüşüm İşlemleri

- Bazen farklı veri tipleri ile çalışırken tip dönüşüm işlemlerine ihtiyaç duyulabilir.

Örneğin string bir değişken içinde tutulan veri ile sayısal bir işlem yapılması istenirse, sayısal bir veri tipine dönüştürmek gerekir

string→int

KARAR YAPILARI

Karar-Koşul Yapıları

Bilgisayarlar sadece sıralı işlemleri gerçekleştirmek için tasarlanmamıştır.

Karar-Koşul yapıları programlama dillerinde karşılaştırmalar ve bu karşılaştırmaların sonucunda gerçekleşmesi istenen işlemlerin kontrolü için kullanılmaktadır.

Karar-Koşul yapıları kısaca bir durumun sonucu doğrultusunda işi veya işleri belirler.

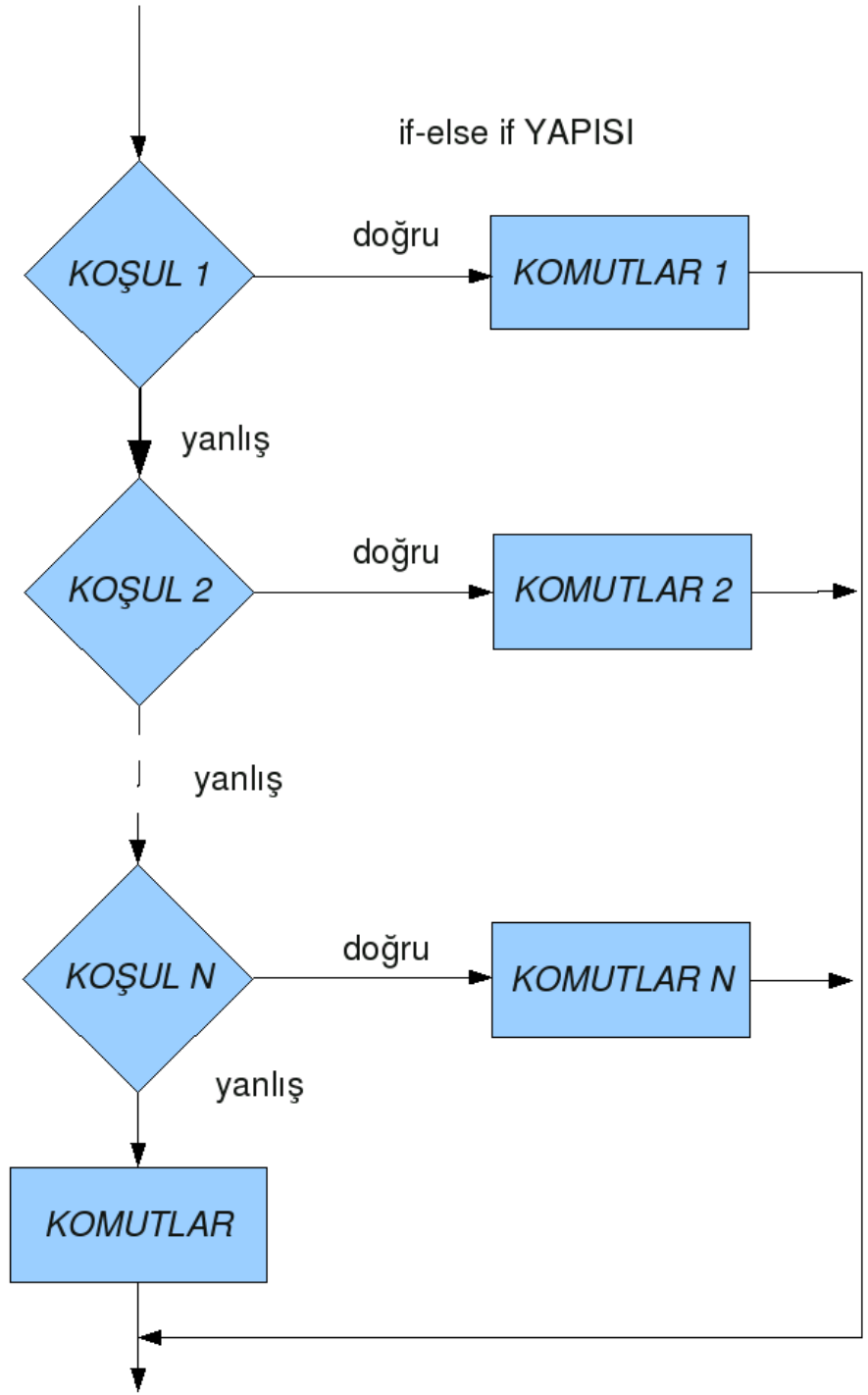

...
Koşul
Evet
Koşulun Sağlanması Durumunda Çalışacak Kod Bloğu
Hayır
Koşulun Sağlanmaması Durumunda Çalışacak Kod Bloğu
...

Örnek:

- Kullanıcının klavyeden girdiği iki sayıdan büyük olanı ekrana yazdıran program.

Örnek:

- Kullanıcıdan istenilen sayının negatif, pozitif ya da nötr olduğunu gösteren program.


if-else if YAPISI
KOŞUL 1
doğru
KOMUTLAR 1
yanlış
KOŞUL 2
doğru
KOMUTLAR 2
yanlış
KOŞUL N
doğru
KOMUTLAR N
yanlış
KOMUTLAR

Örnek

- Kullanıcıdan haftanın kaçıncı günü olduğu bilgisi alınarak, hangi gün olduğunu (Pazartesi, Salı ..) ekrana yazan program.

DÖNGÜLER

Döngü nedir?

Programlamada hangi dil olursa olsun en çok kullanılan yapılardan biri döngü yapılarıdır.

Birden fazla kez tekrarlayan işlemlerde **döngüler** kullanılır.

Kısaca döngüler bir işi, belirlediğimiz sayıda yapan kod blokları olarak düşünebiliriz.

Bir önekle inceleyelim

Ekrana 10 kez ‘Merhaba Dünya‘ yazan bir program yazın.

Bu durumda bu işlemi yapmak için aslında iki yolumuz var.

1. Yol

‘Merhaba Dünya’ yazdıran kodu alt alta 10 kez yazarız ve ekran çıktısı alt alta yazılmış ‘Merhaba Dünya’ yazısı alırız.

2. Yol

Döngü yapısını kullanarak tek bir defa yazarız, döngü burada devreye girip, yazdığımız kodu belirlediğimiz döngü koşulu sayısı kadar tekrarlar.

Bir döngüyü oluşturan 3 ana unsur bulunur;

1. Başlangıç(Başlangıç değeri): Adetli tekrar eden döngülerde başlangıç değeri.
2. Koşul(Bitiş değeri): Döngünün ne zaman biteceğini belirleyen değer.
3. Adım(hareket değeri): Döngünün başlangıç değerinden başlayıp bitiş değerine giderken kaçar kaçar artıp, azalacağının belirlendiği kısımdır.

Unutmayın ! Döngü varsa mutlaka sorgu(koşul) vardır.

Örnek

- Ekrana 10 kez ‘Merhaba Dünya‘ yazan bir program

Örnek

- Klavyeden girilen sayı ile 0 arasındaki sayıları ekrana yazdır.

Örnek

- Klavyeden girilen sayı ile 0 arasındaki çift sayıların toplamını ekrana yazdır.

Programlamada DÖNGÜ İfadeleri

- While döngüsü
- For döngüsü
- Foreach döngüsü

OPERATÖRLER

Operatörler

Programlama dillerinde tek başlarına herhangi bir anlamı olmayan fakat programın işleyişine katkıda bulunan karakterlere **‘operatör’** denir.

3 başlıkta incelenebilir:

1. Matematiksel Operatörler
2. Karşılaştırma Operatörleri
3. Mantıksal operatörler

1. Matematiksel Operatörler

Sayısal işlemler yaparken kullandığımız operatörlerdir.

OPERATÖR	SEMBOL	KULLANIMI	İŞLEMİ
Toplama	+	4+5	4 ve 5'i topla
Çıkarma	-	4-5	4'ten 5'i çıkar
Çarpma	*	4*5	4 ve 5'i çarp
Bölme	/	4/5	4'ü 5'e böl(bölüm alınır)
Mod alma	%	4 mod 5	4'ü 5'e böler fakat kalını alır

2. Karşılaştırma Operatörleri

Bazı durumlarda koşulun doğruluğunu kontrol ederken karşılaştırma operatörlerini kullanırız. Bu operatörlere ilişkisel operatörler de denir.

OPERATÖR	SEMBOL	KULLANIMI	İşlevi
Eşittir	=	X=Y	X' in değeri Y'nin değerine eşittir
Küçüktür	<	X<Y	X' in değeri Y'nin değerinden küçüktür
Büyüktür	>	X>Y	X' in değeri Y'nin değerinden büyüktür
Küçük ya da eşittir	<=	X<=Y	X' in değeri Y'nin değerinden küçük ya da eşittir
Büyük ya da eşittir	>=	X>=Y	X' in değeri Y'nin değerinden büyük yada eşittir

3. Mantıksal Operatörler

Bazı durumlarda, kontrol edeceğimiz koşul, tek bir olaya bağlı olmayabilir. Birden fazla koşulu sorgulamak istediğimiz durumlarda mantıksal operatörler kullanılır. Farklı programlama dillerinde semboller değişebilir fakat kullanımları ve mantıkları aynıdır.

OPERATÖR	ÖRNEK SEMBOL	KULLANIMI	İŞLEMİ
And	&&	x>y && y>z	x değeri y değerinden büyüktür ve y değeri z değerinden büyüktür
Or	||	x>y || x>z	x değeri y değerinden büyüktür veya x değeri z değerinden büyüktür

VE, VEYA operatörlerini kullanırken dikkat edilmesi gerekenler;

VE (AND)

A	B	SONUÇ
YANLIŞ	YANLIŞ	YANLIŞ
YANLIŞ	DOĞRU	YANLIŞ
DOĞRU	YANLIŞ	YANLIŞ
DOĞRU	DOĞRU	DOĞRU

VEYA (OR)

A	B	SONUÇ
YANLIŞ	YANLIŞ	YANLIŞ
YANLIŞ	DOĞRU	DOĞRU
DOĞRU	YANLIŞ	DOĞRU
DOĞRU	DOĞRU	DOĞRU

and operatörü kullanılırken «DOĞRU» sonucuna ulaşılması için her iki koşulunda «doğru» olması gerekir.
or operatörü kullanılırken tek bir koşulun «doğru» olması «DOĞRU» sonucuna ulaşılması için yeterli olmaktadır.

Örnek-1

- Girilen iki sayının 4 işlemini yaparak ekrana yazdıran program. (Toplama, Çıkarma, Çarpma, Bölme)

Örnek-2

- Kullanıcıdan iki açısı istenen üçgenin üçüncü açısını hesaplayan ve sonucu ekrana yazdıran program.

Örnek-3

- Girilen bir sayının tek mi çift mi olduğunu kontrol ederek ekrana yazan program.

Örnek-4

- Vize ve final notu girilen öğrencinin hem vize notu hem final notu 30 ve üzeri ise ekrana «geçti» değilse «kaldı» yazdıran program.

METOTLAR

Metot Nedir?

- Program içerisinde bir işi, bir görevi yerine getirmek için yazılmış kod bloklarına **metot** denir.
- Metotlar alt programlardır. Uygulama içerisinde çağrılana kadar herhangi bir işlem yapmazlar.
- Her metot bir isim ve çalıştırılacak komutların bulunduğu kod bloğundan oluşur.
- Programlama dillerinde hazır metotlar olduğu gibi ihtiyaca göre yeni metotlar da yazılabilir.

Hazır Metot Örnekleri

- Matematiksel Metotlar .. (kök alma, üs alma, faktöriyel vb.)
- Metinsel Metotlar .. (büyük-küçük harfe çevirme vb.)
- Tarih metotları… (sistemdeki tarihi çekme vb.)

Örnek (Matematiksel Metot Örneği)

- Yarıçapı girilen bir dairenin çevresini hesaplayan program

Örnek (Metinsel Metot Örneği)

- Klavyeden girilen bir ismi, büyük harfe çevirerek ekrana yazan ve ismin karakter sayısını bulan program.

Örnek (Tarihsel Metot Örneği)

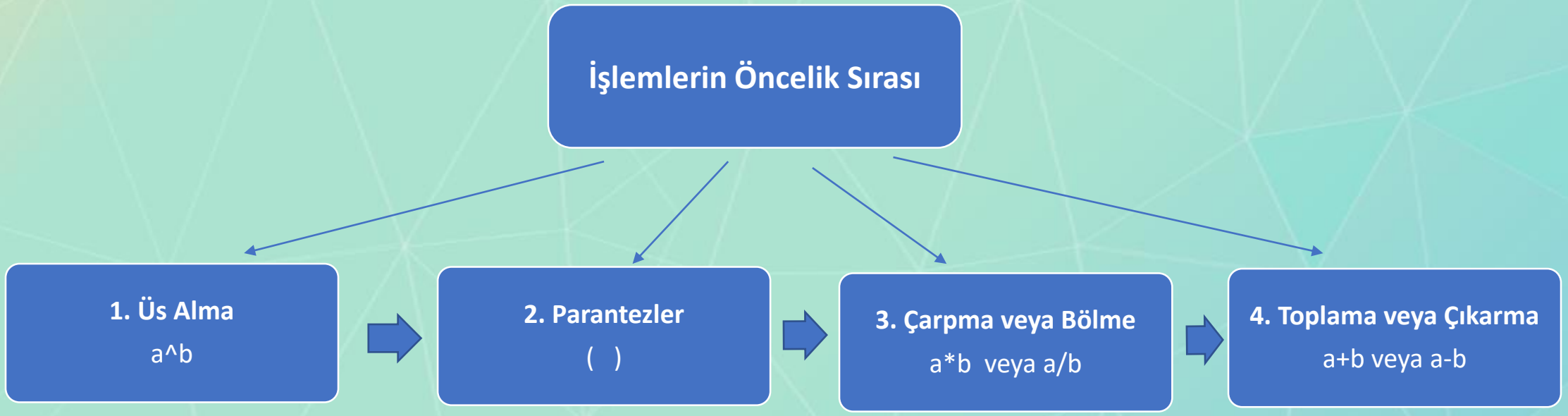
- Sistemden yıl bilgisi alınarak, kullanıcıdan alınan yıla göre kişinin yaşını hesaplayarak ekrana yazdıran ve eğer yaş 18 ve üzeri ise ekrana «Reşit» değilse ekrana «Reşit Değil» yazan program.

İŞLEM ÖNCELİĞİ

İşlemlerin Öncelik Sırası

Matematiksel işlemler içeren kümenizin içindeki matematiksel işaretlerin bir **işlem önceliği** vardır.

İşlemlerin Öncelik Sırası



İşlemlerin Öncelik Sırası

Sıra	İşlem	Bilgisayar Dili
1	Üs alma	a^b
2	Parantezler	(…….)
3	Çarpma ve bölme	a*b ve a/b
4	Toplama ve Çıkarma	a+b ve a-b

NOT:

Bilgisayar diline kodlanmış bir matematiksel ifadede, aynı önceliğe sahip işlemler mevcut ise bilgisayarın bu işlemleri gerçekleştirme sırası soldan sağa(baştan sona) doğrudur.

Örneğin ;

$$Y = A*B/C$$

İşlem önceliğinde aynı sırada olan çarpma ve bölme işlemleri yan yana bulunuyorsa **önce A*B işlemi** yapılacak, **ardından** bulunan sonuç **C ye bölünecektir.**

Matematikte İşlem önceliği Nedir?

Aritmetikte işlemler belirli kurallara göre uygulanır. Önce en iç parantezden başlanarak dışa doğru hesaplamalar yapılır.
Parantez yoksa soldan sağa doğru aşağıdaki sıralamaya göre işlemler yapılır.

Tüm sayı kümelerinde dört işlem yapılırken şu sıra takip edilir:
1. Parantezler ve kesir çizgisi işleme yön verir.
2. Üslü işlemler varsa sonuçlandırılır.
3. Çarpma - bölme yapılır.
4. Toplama - çıkarma yapılır.

Toplama ile çıkarma işlemi kendi arasında öncelik taşımaz. Aynı şekilde çarpma ile bölme işlemi de kendi arasında öncelik taşımaz.
Özelikle çarpma ile bölme de öncelik söz konusu ise bu parantezle belirlenmiştir.