Değerleri bilgisayarda ikili olarak tutarız

(00000000) toplamda 8 adet ikili değer 2 ^8 kadar veri tutabiliriz.

( 0 0 0 0 0 0 0 0 )

En anlamlı 🡪en anlamsız

Değişkenler ikiye ayrılır

**1.ilkel veri tipleri (int char bool etc.)**

2.kullanıcı tanımlı(struct vb veri tipleri)

Değişkenlerin bellekte hangi bölgede tutulduğu sınav

**VERİ TİPLERİ**

-Piyasada büyük projelerde programlama diline göre işi veren seçiyor.

Unsigned değerler 0dan başlıyor. (**char int ve long veri tiplerinde kullanılabilir**.)(javada unsigned yok..)

İnt veri tipi -2^16 +2^16-1 e kadar. Çünkü o -1 0 dır ve pozitif alınır.

İnt veri tipi dillerde şöyle tanımlanabilir. C# ta denedim.

İnt as = 333\_444\_021

**Float ve double arasında bir talkım farklılıklar var**

Float : 10.33333333334// yuvarlar

Double:10.3333333333326 // yuvarlamaz hassastır.

Decimal: 10.333333333333333333333333333333333333333332

Double floata göre daha hassastır.

C# Decimal veri tipi:doubleden de daha hassastır.

En baştaki bir + veya – oldudğunu gösterir.

00000000

Bool 1 byte dir 8 bit 1 bit yer yetmesine rağmen 1 bytedir. Çünkü işlemesi daha kolay.

İnt a = 33; inta a = -22 int a=0;

İf(a){}

//0 harici tüm değerler false dönderir içeri girmez.

String = char[] = {‘a’,’b’}

**Türetilmiş veri tipleri**

Diziler ardarda gelen elemanlardır. Bellekte ardarda yazılır.(bir bütün halinde anlı yon mu )

**Array VS List**

**Array,list gibi tipler birbirinin adresini tutarak yerleşirler belleğe.yani diziler gibi(string dizisi int dizisi) gibi bellekte ardarda sıralanmazlar. Birbirilerini bulmak için list yapıları, ekstradan adres bilgilerini barındırırlar içlerinde. BU DA PROGRAMIN MALİYETİNİ ARTTIRIR. Bununla birlikte diziler kullanılınca diğer tüm dizi elemanları ilk adresten sonra geldiği için tüm dizi elemanları ilk adrese göre belirlenir(aslında diziyi newleyince adres bilgisi elimizde olur). Maliyeti arttırmamak için elimizden geldiğince dizileri kullanmalıyız.**

New Dizi[10]//bellekte 10 elemanlı dizi oluştur.

Pop[9] diyebiliriz

İnt sayılar[] ={2,2,2,2,}

Sayılar bellekte ilk adresi tutar.

İnt [] a= new int[10];//yeni bir dizi yapmadan bu dizi boyutu asla ve asla hiçbir dilde değiştirilemez

**Union(ortaklık) veri timpi(class ve struct gibi())(bellekte bir kere tanımlanır.)**

**Classta oluşturunca her nesne için class içinde ne kadar yer gerekioyrsa o kadar yer ayrılır.**

**Structada yer ayrılır**

Ama unionda nesne1.id = 10

Nesne2=20 dersek nesne1.id de 20 olur

Static

Asci -> UNİCODE

(2^7 alan) (2^16 = 65536 kadar değer alır)

(java gibi dillerde

2 byte yer ayrılmıştır )