

ALGORİTMA VE PROGRAMLAMA I

Yrd. Doç. Dr. Deniz KILINÇ
deniz.kilinc@cbu.edu.tr

Genel Bakış...

2

- Koşul
- Karşılaştırma Operatörleri
- Mantıksal Operatörler ve Doğruluk Tablosu
- Operatör Önceliği
- if-else Bloğu
- İç içe if-else Blokları
- if-else Kullanım Hataları
- switch-case Bloğu
- switch-case Kullanım Hataları
- if-else ve switch-case Kullanım Karşılaştırması
- ? Karşılaştırma Operatörü

1. BÖLÜM

3

Karşılaştırma ve Koşul Deyimleri

Koşul

4

- Koşul ifadelerinin ve akış kontrollerinin programlama dillerinde ve uygulama altyapısında her zaman önemli bir yeri vardır.
- Yazdığımız uygulamanın **hangi koşulda nasıl davranacağını** belirterek bir akış kontrolü oluşturabilmek için koşul ifadelerinin iyi bilinmesi gereklidir.
- Tüm programlama dillerinde olduğu gibi en temel koşul ifadesi **«if-else»** bloklarıdır.

Koşul (devam...)

5

- Sonucu DOĞRU veya YANLIŞ olan herhangi bir ifadeye “koşul” denir.
 - DOĞRU → «1» ile gösterilir.
 - YANLIŞ → «0» ile gösterilir.
- **Örnek:**
if (kisi_boy > 170)
 printf("Uzun boylusunuz\n");
- **(kisi_boy > 170)** ifadesi bir koşuldur.
- Koşullar karşılaştırma operatörleri ve mantıksal operatörler kullanılarak oluşturulurlar.

Karşılaştırma Operatörleri

6

Operatör	Açıklama	Örnek	Anlamı
>	büyüktür	$x > y$	x, y'den büyük mü?
<	küçüktür	$x < y$	x, y'den küçük mü?
==	eşittir	$x == y$	x, y'ye eşit mi?
>=	büyük-eşittir	$x >= y$	x, y'den büyük veya eşit mi?
<=	küçük-eşittir	$x <= y$	x, y'den küçük veya eşit mi?
!=	eşit değil	$x != y$	x, y'den farklı mı?

Mantıksal Operatörler

7

Operatör	Açıklama	Örnek	Anlamı
&&	mantıksal VE	$x > 5 \ \&\& \ x < y$	x, 5'den büyük VE x, y'den küçük mü?
	mantıksal VEYA	$x > 5 \ \ x < y$	x, 5'den büyük VEYA x, y'den küçük mü?
!	mantıksal DEĞİL	$!(x > 5)$	x, 5'den büyük değilse (x, 5'den küçük VEYA 5'e eşitse) ($x \leq 5$)


Mantıksal Operatörlerin Doğruluk Tablosu

8

p	q	(p && q)	(p q)	!p
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Operatör Önceliği

9

Operatör	Öncelik
	En yüksek
* / %	
+ -	
< <= >= >	
== !=	
&&	
=	En düşük

Örnek-1: Mantıksal Operatörler

10

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int x, y, z;
7      int s1, s2, s3;
8
9      x = 10;
10     y = 5;
11     z = -1;
12
13     s1 = (x > y) && !(0);
14     s2 = (z != x) || !(0 == 1);
15
16     printf("s1:%d\n", s1);
17     printf("s2:%d\n", s2);
18     return 0;
19 }
20
```



**Ekran Çıktısı
Ne Olacak?**

s1:1

s2:1

Örnek-2: Mantıksal Operatörler

11

a=2,
b=4 için;

z = (a + 3 >= 5 && !(b < 5)) || a * b + b != 7

z = (a + 3 >= 5 && !1) || a * b + b != 7

z = (a + 3 >= 5 && 0) || a * b + b != 7

z = (5 >= 5 && 0) || a * b + b != 7

z = (1 && 0) || a * b + b != 7

z = 0 || a * b + b != 7

z = 0 || 8 + b != 7

z = 0 || 12 != 7

z = 0 || 1

z = 1

1

if...else Bloğu

12

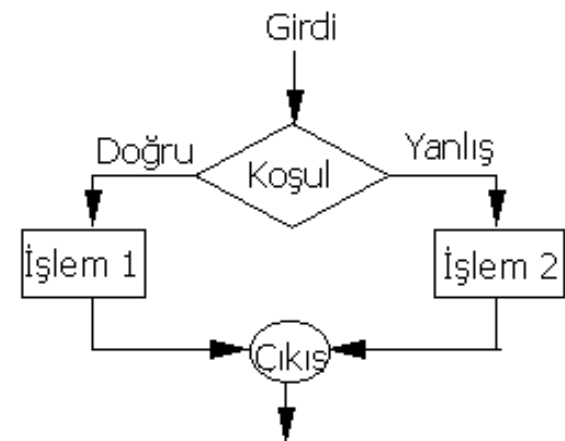
- **if** ve **else** tek bir karşılaştırma deyimi olup **else** kullanımı isteğe bağlıdır.
- Eğer bu koşul olumlu ise **if** den sonraki bölüm yürütülür ve **else** den sonraki bölüm atlanır.
- Koşul olumsuz ise **if** den sonraki küme atlanır ve eğer varsa, **else** den sonraki kümedeki işlemler gerçekleştirilir.

if (**koşul**)

{ Doğru ise İşlem1 }

else

{ Yanlış ise İşlem2 }



if...else Bloğu (devam...)

13

Örnek 1

```
if (sayi < 0) {  
    printf("Negatif sayı girdiniz\n");  
}  
else {  
    printf("Sıfır veya pozitif bir sayı  
girdiniz\n");  
}
```

Örnek 2

```
if (sayi < 0)  
    printf("Negatif sayı girdiniz\n");  
else  
    printf("Sıfır veya pozitif bir sayı  
girdiniz\n");
```

Not: Her iki örnek aynı işlemi gerçekleştirmektedir. Tek fark birinci örnekte { } sembollerinin kullanılmasıdır. Tek satır deyim içeren bloklarda, { } sembollerinin kullanılmasına gerek yoktur.

İç içe if...else Blokları (Nested If)

14

- **if-else** yapıları içerisine başka **if-else** yapıları yerleştirerek birden fazla koşul aynı anda test edilebilir.
- **Örnek:**

```
if (sayi < 0)
    printf("Negatif sayı girdiniz\n");
else
    if (sayi > 0)
        printf("Pozitif bir sayı girdiniz\n");
    else
        printf("Sıfır sayısını girdiniz\n");
```

İç içe if...else Blokları (devam...)

15

- Diğer bir kullanım şekli aşağıdaki gibidir:

```
if(koşul_1) {  
    deyimler; //(küme_1)  
}  
else if(koşul_2) {  
    deyimler; //(küme_2)  
}  
....  
....  
else if(koşul_n-1) {  
    deyimler; //(küme_n-1)  
}  
else {  
    deyimler; //(küme_n)  
}
```

Örnek: Girilen notun harf karşılığını bulma

16

Girilen not;

- 60 ile 65 arasındaysa DD
- 65 ile 70 arasındaysa DC
- 70 ile 75 arasındaysa CC
- 75 ile 80 arasındaysa CB
- 80 ile 85 arasındaysa BB
- 85 ile 90 arasındaysa BA
- 90 ile 100 arasındaysa AA

yazdıran bir program yazalım. Bunların dışındaysa «*Uygun değer girmediniz!*» şeklinde uyarı verip, programdan çıkacak.

Örnek: Girilen notun harf karşılığını bulma

17

```
int main()
{
    int ogrenci_not;
    printf("Notu giriniz:");
    scanf("%d",&ogrenci_not);

    if (ogrenci_not>60 && ogrenci_not<=65)
        printf("Harf kasiligi:DD");
    else if (ogrenci_not>65 && ogrenci_not<=70)
        printf("Harf kasiligi:DC");
    else if (ogrenci_not>70 && ogrenci_not<=75)
        printf("Harf kasiligi:CC");
    else if (ogrenci_not>75 && ogrenci_not<=80)
        printf("Harf kasiligi:CB");
    else if (ogrenci_not>80 && ogrenci_not<=85)
        printf("Harf kasiligi:BB");
    else if (ogrenci_not>85 && ogrenci_not<=90)
        printf("Harf kasiligi:BA");
    else if (ogrenci_not>90 && ogrenci_not<=100)
        printf("Harf kasiligi:AA");
    else
        printf("Uygun deger girmediniz!");
    return 0;
}
```

Koşullarda Hızlı Değerlendirme

18

- C programlamada, koşulların mantıksal değerlendirmelerinde, **beklenen değere ulaşıldığında** koşulun tamamı değerlendirilmeden **koşuldan çıkılır**.
- **VEYA** kullanımında koşulun ilk kısmı **DOĞRU** (**true**) ise koşulun tamamı **DOĞRU** olacağı için koşulun ikinci kısmı değerlendirilmez.
- **VE** kullanımında koşulun ilk kısmı **YANLIŞ** (**false**) ise koşulun tamamı **YANLIŞ** olacağı için koşulun ikinci kısmı değerlendirilmez.
- **Örnek:** **if** ((a>b) && (a<c)) eğer (a>b) **YANLIŞ** ise (a<c) değerlendirilmez.
- **Örnek:** **if** ((a>b) || (a<c)) eğer (a>b) **DOĞRU** ise (a<c) değerlendirilmez.

Koşullarda Hızlı Değerlendirme (devam...)

19

- Yazdığınız programın **performansı** açısından koşulların nasıl değerlendirildiğini iyi bilip, ona göre koşul sırası koymamız gerekmektedir.

Örnek: `if ((öğrenci_no==12345) && (öğrenci_isim="Veli"))`

- Bu örnekte en hızlı şekilde **YANLIŞ** (**false**) sonucu almak istiyorsak, en çok seçilebilir olan koşulu en sola yani başlangıca koymamız gerekmektedir.

Örnek: if-else değerlendirme sorusu

20

1. Program Parçası

```
if (x < 5)
if (y > 5)
printf("aaaa\n");
else
printf("bbbb\n");
printf("cccc\n");
```

Soru 1: x=4 ve y=6 iken
çıktı ne olur?

?

Soru 2: x=6 ve y=4 iken
çıktı ne olur?

?

2. Program Parçası

```
if (x < 5) {
if (y > 5)
printf("aaaa\n");
}
else {
printf("bbbb\n");
printf("cccc\n");
}
```

?

?

Örnek: Girilen sayının 10-50 aralığında olup olmadığının kontrolü

21

- Girilen sayı 10-50 arasındaysa ekrana yazdıran bir program yazalım.
- 10'dan küçük olma ve 50'den büyük olma kontrollerini de ekleyerek bu kontrollerin sonucunu da ekrana yazdıralım.

Örnek ekran çıktıları:

- 12 → "Girilen sayı 10-50 arasındadır"
- 9 → "Girilen sayı 10'dan küçüktür"
- 51 → "Girilen sayı 50'den büyüktür"

Not: Uygulama dersinde programı yazacağız.

if-else Kullanım Hataları

22

Hata-1

```
if sayi==10 //Söz dizimi hatası  
    printf ("Girilen sayi 10\n");
```

- Koşul ifadesinde () sembolleri kullanılmadığı için “**syntax error**” hatası alınır (Söz dizimi hatası).

if-else Kullanım Hataları (devam...)

23

Hata-2

```
if (sayi==10) ; //Boş ifade  
printf ("Girilen sayi 10\n");
```

- Koşul ifadesinden sonra ; sembolü kullanılmıştır. C derleyicisi için boş bir ifadedir. Sonraki satır bu satırdan bağımsızdır.

if-else Kullanım Hataları (devam...)

24

Hata-3

```
if (10<=sayi<=50) //Mantıksal Hata
```

- Sonucu **her zaman doğru** olan mantıksal bir hatadır.
- Koşul ifadesi soldan sağa doğru değerlendirilir. **sayi** değişkenin değeri 10 ile karşılaştırılır.
 - 10'dan küçükse **false** yani **0** sonucu döner,
 - 10'dan büyükse **true** yani **1** sonucu döner,
 - Her iki durum da (0 veya 1 sonucu) dönen değer 50'den küçük olacağı için bu ifade her zaman doğru sonuç verir.
- Bunun yerine `((sayi>=10) && (sayi<=50))` koşul ifadesi kullanılmalıdır.

if-else Kullanım Hataları (devam...)

25

Hata 4

```
if (sayi=10) //Mantıksal Hata
```

- Sonucu **her zaman doğru** olan mantıksal bir hatadır.
 - Koşul ifadesinde karşılaştırma operatörü `==` yerine, atama operatörü `=` kullanılmıştır.
 - Atama sonucunda 10 değeri oluşur.
 - 10 değeri 0'dan büyük olduğu için sonuç “true” dönecektir ve bu da yanlış bir sonuç üretecektir.
- Atama operatörü yerine, `==` karşılaştırma operatörü kullanılmalıdır.

switch...case Bloğu

26

- Bir **değişkenin** aldığı **değerlere** bağlı olarak, program akışındaki belirli işlemlerin yaptırılmasını sağlar.
- Aslında bu tür işlemleri **if** deyimi yardımıyla yerine getirebiliriz.
- Ancak bazı uygulamalarda **if** deyiminin kullanılması programın **karmaşıklığını artırabilir** ve **okunabilirliğini azaltabilir**.
- Eğer bir değişkenin değeri belirli **sabitlerle** karşılaştırılacak ve bunun sonucunda farklı işlemler yapılacak ise **if** deyimi yerine **switch** deyimi kullanmamız daha sağlıklı olacaktır.
- Sadece **char** ve **int** veri türleri için kullanılabilir.

switch...case Bloğu (devam...)

27

```
switch (değişken) {
```

```
    case sabit1:  
        deyim1  
        break;
```

değişken değeri ***sabit*₁** değerine eşitse **deyim₁** çalıştırılır ve **break** kullanılarak bloktan çıkış yapılır.

```
    case sabit2:  
        deyim2  
        break;
```

```
    ...
```

```
    case sabitn:  
        deyimn  
        break;
```

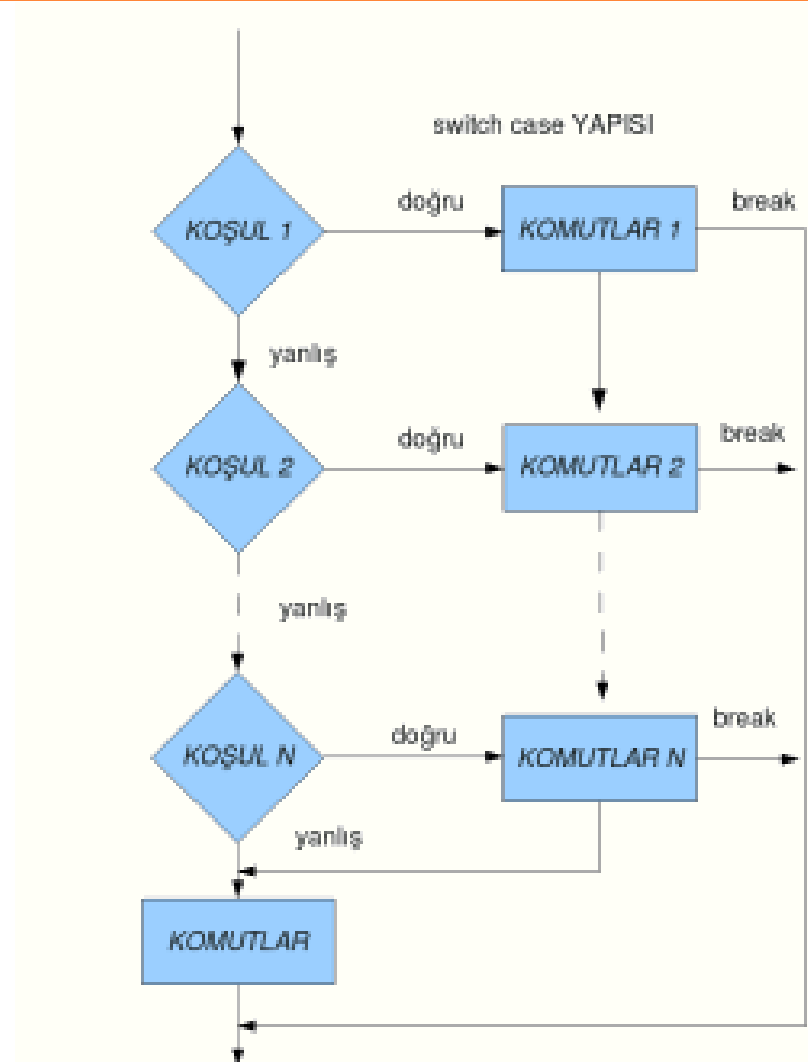
```
default:  
    deyimd;
```

```
}
```

değişken değeri herhangi bir sabite eşit değilse **default** sabitinin deyimini ***deyim_d*** çalışır.

switch...case Bloğu Akış Şeması

28



Örnek: Girilen il plaka kodlarına göre şehir isimlerinin yazdırılması

29

- Girilen il plaka koduna göre ekrana ilgili şehir ismi yazdırılacaktır.
- Program tarafından desteklenecek iller:
 - ANKARA
 - İSTANBUL
 - İZMİR
 - MANİSA
- Bunların dışında bir plaka kodu girilirse «TANIMSIZ PLAKA KODU» şeklinde hata mesajı verilecektir.

Örnek: Girilen il plaka kodlarına göre şehir isimlerinin yazdırılması (switch-case bloğu)

30

```
int main()
{
    int plaka_kodu;
    printf("İlin plaka kodunu giriniz: ");
    scanf("%d", &plaka_kodu);

    switch (plaka_kodu) {
        case 6:
            printf("ANKARA\n");
            break;
        case 34:
            printf("İSTANBUL\n");
            break;
        case 35:
            printf("İZMİR\n");
            break;
        case 45:
            printf("MANİSA\n");
            break;
        default:
            printf("TANIMSIZ PLAKA KODU\n");
    }
    return 0;
}
```

Örnek: Girilen il plaka kodlarına göre şehir isimlerinin yazdırılması (if-else bloğu)

31

```
int main()
{
    int plaka_kodu;
    printf("İlin plaka kodunu giriniz: ");
    scanf("%d", &plaka_kodu);

    if (plaka_kodu==6)
        printf("ANKARA\n");
    else if (plaka_kodu==34)
        printf("İSTANBUL\n");
    else if (plaka_kodu==35)
        printf("İZMİR\n");
    else if (plaka_kodu==45)
        printf("MANİSA\n");
    else
        printf("TANIMSIZ PLAKA KODU\n");

    return 0;
}
```

Örnek: Klavyeden harf girilmesi

32

- Klavyeden 'a' veya 'A' harfi girildiğinde "A harfi girdiniz" mesajını ekrana yazdıracak bir program yazılacaktır.
- Başka bir harf girilirse "A harfi girmediniz" şeklinde hata mesajı verilecektir.

Örnek: Klavyeden harf girilmesi

33

```
int main()
{
    char harf;
    printf("Bir harf giriniz: ");
    scanf("%c", &harf);
    switch (harf){
        case 'a':
        case 'A':
            printf("A harfi girdiniz\n");
            break;
        default:
            printf("A harfi girmediniz\n");
    }
    return 0;
}
```

switch-case Kullanım Hataları

34

- **break** deyiminin kullanılmasının unutulması.
- switch-case bloğundaki **{ }** sembollerinin kullanımlarının unutulması.
- **default** deyiminin ve içeriğinin kullanımının unutulması.
- Kontrol edilen değişken ile verilen sabitlerin farklı tipte olması.
- **int** ve **char** veri türleri dışında değişken tanımlamaya çalışmak (double, string)

if-else ve switch-case

Kullanımlarının Karşılaştırması

35

1. İşlevsellik:

- C programlama dilinde
 - switch-case deyimi sadece **char** ve **int** veri türlerini destekler. Bu durum if-else deyimine göre dezavantajdır.
 - switch-case deyimlerinde aralık kontrolü ($x > 1000$) yapılamaması yine bir dezavantajdır.



if-else ve switch-case

Kullanımlarının Karşılaştırması (devam...)

36

2. Performans:

- Her iki karşılaştırma deyiminin kullanımı farklı derleyicilerde ve farklı programlama dillerinde farklı performans sonuçlarına üretebilir.
- Genelde **switch-case** kullanımını **daha performanslı** çalışır.



if-else ve switch-case

Kullanımlarının Karşılaştırması (devam...)

37

3. Okunabilirlik:

- **switch-case** deyimlerinin kod okunabilirliği her zaman daha yüksektir.

```
/**  
 * Code Readability  
 */  
if (readable()) {  
    be_happy();  
} else {  
    refactor();  
}
```

? Karşılaştırma Operatörü

38

- Bu operatör, **if-else** karşılaştırma deyiminin yaptığı işi sınırlı olarak yapan bir operatördür. Genel yazım biçimi:

(koşul) ? deyim1 : deyim2;

- İlk önce koşul sınanır.
- Eğer koşul **olumluysa** deyim1, **olumsuzsa** deyim2 değerlendirilir.
- deyim1 ve deyim2 de atama işlemi yapılamaz.
- Ancak **koşul deyiminde atama işlemi yapılabilir**.
- deyim1 ve deyim2 yerine fonksiyon da kullanılabilir.

? Karşılaştırma Operatörü (devam...)

39

- **Örnek:**

$x = (a > b) ? a : b;$

- Yukarıdaki ifadede **koşul** a'nın b'den büyük olmasıdır.
- Eğer **olumluysa** x adlı değişkene **a değeri** atanır,
- **Olumsuzsa** x adlı değişkene **b değeri** atanır.
- Örneğin if-else bloğu ile gösterimi:

if (a > b)

 x = a;

else

 x = b;

? Karşılaştırma Operatörü (devam...)

40

- Örnek:

```
int sayi = 15;
```

```
x = ( sayi < 10 ) ? printf("yazi1") : printf("yazi2");
```


KAYNAKLAR

41

- N. Ercil Çağıltay ve ark., C DERSİ PROGRAMLAMAYA GİRİŞ, Ada Matbaacılık, ANKARA; 2009.
- Milli Eğitim Bakanlığı "Programlamaya Giriş ve Algoritmalar Ders Notları", 2007
- <http://tr.wikipedia.org/wiki/Code::Blocks>
- <http://www.codeblocks.org>
- <http://www.AlgoritmaveProgramlama.com>
- <http://www1.gantep.edu.tr/~bingul/c>



Algoritma ve Programlama

İYİ ÇALIŞMALAR...

Yrd. Doç. Dr. Deniz KILINÇ
deniz.kilinc@cbu.edu.tr