



YOZGAT BOZOK ÜNİVERSİTESİ

Programlama Dilleri

Prensipleri

Dr. Öğr. Üyesi R. Sinan ARSLAN



- Değişkenlerin İsim önceliği
- değişken tipi
- sabitler
- işlemciler
- işlemci yükleme
- atama deyimi



Temel Programlama Kavramlar

- Bir yada birden fazla bellek alanının soyutlanması için değişken kullanmak gereklidir.
- Her değişken isim, adresi, değer, tip, yaşam süresi ve kapsam özelliklerine sahiptir.
- Programda bir değişkene atanan isim onun kullanıldığı tüm yerlerde çağrı için kullanılır.
- aynı isimli değişkenlerin programın farklı yerlerinde farklı adreslere bağlanabilmesi mümkün olduğu için isimleri aynı olsa da bu değişkenler farklı birer tanım gibi düşünülmelidir.
- Benzer şekilde farklı isimli değişkenlerinde aynı adres alanını işaret etmesi de bazı programlama dilleri için mümkündür. Bu noktada fazla sayıda değişkenin aynı adres alanını işaret etmesi sorunları beraberinde getirir.



Temel Programlama Kavramları

- Fortranda equivalence kavramı ve C dilinde Union kavramı farklı değişkenlerin aynı adres alanını kullanmasını tanımlar.

```
union bozok {
```

```
    char a[3], b[3];
```

```
};
```

x.a[0] = 'a'; denildiğinde x.b[0]'da aynı değeri alacaktır.

- Bir değişken kullanılmadan önce mutlaka tanımlanmış olmalıdır. Bu tanımlardan çalışma zamanından önce tanımlanmış ve çalışma boyunca değiştirilemez olanlarına statik bildirimler denilir. Çalışma zamanından ortaya çıkan ve değişebilen bildirimlere dinamik bildirim denilir.
- Bir değişken tanımlandıktan ve tip bildirimi yapıldıktan sonra bir bellek alanı ile ilişkilendirilir. Değişkenin bu bellek alanı ile bağlı kaldı süre yaşam süresi olarak adlandırılır. Değişken program içerisinde hangi alanlarda geçerli ise geçerli olduğu alan kapsamını belirler.



değişkenlerde isim özelliği

- Değişkenlerin tanımlanması, erişim sağlanması için ona bir isim verilmesi şarttır.
- Bu isimlendirme özelliği program içerisinde etiketler, alt programlar, formal parametreler ve bir çok program yapısında kullanılır.
- Değişkenlerde isimlendirme için:
 - isim uzunluğu en fazla ne kadar olabilir.
 - hangi karakterler kullanılamaz.
 - büyük küçük harf duyarlılığı
 - özel kelime, ayrılmış kelime ve anahtar kelime özelliklerine dikkat edilmelidir.

değişkenlerde maksimum uzunluk

- isimlendirme da ilk dönemde tek karakterli isimlendirme kullanılırdı.
- Günümüzde ise çok daha uzun isimlendirmeler kullanılmakta ve uzun isimleri bölmek için genelde _ işareti kullanılmaktadır.
- kısa bir isimlendirme, değişkenin özelliğini tanımlamaya yeterli olmayacağı için tercih edilmemelidir.



değişkenlerde maksimum uzunluk örneği

Programlama Dili	Maksimum uzunluk izni
fortran 1	6
Java	limit yok
c++	derleyici göre değişmektedir. genelde limit yoktur.
lisp	limit yok
cobol	30



Büyük küçük harf duyarlılığı

- çoğu programlama dilinde büyük küçük harf duyarlılığı yoktur.
- Ancak c, c++, java gibi dillerde vardır.
- örneğin bozok ile BOZOK isimleri farklı değişkenleri ifade edecektir.
- Bu durum gerçek anlamları aynı olan değerlerin programda farklı anlam ifade etmesi sebebiyle kodun okunabilirliğini ve anlaşılabilirliğini düşürür.



Özel Kelimeler

- Anahtar sözcükler-keyword- yalnızca belirli bağlamlarda kullanılan özel kelimelerdir.
- Bu sözcükler tanımlayıcı olarak kullanılamazlar.
- do, for, while C dili için, procedure, begin, end PASCAL için ayrılmış kelimelere örnektir.
- C dilinde isimlendirme özellikleri:
 - isimler ingilizce büyük ve küçük harfler, rakamlar ve altçizgi işaretinden oluşabilir.
 - ilk karakteri rakam olamaz.
 - ilk 31 karakter anlamlıdır.
 - büyük küçük harf ayrımı vardır.
 - int, main, void gibi özel kelimeler olamazlar.



Değişken tipi

- bir değişkenin alabileceği değer aralığı ve bu değişken üzerinde hangi işlemlerin yapılabileceğine o değişkenin tipine göre karar verilir.
- veri tipleri, temel veri tipleri, türemiş veri tipleri ve kullanıcı tanımlı türler olarak incelenebilir.
- tamsayı, mantıksal, karakter, karakter dizisi temel veri tiplerine örnektir. tüm programlama dillerinde kullanılan basit tiplerdir.
- türemiş tipler, çeşitli veri tiplerine olabilen yapıdadırlar.
- yapısal tipler, tipin bileşenlerini oluşturur ve dilin okunabilirliği ve ifade gücüne olumlu katkı yapar.



Değişken tipi

- C dilinde temel veri tiplerinin alt üst sınırları ve kapladıkları bellek alanları aşağıdaki gibidir.
 - char, -128, +127, 1 byte
 - short, -32768, +32767, 2 byte
 - int, -2147483648, +2147483647, 4 byte
 - long, -2147483648, +2147483647, 4 byte
 - double, -1.7×10^{-308} , $+1.7 \times 10^{+308}$, 8 byte
- bazı veri tiplerinin farklı isimli olmasına rağmen aynı alanı kaplaması ve aynı aralığı kullanması derleyiciye göre değişkenlik gösterecektir.



Değişken tipi

- c++ dilinde kullanıcı tanımlı veri tipleri enum, struct, class gibi yapılardır. C dilinde enum ve struct yapısıdır.

```
struct deger{
```

```
    int sayi;
```

```
    deger() {}
```

```
    int sayi() const{
```

```
        return sayi;
```

```
    }
```

```
};
```

C++ dilinde struct yapısı
örneği



Sabitler

- Belirli bir tipteki bir değerin kodlanmış gösterimi içeren ancak program çalıştırılması sırasında değiştirilemeyen bellek hücresine sabit denir.
- Bellek hücresi ilişkisi değer ilede kuruluyor ise, isimlendirilmiş sabit denir.
- Eğer tekrarlı bir kullanım söz konusu ise isimlendirilmiş sabit kullanımı daha uygundur. (PI = 3.14 gibi)
- Sabit tanımlaması her dilde bulunmak ile birlikte kullanım farklılığı vardır. C++ dilinde sabit tanımlandığı yerde değeri de atanması gerekli iken, Java da böyle bir zorunluluk yoktur.
- Kullanım şekilleri aşağıdaki gibidir:
 - Pascal const,
 - c/C++'da #define, const
 - Lisp defconstant
 - Java final



Programlama Dillerinin İşlemcileri

- Genel Özelliklere göre
 - İşlenen sayısı
 - işlemcinin yeri
 - işlem önceliği
 - birleşme özelliği
- işlenenlere göre
 - sayısal işlemciler
 - ilişkisel işlemciler
 - mantıksal işlemciler



Genel Özellikler-İşlenen sayısı

- Bir işlemcinin sonuç üretmek için kaç operand gerektirdiğini ifade eder.
- tekli(unary), ikili(binary), üçkü(ternary) olabilir.
- $-(-20, -35)$, $\&$ (adres gösterimi) tekli işlemcilere örnek verilebilir.
- $+, -, *, /$ ikili işlemcilere örnek olarak verilebilir.
- c, c++, java da kullanılan $?:$ operatörün de ternary operatöre örnektir.
 - $a = (c=0)?0:s/c$



Genel Özellikler-İşlemcinin yeri

- genel olarak operandlar arasında işlemleri koyma genel bir kural olsa da önüne veya arkasında koyulabilir.
- $++x$, $--y$
- $x++$, $y++$
- $x = x+1$, $y = y-1$



Genel Özellikler-İşlem önceliği

- birden çok işlemcinin aynı anda kullanıldığı durumlar parantez yok ise hangisinin daha önce işleneceğine işlem önceliği denir.
- $a+b*c$, $a+b+c$
- öncelik ve birleşme kavramları ile birlikte değerlendirilmelidir.
- her programlama dilinde bir işlem önceliği sırası vardır.
- c,c++,java için yüksek öncelikten düşük önceliğe doğru sıralama:
 - ++,--(postfix)
 - ++,--(prefix)
 - +,-(işaret)(+50, -30)
 - *,/,%
 - +,-(binary)
- Ada
 - **,abs
 - *,/,mod
 - +,-(işaret)
 - +,-(binary)



Genel Özellikler-Birleşme Özelliği

- $x+y+z$ veya $a*b*c$ gibi bir ifadenin değerlendirilmesi için birleşme özelliği kuralları kullanılmalıdır.
- eğer aynı öncelik seviyesinde birden fazla operatör yanyana kullanılmış ise hangi sırada yapılacağı programlama dilinin birleşme özelliğine göre belirlenir.
- sağ veya sol birleşmeli olarak belirlenebilir.
- Programlama Dillerinde birleşme kuralları
 - c/C++
 - $++$ (postfix), $--$ (postfix), $*$, $/$, $\%$, $+$ (ikili), $-$ (ikili) \rightarrow sol birleşmeli
 - $++$ (prefix), $--$ (prefix), $+$ (tekli), $-$ (tekli) \rightarrow sağ birleşmeli
 - ada
 - $**$ dışında tüm operatörler \rightarrow sol birleşmeli
 - $**$ birleşme özelliği yoktur.



Nitelğine göre işlemciler-Sayısal işlemciler

- sayısal işlemlerde kullanılırlar.
- her programlama dilinde farklı karakterler ile ifade edilebilir.
- C dilinde:
 - * çarpma, / bölme, % mod, + toplama, - çıkarma
- işleme giren işlemci türlerinden en büyük tür hangisi ise sonuçta o tür olarak belirlenecektir.

```
int a = 3, b = 2;
```

```
double sonuc = a/b;
```

```
printf("%d", sonuc); -> çıktı 1 olacaktır.
```

DİKKAT: Her ne kadar $3/2 = 1.5$ olsa ve sonuc double bir değişkene atansa bölünen değerlerin her ikisinde int olduğundan sonuç int olacaktır. !!



Nitelğine göre işlemciler-İlişkisel İşlemciler

- İki değeri karşılaştırma için kullanılır.
- sonuç doğru veya yanlıştır.
- char, int , float gibi veri tipleri veya sınıflar karşılaştırılabilir.
- C/C++
 - > büyüktür, < küçüktür, == eşittir, != eşit değildir, >= büyük eşit, <= küçük eşit
- Pascal
 - > büyüktür, < küçüktür, = eşittir, <> eşit değildir, >= büyük eşit, <= küçük eşit
- Ada
 - > büyüktür, < küçüktür, = eşittir, /= eşit değildir, >= büyük eşit, <= küçük eşit
- Java dilinde bunların haricinde equals metodu da kullanılmaktadır. nesnelerin karşılaştırılması dahil olmak üzere tüm karşılaştırmalarda kullanılabilir.
- (0) -> false, (44) -> true değerini döndürür. bunun dışında karşılaştırma sonucuna göre karar verilir.



Nitelğine göre işlemciler-Mantıksal İşlemciler

- Birden çok koşulun birleştirilmesi için kullanılırlar.
- AND, OR, NOT, XOR gibi operatörler örnek olarak verilebilir.
- C, Java
 - $\&\&$ \rightarrow AND
 - $||$ \rightarrow or
 - $!$ \rightarrow not
- mantıksal işlemciler, mantıksal işlenenleri(boolean) olarak mantıksal değer oluşturular(doğru, yanlış).
- Mantıksal operatörler de işlem önceliği çoğu dilde NOT, AND, OR şeklindedir.
- $a+10>c*2$ ifadesinde önce sayısal işlemler yapılmalı sonra ilişkisel işlemler yürütülmelidir. sayısal işlemler her zaman ilişkisel işlemlerden önceliklidir.



Nitelğine göre işlemciler-Mantıksal İşlemciler

- Genel öncelik sırası aşağıdaki gibidir. (Yüksekten düşüğe doğru)
 - $!$, $++$, $--$, $+$, $-$ (tekli operatörler)
 - $*$, $/$, $\%$, $+$, $-$ (aritmetik)
 - $>$, $<$, \leq , \geq , $==$, $!=$ (ilişkisel)
 - AND $\&\&$, OR $||$ (mantıksal)
 - $?:$ (koşul)
 - $=$, $+=$, $-=$, $*=$, $/=$, $\%=$ (atama)
- Örnekler:
 - $-a+b$, $a*(b-14)/c \bmod d$, $-a \text{ Or } c=d \text{ and } e$, $x>b \text{ xor } c \text{ or } d\leq 45$ ifadelerinin sonucu nedir. (a, b,c,d için farklı değerleri ile test edilebilir.)



İşlemci Yükleme

- İşlenenlerin sayısı, tipine göre işlemcilerin anlamları değişebilir.
- örneğin + aritmetik toplama anlamına gelebildiği gibi bazı dilleri karakter katarlarını birleştirmek içinde kullanılabilir.
- - hem çıkarma, hemde negatif sayı ifade etmek anlamına gelir.
- Bu duruma operator overloading denir. Bu durum derleyicinin gücünü artırırken okunabilirliği azaltır.



Atama Deyimi

- Emir esaslı programlama dillerinde en temel sıralı işlem deyimi atama deyimidir. atama deyimi ile sağ tarafın içeriği sol tarafa aktarılmak istenir.
- $\langle \text{hedef de\u011fi\u015fen} \rangle \langle \text{atama_i\u015flemcisi} \rangle \langle \text{ifade} \rangle$
- $\langle \text{ifade} \rangle$ 10 veya $a*b$ gibi bir de\u011fer iken, $\langle \text{hedef de\u011fi\u015fen} \rangle$ bir bellek adresini g\u00f6sterir.
- c, C++, java gibi dillerde $\langle \text{atama i\u015flemcisi} \rangle =$ iken farklı dillerde farklı kullanımlar olabilir.
- Atama i\u015fleminde iki tarafın tip uyumu olması de dikkat edilmesi gereken bir problemdir.



Atama Deyimi Özellikleri

ÇOKLU HEDEFLİ ATAMA

- Bazı programlama dillerinde sağ taraftaki değer sol taraftaki birden çok değişkene atanabilir.
 - $t1, t2 = 0 \rightarrow OL$
 - $t1 = t2 = 0 \rightarrow java$
 - $sayi\ s1, s2, s3 ; s1 = s2 = s3 = new\ Sayi(100); \rightarrow$ şeklindeki kullanımda 1 nesne türetilir ve 3 değişkene referans olarak atanır.

KOŞULLU HEDEFLER

- atama 3'lü operatör kullanımı ile koşula göre yapılabilir. C, C++, Java gibi dillerde bu mümkündür.
 - $s = (a \% 2 == 0 ? "çift" : "tek");$

BİLEŞİK ATAMA

- bir atama işlemcisi ile bir ikili işlemci birleştirilerek bileşim atama işlemcileri oluşturulur. $+=, *=, /=, \%=$ örnek olarak verilebilir.
 - $t += 100;$



Atama Deyimi Özellikleri

TEKLİ ATAMA İŞLEMCİLERİ

- tek bir değişkene uygulanır.
- $i = +1, j = -1$ gibi.
- $+$, $-$ değerleri pozitif veya negatif sayı olduğunu gösterir.
- $++$, $--$ operatörleri de tek değişken üzerinde işlem yaparlar.
- örnekler
 - $k = 'A'$ için $k++ \rightarrow B$ değerini üretir.
 - $\text{int } x = 10; \text{int } y = -++x \rightarrow y$ değerine -11 değeri atanır.



Sorular

- Java dilindeki anahtar kelimelere 10 tane örnek verip kod içerisinde kullanınız.
- Java dilinde işlemci yüklemesinin olmamasının nedeni nedir ?
- c++ da isim kurallarını açıklayınız.

`int x= 1;`

`sysout (x++ + ++x * x--/--x)` değerinin sonucu nedir.