

Programlama Dillerinin Seviyelerine Göre Sınıflandırılması

- Önce seviye kavramının ne anlama geldiğini açıklamak gerekmektedir.
- Seviye, bir programlama dilinin insan algılamasına olan yakınlığının bir ölçüsüdür.
- **Yüksek seviyeli diller** insan algılayışına daha yakın, alçak seviyeli diller de bilgisayarın doğal çalışmasına daha yakın olan dillerdir.
- Dillerdeki seviye yükseldikçe programcının işi de kolaylaşır. Öyle ki, çok yüksek seviyeli programlama dillerinde artık bir işin nasıl yapılacağına ilişkin değil, ne yapılacağına ilişkin komutlar bulunur.
- Seviyenin yükselmesi programcıya kolaylık sağlamakla birlikte genel olarak verimliliği ve esnekliği de azaltır.

1

- Çok Yüksek Seviyeli Programlama Dilleri ya da Görsel Diller (FOXPRO, PARADOX, ACCESS..., VISUAL BASIC, IV.KUŞAK DILLER)

- Yüksek Seviyeli Programlama Dilleri (PASCAL, COBOL, FORTRAN, BASIC,...)

- Orta Seviyeli Programlama Dilleri (C)

- Alçak Seviyeli Programlama Dilleri (Sembolik Makine Dilleri)

2

C NASIL BİR DİL?..

- **Orta seviyeli bir dildir.** Yazılan C kodu ile makina kodu arasında bağlantı kolaylıkla kurulabilir.
- **Sistem programlama dilidir.** Bugün işletim sistemleri, derleyiciler, editörler gibi sistem programlarının hemen hepsi yoğun olarak C kodu içermektedir. Ancak sistem programlarının dışında da birçok uygulamada C verimli olarak kullanılabilir.
- **Algoritmik bir dildir.** Yalnızca dilin sintaks ve semantik yapısını bilmek yetmez. Problemleri çözebilecek bir algoritma bilgisine de ihtiyaç duyulur.
- **Diğer diller arasında taşınabilirliği en fazla olanlardan biridir.**
- **İfade gücü yüksek ve okunabilirlik özelliği kuvvetli bir dildir.**
- **Çok esnek.** Bu yüzden programcının hata yapmayacak bir bilgiye ve deneyime sahip olması gerekir.
- **Atomik bir dildir.** Cde altprogramlama tekniği ileri düzeyde kullanılmakta-dır.
- **Güçlü bir dildir.** Tasarım özellikleri çok iyidir. Cye ilişkin yapıların ve operatörlerin bir kısmı daha sonra diğer diller tarafından da benimsenmiştir.
- **Verimli bir dildir.** C programları seviyesi dolayısıyla daha hızlı çalışır.
- **Doğal bir dildir.** C bilgisayar sisteminin çalışma biçimiyle uyum içindedir.
- C++ ile nesne yönelimlilik özelliğine de sahip olmuştur.

3

PROGRAMIN ÇALIŞMASI

source ----> compiler -----> object -----> link
kaynak derleyeci amaç bağlama

- kaynak kod : C dili ile yazılmış olan program.
- derleyeci : Kaynak kodu makina koduna çevirir
- amaç kodu : Kaynak kodun makina dilindeki karşılığı
- bağlama : Birden fazla amaç kodu dosyasının tek dosyada birleştirilmesi

4

İLK PROGRAM

```
#include<stdio.h>
#include<conio.h>
main()
{
    printf("Bu bir satirlik yazidir.");
}
```

5

VERİ TİPLERİ

int Tip

- Integer = Tamsayı
- Tamsayıları içerir. Bellekte 2 Byte tutar.
- 5 , -19 , 25000 gibi
- Minimum : -2^{31} = -32768
- Maksimum : $2^{31}-1$ = 32767

Long tip

Uzun tamsayı. Bellekte 4Byte tutar.
Minimum : **-2,147,483,648**
Maksimum : **2,147,483,647**

6

VERİ TİPLERİ

Gerçel Tipler (Float, Double)

- Gerçel sayıları içerirler.

float : Bellekte 4 Byte yer tutar.

- $3.4E-38$ ile $3.4E+38$ aralığında değer alır.
- Hassasiyet 7-8 basamaktır.

double : Bellekte 8 Byte ter tutar.

- $1.7E-308$ ile $1.7E+308$ aralığında değer alır.
- Hassasiyet 15-16 basamaktır.

- **NOT: Bilimsel gösterim biçimi** $2.5*10^3 = 2.5E3$

7

VERİ TİPLERİ

Char Tip

- Char : Karakter :
- Alfanumerik karakterleri içerir.
- '5' , '*' , 'K' gibi

8

VERİ TİPLERİ



Tip	Aralık	Uzunluk (Byte)
unsigned char	0 .. 255	
char	-128 .. 127	
int	-32,768 .. 32,767	
long	-2,147,483,648 .. 2,147,483,647	
float	$3.4 * (10^{**}-38) .. 3.4 * (10^{**}+38)$	
double	$1.7 * (10^{**}-308) .. 1.7 * (10^{**}+308)$	

9

Belirleyiciler (Identifler), Değişkenler



- Değişkenlere isim verme
- **Kısıtlamalar**
- İlk karakteri rakam olamaz
- Sadece harfler, rakamlar ve _ (underscore) karakterinin birleşiminden oluşabilir.
- C diline ait saklı (reserved) sözcükler kullanılamaz.
- En fazla 31 karakter olabilir.
- **Küçük - büyük harf ayırımı vardır.**
- **Türkçe karakter kullanılmayacak**

10

Matematiksel ifadeler



- + toplama, - çıkarma, / bölme, * çarpma için kullanılır.
- % tamsayı bölme sonucundaki kalanı verir (modulus)
- $7.0 / 2 \text{ ---> } 3.5$ (nokta sayının gerçel yorumlanmasını sağlar)
- $7 / 2 \text{ ---> } 3$ (sayılar int kabul edilip sonuçta int tipine dönüştürülür)
- $7 \% 2 \text{ ---> } 1$
- $2 \% 7 \text{ ---> } 2$
- Matematiksel ifadeler hesaplanırken izlenen adımlar:
 1. Önce parantez içindeki ifadeler hesaplanır. İççe parantezler var ise hesaplamaya en içteki parantezden başlanır.
 2. İlk önce *, / ve % işlemleri daha sonra + ve - işlemleri yapılır.
 - Öncelik sırası aynı olan işlemlerde hesaplama soldan sağa doğru yapılır

11

Bir C Programının Genel Yapısı



- başlık dosyaları: Kullanılan deyimler ile ilgili main()
{
sabitler ; sabitlerin tanımlama bölümü
değişkenler; değişkenleri tanımlama bölümü
deyimler; Program gövdesi
}

12

printf

- Program içinde istenen değerleri ekranan göndermek için kullanılır.
 - **printf(kontrol, <parametreler>)**
- <kontrol> ile belirtilen bilgiye göre parametreleri yazar. Kontrol metni iki kısımdan oluşur.
- sıradan karakterler: doğrudan yazılırlar.
 - dönüşüm/biçim belirleyici karakterler: Değerlerin nasıl yazılacağı bildirilir. Değişkenlerin türüne göre farklılık gösterir.
- int k ;
 - k=10;
 - printf(" sonuç = %d;", k);
 - -----kontrol----- parametreler

13

DÖNÜŞ KARAKTERLERİ

- %d : decimal (integer)
- % f : float/double sayıyı mmm.nnnnn biçiminde yaz
- % c : char (tek karakter)
- % e : float/double sayıyı bilimsel gösterimde yaz
- % ld : long integer

ESC dizileri :

- \n : satır başı,
- \t : tab,

14

ÖRNEKLER

```
#include<stdio.h>
#include<conio.h>
main()
{ int i;
  i = 5 ;
  printf("%d",i) ;           5
  printf("Sayı=%d",i) ;     Sayı=5

  printf("i=\n") ;         i= yazar ve alt satıra geçer
  printf("%d",i) ;         5

  printf("i=%d\n",i) ;     i=5
  printf("%d %d",i, 5*i);  5 25
}
```

15

ÖRNEK

```
#include<stdio.h>
#include<conio.h>
main()
{
  int i,k;
  i=100;
  k=200;
  printf("\n birinci sayının değeri=%d",i);
  printf("\n ikinci sayının değeri=%d",k);
  getch();
}
```

16

ÖRNEK

```
#include<stdio.h>
#include<conio.h>
main()
{
float vize,final,ortalama;
vize=98;
final=85;
ortalama=vize*0.4+final*0.6;
printf("\n Ogrencinin hesaplanan notu=%f",ortalama);
getch();
}
```



CONST (sabit) : Değeri programın çalışması boyunca değiştirilemez. Kullanım biçimi

const tip **Belirleyici=değeri**;

```
const float PI=3.14;
```

Tip belirtilmez ise tamsayı kabul edilir.

DEĞİŞKENLER

Değeri programın çalışması boyunca değiştirilebilir.

Kullanım biçimi:

Tip Belirleyici, [Belirleyici] ;

```
int i;
```

```
float alan;
```

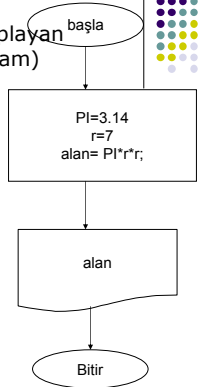


18

Örnek :

- Yarıçapı belli dairenin alanını hesaplayan programı yazınız. (ilk yazılan program)

```
#include <stdio.h>
main()
{
const float PI=3.14;
float r, alan;
r = 7;
alan = PI*r*r;
printf("Dairenin Alanı = %f", alan);
}
```



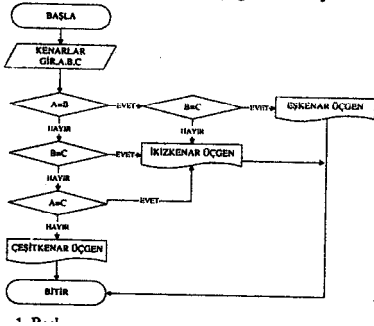
19

- main()
{
 int index;
 index = 13;
 index = 27;
 printf("Indexin degeri simdi %d\n",index);
 index = 10;
 printf("Indexin degeri simdi %d\n",index);
}



20

Klavyeden üç kenar uzunluğu girilen üçgenin çeşidini bulan programın akış şeması



21

BILGI SATIRLARI NASIL EKLENİR

/* Bu satiri, derleyici kullanmaz */

/* arası derleyici için anlamsızdır*/

22

• Biçimli yazdırma

• % karakteri ile dönüşüm karakteri arasına aşağıdaki karakterler de kullanılabilir.

• %m.n f

m : yazılacak değer için ayrılan toplam basamak sayısı

n : ondalık değer için ayrılan basamak sayısı

- `x = 128.5 ;`
- `printf("%7.2f",x) ;` **# 128.50**
- `x = 85.47 ;`
- `printf("%6.3f",x) ;` **85.470**
- `printf("%6.1f",x) ;` **## 85.5**

23

scanf İşlevi

• Klavyeden veri okumak için kullanılır. Yapı olarak printf işlevi ile hemen hemen aynıdır. Kullanım biçimi:

- `scanf(format, &değişken)`

- Girilen değerler, belirtilen biçimlere göre değişkenlere aktarılır.
- Değişkenler işaretçi tipinde olmalıdır. Yani parametre olarak değişkenin adresi gönderilmelidir.
- `int i;`
- `scanf ("%d", &i);`
- `float a,b,c;`
- `scanf("%f %f %f ", &a, &b, &c);`

24



```
#include<stdio.h>
#include<conio.h>
```

```
main()
{
    float vize, final, ort;
    clrscr();
    printf("\n Vize notunu giriniz "); scanf("%f", &vize);
    printf("\n Final notunu giriniz "); scanf("%f", &final);
    ort = vize * 0.4 + final * 0.6;
    printf("\n Öğrencinin Ortalaması = %f", ort);
    getch();
}
```

25

Mantıksal ifadeler

• **İlişkisel işleçler(operatör)** : iki değer arasındaki ilişkiyi test etmek için kullanılır.

- **işleç anlamı**
- > büyük
- >= büyük - eşit
- == eşit
- < küçük
- <= küçük - eşit
- != eşit değil
-



26

Mantıksal işleçler

İki mantıksal ifade arasındaki ilişki üzerindeki ilişkide kullanılır.

İşleç Anlamı

- ! DEĞİL (NOT)
- && VE (AND)
- || VEYA (OR)

- (X>0) && (X>Y)
- (X>0) || (Y>0)

27

İfadelerde işleçlerin yürütülme sırası

- işleç _____ Önceliği
- () en yüksek (ilk yürütülür)
- !
- *, /, %
- +, -
- <, <=, >=, >
- ==, !=
- &&, ||
- = en düşük (son yürütülür)



28

```
#include<stdio.h>
#include<conio.h>
```

```
main()
{
    float fiyat,kdv,kdvli;
```

```
    printf("\n kdv oranını giriniz "); scanf("%f", &kdv);
    printf("\n Ürün fiyatını giriniz "); scanf("%f", &fiyat);
    kdvli= fiyat+(kdv*fiyat)/100;
    printf("\n Ürünün Kdv li fiyatı = %f", kdvli);
    getch();
}
```

29

```
#include<stdio.h>
#include<conio.h>
main()
```

```
{
    float fahrenheit,derece;
    printf("\n sıcaklığı fahrenheit cinsinden giriniz");
    scanf("%f", &fahrenheit);
    derece=( fahrenheit-32)*5/9;
    printf("\n Sıcaklık= %f derecedir", derece);
    getch();
}
```

30

```
/* mod operatörü ile girilen saniyeyi dakikaya çeviriyor */
```

```
#include<stdio.h>
#include<conio.h>
```

```
main()
{
    int saniye,dakika,kalan;
    printf(" zamani saniye olarak giriniz \n");
    scanf("%d",&saniye);
    dakika=saniye/60; /* tamsayı bolmesi yapıyor*/
    kalan=saniye % 60; /* mod alıyor */
    printf("%d saniye, %d dakika ve %d saniye yapıyor",
    saniye,dakika,kalan);
    getch();
}
```

31

• Klavyeden girilen paralel iki direncin, toplam direnc değerini hesaplayan programı yapınız

```
#include<stdio.h>
#include<conio.h>
main()
{int r1,r2;
    float toplam;
    printf("\n Birinci direnc degerini giriniz ");
    scanf("%d", &r1);
    printf("\n İkinci direnc degerini giriniz ");
    scanf("%d", &r2);
    toplam=(r1*r2)/(r1+r2);
    printf("\n Toplam direnç deđeri=%f",toplam);
    getch();}
```

32

if-else Deyimi

Birkaç seçenektan birini seçmek veya bir deyimin bir koşula bağlı olarak işlemek için kullanılır.

if (mantıksal ifade)

```
{ blok_doğru;}
```

else

```
{ blok_yanlış;}
```

Mantıksal ifade doğru ise blok_doğru, yanlış ise else sözcüğünden sonraki blok_yanlış yürütülür. else kısmı seçimlidir, gerekmiyorsa kullanılmayabilir.

33

Girilen sayının tek/çift olduğunu yazan program

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main()
```

```
{
```

```
int i, kalan;
```

```
scanf("%d", &i);
```

```
kalan=i%2;
```

```
if ( kalan == 1)
```

```
printf("Tek Sayı");
```

```
else
```

```
printf("Çift Sayı ");
```

```
getch();
```

34

```
#include <stdio.h>
#include <conio.h>
```

```
main()
```

```
{
```

```
float vize, final, ort;
```

```
printf("\n Vize notunu giriniz "); scanf("%f", &vize);
```

```
printf("\n Final notunu giriniz "); scanf("%f", &final);
```

```
ort = vize * 0.4 + final * 0.6;
```

```
if (ort >= 50 && final >= 50)
```

```
{printf("\n Gectiniz");
```

```
printf("\n Öğrencinin Ortalaması = %f", ort);}
```

```
else
```

```
printf("\n Bütünleme");
```

```
getch();
```

```
}
```

35

```
#include <stdio.h>
#include <math.h> /* sqrt işlevi için */
#include <conio.h>
```

```
main()
```

```
{
```

```
float a, b, c, delta, x1, x2;
```

```
printf("a değerini giriniz \n"); scanf("%f", &a);
```

```
printf("b değerini giriniz \n"); scanf("%f", &b);
```

```
printf("c değerini giriniz \n"); scanf("%f", &c);
```

```
delta = b * b - 4 * a * c;
```

```
if (delta < 0)
```

```
printf("Gerçel kök yoktur.\n");
```

```
else
```

```
{
```

```
x1 = (-b + sqrt(delta)) / (2 * a);
```

```
x2 = (-b - sqrt(delta)) / (2 * a);
```

```
printf("Birinci kök = %f\n", x1);
```

```
printf("ikinci kök = %f\n", x2);
```

```
}
```

```
getch();
```

```
}
```

36

If /Else if deyimi

```
if (mantıksal_ifade)
{ ifade bloğu;}
else if (mantıksal_ifade)
{ ifade bloğu;}
else if (mantıksal_ifade)
{ ifade bloğu;}
else
{ ifade bloğu;}
```

37

[Verqi iadesini hesaplayan programı yazınız. Fatura toplamı](#)
[Fatura toplamı 0-60 :ise %10 kdv,](#)
[Fatura toplamı 60-200 : %8 kdv ,](#)
[Fatura toplamı >200 : %5 kdv](#)

```
main()
{
float fat_top, vergi_iade;
printf("Fatura toplamlarını giriniz ");
scanf ("%f", &fat_top);
if (fat_top < 60)
    vergi_iade = fat_top * 0.10;
else if (fat_top < 200)
    vergi_iade = fat_top* 0.08;
else
    vergi_iade = fat_top * 0.05;
printf("Ödenecek vergi iadesi = %f\n", vergi_iade);
getch();
}
```

38

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
main()
{ int sayi,rasgele;
float ortalama;
randomize();/* Her seferinde farkli bir deger tutmasi icin*/
rasgele=random(100);
printf("\n Tahmin icin sayi giriniz..");scanf("%d",&sayi);
if (sayi>rasgele)
    printf("\n Girilen sayi büyük...");

else if (sayi<rasgele)
    printf("\n Girilen sayi küçük...");

else
    printf("\n Tebrikler Bildiniz...");

    printf("\n Bilgisayarın tuttuğu sayı =%d ...",rasgele);
getch();}
```

39

switch Deyimi

```
switch(değişken) {
    case seçenek1 : Deyim; break;
    case seçenek2 : Deyim; break;
    :
    :
    default : Deyim;
}
```

Değişkenin aldığı değere eşit seçeneğin olup olmadığını bakar. Var ise o noktadan sonraki deyimler yürütülür. **switch** deyiminin sonuna gelindiğinde veya **break** deyimi ile karşılaşıldığında yürütme işlemi durur ve programın akışı switch deyimini izleyen deyim ile devam eder.

40

switch Deyimi

- Switch'deki Değişken integer veya char tipinde olmalıdır
- Değişken ile switch'deki seçenekler aynı tipte olmalıdır.
- default kısmı seçimlidir. Seçeneklerin hiçbiri uygun değil ise yürütülür.

41

```
#include <stdio.h>
#include <conio.h>
```

```
main()
{
    char islem;
    int s1, s2, s3;
    printf(" \n işlem için + - * veya/ giriniz "); scanf("%c", &islem);
    printf("\n İki sayı giriniz" ); scanf("%d %d",&s1,&s2);
    switch (islem) {
        case '+': s3 = s1 + s2; break;
        case '-': s3 = s1 - s2; break;
        case '*': s3 = s1 * s2; break;
        case '/': s3 = s1 / s2; break;
        default : printf ("Hatalı işlem");
    }
    printf("\nSonuç = %d",s3);
    getch();
}
```

42

```
#include <stdio.h>
#include <conio.h>
main() {
    printf("\n Ay için 1-12 arası değer giriniz");
    scanf("%d", &ay);
    switch (ay) {
        case 1: printf("\n Ocak");break;
        case 2: printf("\n Şubat");break;
        case 3: printf("\n Mart"); break;
        case 4:printf("\n Nisan"); break;
        case 5:printf("\n Mayıs "); break;
        case 6: printf("\n Haziran"); break;
        case 7: printf("\n Temmuz "); break;
        case 8: printf("\n Ağustos "); break;
        case 9: printf("\n Eylül "); break;
        case 10: printf("\n Ekim "); break;
        case 11: printf("\n Kasım "); break;
        case 12: printf("\n Aralık "); break;
        default: printf("\n 1-12 arası değer giriniz ");
    }
    getch();}
```

43

```
#include <stdio.h>
#include <conio.h>
main() {
    scanf("%d", &ay);
    switch (ay) {
        case 3:
        case 4:
        case 5: printf("ilkbahar"); break;
        case 6:
        case 7:
        case 8: printf("yaz"); break;
        case 9:
        case 10:
        case 11: printf("sonbahar"); break;
        case 12:
        case 1:
        case 2: printf("kış"); break;
    }
    getch();}
```

44