315 Programlama Dilleri

Yrd. Doç. Dr. Ahmet Arif AYDIN

Özet

- Associative arrays (hash- dictionary)
- Record
- Tuple
- Union
- Pointer
- Reference

Expressions and assignment statements

(İfadeler ve atamalar)

Giriş

- Programlama dillerinde yapılacak işlemler ifadeler(expressions) ile gerçekleştirilir
- Programcıların programlama dillerinin Syntax ve Semantiğini anlaması gerekmektedir
- BNF ile programlama dillerinin syntax yapısı tanımlandı.
- Bu bölümde ise ifadelerin manası üzerinde durulacaktır.

Aritmetik İfadeler

- Programlama dillerinde bulunan aritmetik ifadelerin kullanım biçimi ve karakteristikleri matematikten kalıtsal olarak (inheritance) devralınmıştır.

Aritmetik İfadeler

- Programlama dillerinde aritmetik ifadeleri oluşturan elemanlar
  - Operatörler
  - Terimler (operands)
  - Parantezler
  - Fonksiyon çağrıları

Aritmetik İfadeler

- Programlama dillerinde operatörler üzerinde işlem gerçekleştirdikleri terim sayısına göre üçe ayrılırlar :
  - Unary (bir adet terim)
  - Binary (2 adet terim)
  - Ternary (3 adet terim)

Aritmetik İfadeler

Aritmetik ıfadeler bir aritmetik işlemi tanımlamak işin yazılırlar.
Aritmetik bir ifadenin uygulanması :

1. Aritmetik ifadede bulunan terimlerin hafızadan alınması

2. Terimler uzerinde belirtilen işlemi gerçekleştirilmesi

Operatör Değerlendirme Önceliği

Precedence (öncelik)

a + b * c a=2 , b= 3 , c =5

Soldan sağa işlem yapılırsa 25 Sağdan sola işlem yapılırsa 17

Operatör Değerlendirme Önceliği

Precedence (öncelik)

a + b * c a=2 , b= 3 , c =5 **Sonuc 17**

Bu işlem yapılırken operatör önceliği dikkate alınarak işlem gerçekleştirilir.

Matematikte çarpma operatörünün toplama operatörüne göre önceliği bulunmaktadır

Operatör Değerlendirme Önceliği

Precedence (öncelik)

Programlama dillerinin bir çoğunda (imperative dillerde) operatör önceliği matematikte tanımlanan öncelik ile aynıdır.

1. üs alma işlemi (exponentiation)
2. Carpma ve bolme (multiplication and division)
3. Toplama ve cıkarma (addition and suntraction)

Operatör Değerlendirme Önceliği

a + (- b) * c

legal

a + -b * c

illegal

Operatör Değerlendirme Önceliği

unary minus operator

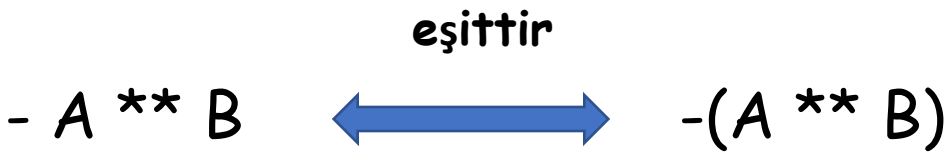
```
- a / b

- a * b

- a ** b
```

Operatör Değerlendirme Önceliği

Fortran, Ruby, Visual Basic, ve Ada dillerinde üs alma operatörü bulunmaktadır.



Operatör Değerlendirme Önceliği

Fortran	Pascal	C	Ada
		++, -- (post-inc., dec.)	
**	not	++, -- (pre-inc., dec.), +, - (unary), &, * (address, contents of), !, ~ (logical, bit-wise not)	abs (absolute value), not, **
*, /	*, /, div, mod, and	* (binary), /, % (modulo division)	*, /, mod, rem
+, - (unary and binary)	+, - (unary and binary), or	+, - (binary)	+, - (unary)
		<<, >> (left and right bit shift)	+, - (binary), & (concatenation)
.eq., .ne., .lt., .le., .gt., .ge. (comparisons)	<, <=, >, >=, =, <>, IN	<, <=, >, >= (inequality tests)	=, /= , <, <=, >, >=
.not.		==, != (equality tests)	
		& (bit-wise and)	
		^ (bit-wise exclusive or)	
		| (bit-wise inclusive or)	
.and.		&& (logical and)	and, or, xor (logical operators)
.or.		|| (logical or)	
.eqv., .neqv. (logical comparisons)		?: (if . . . then . . . else)	
		=, +=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |= (assignment)	
		, (sequencing)	

Programming Language Pragmatics

Operatör Değerlendirme Önceliği

Associativity (Birleşirlik)

Programlama dillerinde operatorlerin sag veya sol birleşirlik özelliği bulunmaktadır

Java

a – b + c

Programlama dillerinde genellikle aynı seviyedeki operatörler soldan sağa değerlendirilir

A ** B ** C

Üs alma işlemi sagdan sola degerlendirilir Fortran ve Ruby

Operatör Değerlendirme Önceliği

Language	*Associativity Rule*
Ruby	Left: *, /, +, -
	Right: **
C-based languages	Left: *, /, %, binary +, binary -
	Right: ++, --, unary -, unary +
Ada	Left: all except **
	Nonassociative: **

Operatör Değerlendirme Önceliği

Parentheses

Öncelik (precedence) ve birleşirlik (assosiativity) kurallarının uygulanması parantez kullanılarak değiştirilebilir.

(A + B) * C

Çarpma işlemi toplama işlemine göre önceliği olmasına rağmen parantezden dolayı önce toplama işlemi gerçekleştirilir

Ruby Operatorler

Ruby dilinde aritmetik ilişkisel ve atama operatörleri method olarak tanımlanmıştır.

a + b

a degeri + methodunu cagırıyor
b degeri de parametre olarak gönderiliyor.

Lisp Operatörleri

Lisp dilinde aritmetik ve mantıksal işlemler alt programlar (sub programs) ile gerçekleştirilmektedir.

a + b * c	(+ a (* b c))
JAVA	LISP
	+ ve * fonksiyon isimleridir

Side effect

Yan etki: bir fonksiyon global bir değişkeni değiştirse yan etki oluşturabilir.

```
int a = 5;

int fun1() {
    a = 17;
    return 3;
}

void main() {
    a = a + fun1();
}
```

Çözüm:

1- Dil tasarımcısı fonksiyonel yan etkiyi ortadan kaldırabilir (uygulanması zor, global değişken erişimi ortadan kaldırılıyor)

2- Operatörlerin terimlerinin belirli sıraya göre değerlendirilmesi sağlanır. (java soldan-saga degerlendirme)

Overloaded Operators

Çok amaçlı (overloaded) operatör birden fazla işlem için kullanılmaktadır.

JAVA

a + b (integer)

a + b (Strıng)

C++

& logical bitwise AND

& değişkenin adresini verir

Overloaded Operators

Çok amaçlı (overloaded) operatör birden fazla işlem için kullanılmaktadır.

JAVA	C++
a + b (integer)	& logical bitwise AND
a + b (Strıng)	& değişkenin adresini verir

Okunabilirliğe (readability) zarar verebilir .
Çünkü aynı görünen operatör farklı görevleri yerine getirmektedir

Type Conversions

Tip dönüşümleri

- Daralan (narrowing) dönüşüm
  - double dan float a geçiş
  - Güvenli değil
- Genişleten (widening) dönüşüm
  - İnteger dan floata geçiş
  - Genellikle güvenli

Type Conversions

Coercions:

işlem yapılacak olan iki terimden birinin tipinin diğerine derleme zamanında dönüştürülmesidir.

- JAVA (strıng1+ strıng2 işlemi statik tıp bağlama ile derleme zamanında terimlere uygun işlem tanımlanır)
- JAVA da karışık mod (mixed-mode) ifadeler kullanılır.

```
int a;
float b, c, d;
...
d = b * a;
```

JAVA da karışık mod ifadelerin kullanımına izin verilmesi derleme anında tıp hatası oluşmasını engelledi

a teriminin (operand) tipi derleyici tarafından coercion ile float a cevirildi

Explicit Type Conversions

Açık tıp dönüşümleri (casts)

Bir çok programlama dili açık tip dönüşümüne izin vermektedir

C (int) değişken

ADA Float (toplam)

Type Conversion Problems

Tip kontrolu (type checking) olan dillerde terim tip hataları oluşmaz.

- coercion işlemi
- Aritmetik işlemlerin sınırları
  - Sıfıra bölmeye
- Bilgisayarın aritmetik sınırları
  - İşlemin sonucu bellekte tutulması gereken yere (ayrılan yere) sıgmıyorsa
  - Owerflow (sonuc cok buyuk)
  - Underflow (sonuc cok küçük)

Relational Expressions

- iki terimi karşılaştıran operatörlerdir.
- İlişkisel operatörler birden fazla tip için kullanılabildiğinden çok amaçlı (overloaded) operatörler denir.

< , > , <= , >= , <> , != , ==

Relational Expressions

- İşlem önceliğinde (precedence) aritmetik operatörler ilişkisel operatörlerden önce gelir.
  - x+1 < 2 * y

C tabanlı diller	!=
Ada	/=
Lua	~=
Fortran 95+	.NE. veya <>
ML ve F#	<>
Python	! = veya not ==
Ruby	! =

EŞİT DEĞİLDİR'in Farklı dillerde ki gösterimi

Relational Expressions

c

A< B < C

Karşılaştırma
nasıl yapılır ?

C dilinde ilişkisel operatörler soldan sağa işleme alınır.
B ve C kesinlikle karşılaştırılmazlar!

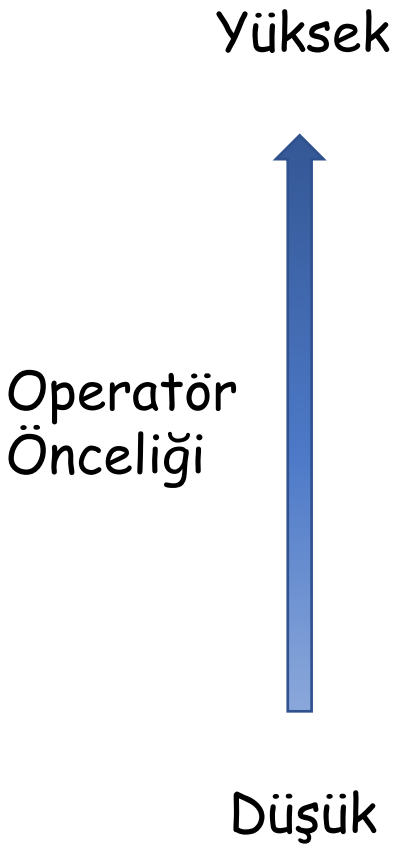
Relational Expressions

```
aaaydin-3:~ ahmetaydin$ irb
irb(main):001:0> "Arif"== "arif"
=> false
irb(main):002:0> "Arif" != "arif"
=> true
irb(main):003:0> 23 == 44
=> false
irb(main):004:0> 23 != 44
=> true
irb(main):005:0>
```

Boolean Expressions

- AND , OR ve NOT operatörlerinden oluşur
- İşlem sonucu True veya False
- Operatör önceliği programlama diline göre değişir
  - Ada AND ve OR aynı öncelikte
  - C tabanlı dillerde AND OR dan daha öncelikli

Relational and Boolean Expressions



postfix ++, --

unary +, -, prefix ++, --, !

*, /, %

binary +, -

<, >, <=, >=

=, !=

&&

||

Short-Circuıt Evaluation

Bütün terimleri kullanmadan sonuca erişime kısa devre (short-circuit) değerlendirme denir

(3 * x) * (y / 2 - 1)

Eğer x = 0 ise sonuc 0

(b <= 0) && (a > 10)

Eğer a = 3 ise sonuc FALSE

Ruby, Perl, ML, F#, ve Python da bulunan mantıksal operatorler
kisa devre değerlendirmeyi kullanmaktadır

Assignment Statements

- Atama işlemi değişkenlerin bağlandıkları değerin değişmesi işlemidir
- C tabanlı dillerde , FORTRAN , BASIC de atama sembolu =
- ADA , PASCAL ALGOL : =

Assignment Statements

Şartlı atamalar

```
( $şart ? $count1 : $count2) = 0;   PERL
```

```
if ( $şart )
{
    $count1 = 0;
}
 else
{
$count2 = 0;
}
```

Assignment Statements

Birleşik (compound) operatörler

toplam += deger; toplam = toplam+ değer;

Perl, JavaScript, Python, ve Ruby.

Unary Assignment Statements

Tekli (unary) atama operatörleri

++ Sayaç

Sayaç ++

Sayaç --

--Sayaç

Unary Assignment Statements

Tekli (unary) atama operatörleri

++ Sayaç

Sayaç ++

Sayaç --

--Sayaç

- count ++
- (count ++)