

5. BÖLÜM (10ncu ve 11nci hafta)

5.1 Fonksiyon Alt Programları

Bilgisayar programları, komutları, satırları kolayca takip edilebilir, sade ve anlaşılır olabilmeleri için eklemeli (modüler) bir yapıya sahip olarak tasarlanırlar. Yani bilgisayara yaptırılacak işler mümkün olduğunca küçük program parçalarına veya birimlerine bölünerek hazırlanır. Bu modüler programlar pratikte büyük yararlar sağlamaktadır. Tekrarlanan program satırlarının ana programdan ayrı bir bölüm şeklinde hazırlanıp ana program veya diğer ek programlar içinde kullanılması günümüz programcılığında önemlidir. Örneğin; verilerin sıralanması işlemi bir modül olarak hazırlanabilir. Sıralacak veriler alt programa yollanır ve burada sıralanıp sıralamanın yapılması istenilen satırlara geri dönülür. Dosya açma/kapatma, dosyanın var olup olmadığının kontrol edilmesi için bir alt program veya bir modül oluşturulabilir. Yazıcıya veri yollanması, yazıcının açık olup olmaması, yazıcılar arasından uygun yazıcıyı seçme işlemleri modüler programlarla yapılabilir. Modüllerin ana programdan ayrı olarak yazılması ve test edilmesi, programı bir bütün olarak ele alıp üzerinde uğraşmaktan çok daha kolaydır. Özetle, modüler programlar/alt programlar iş paylaşımını ve yazılımın bütününe kısa sürede tamamlanmasını veya bitirilmesini sağlar. Bir dökümandaki karakterlerin sayılması, karakterlerin küçük harflere veya büyük harflere dönüştürülmesi, rakamların yazıya dönüştürülmesi gibi sürekli tekrarlanan işlemlerin program parçaları şeklinde yazılması uygundur. Bu program parçalarına alt programlar denir. C/C++ da tek tip alt program vardır: fonksiyon. Fonksiyonun genel yazımı aşağıda verilmektedir:

```
Dönüş_tipi fonksiyon_adi(parametre_tanım_listesi)
{
    işlemler;
    Dönecek_olan_değer;
}
```

Buradaki

dönüş_tipi fonksiyon_adi ile dönecek olan değer tipidir (tamsayı, kesirli sayı, karakter vs.)

fonksiyon_adi fonksiyonun işlevini kısaca tanımlayan bir isimdir,

parametre_tanım_listesi virgülle birbirlerinden ayrılmış olan veriler,

işlemler fonksiyon içinde yapılacak işlemleri belirtir,

Dönecek_olan_değer fonksiyonun çağrıldığı yere (return deyimi ile) geri yollanacak değeri tanımlar.

Fonksiyonlar bir çeşit alt programlardır (subroutines). Birden fazla değer alt programa yollanabilir ama dönen değer tek olur. Bir ana programı nasıl düzenliyorsan bir alt programda aynı şekilde düzenlenir. Alt program içinde kullanılacak değişkenlerin tanımlanması, komutların yazılışı vs. Ana program veya diğer alt programlar tarafından kullanılacak olan alt programlar kullanılmadan önce

(yerel veya genel amaçlar için) tanımlanmalıdırlar. C veya C++ dilinin kendisi de bir (fonksiyon) alt programdır (main() {...}).

Bir fonksiyon alt programı hazırlanırken yapılacak işlemler sırası için bir algoritma aşağıdaki çizelgedeki gibi verilebilir:

Çizelge 5-1 Fonksiyon alt programlarının hazırlanması için algoritma.

1. Giriş ve çıktı bilgilerinin tanımlanması,
2. Nesnelerin tanımlanması,
3. Yapılacak işlemlerin belirlenmesi,
4. Algoritma hazırlanması,
5. Kodlamanın yapılması,
6. Test yapılması : çalıştırma ve hata ayıklama
7. Kullanıma hazır hale getirilmesi.

Aşağıdaki örnekte fonksiyon alt programı kullanılarak hazırlanmış bir program verilmektedir.

Örnek 5-1 Aşağıda, Fahrenheit sıcaklık skalasından Celcius sıcaklık skalasına geçiş yapmak için fonksiyon alt programı kullanılarak bir program hazırlanmıştır.

Program 5-1 Sıcaklık değerlerinin Fahrenheit tan Celcius a fonksiyon ile çevrilmesi.

```
/* sıcaklığın Fahrenheit biriminden Celcius birimine çevrilmesi
   Girdi : g_f fonksiyonu
   çıktı : g_c fonksiyonu
*/
float f_c (float g_f); /* fonksiyon protipi tanımlanır*/
main() {
    float g_f, g_c; // değişkenler tanımlanır
    printf("Program sıcaklıkları Fahrenheit biriminden ");
    printf("derece birimine çevirir. \n\n");
    printf("Fahrenheit değerini giriniz : ");
    scanf("%f", &g_f);
    printf("%f Fahrenheit = %f Celcius \n", g_f, f_c(g_f));
    return 0;
} /* ana program sonu */
/* Fahrenheit biriminden Celcius birimine geçiş fonksiyonu
   * Gelen değer : g_f
   * Dönen değer : Celcius cinsinden sıcaklık
*/
float f_c(float g_f){ float a;
    a = (g_f-32.0)/1.8;
    return (a);}
```

Örnek 5-2 Önceki bölümde reaktör kontrolü ile ilgili bir program döngülerle yapılmıştı. Aşağıda bu programın sıkça tekrarlanan kısımları ($f1=pow((1-x),2)/pow((1+x),2)$; ve $f2=4.*x/pow((1+x),2)$;) fonksiyon alt programı içine alınarak yeniden yazılmıştır.

Program 5-2 Reaktör kontrolü programının fonksiyon kullanılarak yazılması.

```
#include <iostream>
#include <math.h>
using namespace std;
main(){ float x = 0.0, // başlangıç değeri atanır
        x1 = 2.0, // son değer
        dx = 0.1; // adım miktarı
float f1(float x), f2(float z); // fonksiyon tanimi
cout << " m2/m1 f1=K1/Ki f2=K2/Ki \n" <<
      "-----\n";
do { cout << x << " " << f1(x) << " " << f2(x) << endl;
    x = x + dx;
    } while (x < x1);
} // program sonu
// f1 ve f2 fonksiyon alt programları
float f1(float x){
    float u=pow((1-x),2)/pow((1+x),2);
    return u;}
float f2(float x){
    float z = 4.*x/pow((1+x),2);
    return z;}
```

Programın çalıştırılması sonucu elde edilen sonuç aşağıdaki gibidir:

```
m2/m1 f1=K1/Ki f2=K2/Ki
-----
0 1 0
0.1 0.669421 0.330579
0.2 0.444444 0.555556
0.3 0.289941 0.710059
0.4 0.183673 0.816327
0.5 0.111111 0.888889
0.6 0.0625 0.9375
0.7 0.0311419 0.968858
0.8 0.0123457 0.987654
0.9 0.00277008 0.99723
1 3.55271e-015 1
1.1 0.00226758 0.997732
1.2 0.00826447 0.991735
```

```
1.3 0.0170133 0.982987
1.4 0.0277778 0.972222
1.5 0.04 0.96
1.6 0.0532545 0.946745
1.7 0.0672154 0.932785
1.8 0.0816327 0.918367
1.9 0.096314 0.903686
```

```
-----
Process exited with return value 0
Press any key to continue . . .
```

Örnek 5-3 Öğrencilerin almış oldukları notlar kullanılarak başarı notunu harf olarak belirleyen fonksiyon kullanılarak hazırlanmış bir program aşağıda verilmektedir. Vize ve final sınavları girildikten sonra vize sınavının ağırlığına bağlı olarak vize ve final sınavlarının ağırlıklı notlarından başarı notu hesaplanmaktadır.

Program 5-3 Fonksiyonların kullanımı ve not hesaplanması.

```
/* fonksiyonların kullanımı
   not değerlerinin 1 ile 10 arasında girilmesi
   gerekmektedir */
#include <stdio.h>
/* fonksiyon protipi */
char notu(int not1, int not2, float a);
int main(){
int vize,
    final,
    devam;
float ağırlık;
char harf;
    devam = 1;
    ağırlık ;
    vize = 0;
    final = 0;
do{
    printf(" isleme devam etmek için 1, bitirmek için 0 giriniz\n");
    scanf("%d", &devam);
    if (devam == 1){
        printf("\n vize notu (0-10 arası) : ");
        scanf("%d", &vize);
        printf("\n final notu (0-10 arası): ");
```

```

scanf("%d", &final);
printf("\n Vizenin agirlik degeri (0.0-1.0 arasi) : ");
scanf("%f", &agirlik);
harf = notu(vize, final, agirlik);
printf(" \n gecme notu = %c \n\n", harf);
}
} while (devam == 1);
return(0);
} /* ana program sonu */
/* fonksiyon */
char notu(int not1, int not2, float a)
{ float ort;
ort = (a*(float) not1 + (1.0 - a)*(float) not2);
if (ort >= 7.0 )
return 'A';
else
return 'B';}

```

Programın çalıştırılması sonucu elde edilen sonuç aşağıdaki gibidir:

```

isleme devam etmek icin 1, bitirmek icin 0 giriniz
1

vize notu (0-10 arasi) : 3

final notu (0-10 arasi): 6

Vizenin agirlik degeri (0.0-1.0 arasi) : 0.4

gecme notu = B

isleme devam etmek icin 1, bitirmek icin 0 giriniz
0

-----
Process exited with return value 0
Press any key to continue . . .

```

Örnek 5-4 Aşağıdaki programda isim ve deger değişkenlerindeki değerlerin fonksiyon kullanılarak ekrana yazdırılması sağlanmaktadır. İçerik dosyaları için tırnak işaretleri kullanıldığına dikkat ediniz.

Program 5-4 Fonksiyon kullanarak isimlerin ekrana yazdırılması.

```

/* fonksiyonların kullanımı */
#include "stdio.h" /* printf, scanf */
#include "stdlib.h"
void yaz(char isim, int deger);
main(){ int deger;
        char isim;
        FILE *fin;

        fin = fopen("veri.dat","r"); // dosya okuma amaçlı açılır
        while (fscanf(fin,"%c %d\n", &isim, &deger) !=EOF)
            { yaz(isim, deger); }
        fclose(fin); // dosya kapatılır
return 0; }
void yaz(char isim, int deger)
{ int i;
  printf("%c",isim);
  for (i = 1; i<= deger; i++)
      printf("*");
  printf("\n");}

```

Yukarıdaki program için örnek veri.dat dosyası içeriği aşağıdadır:

```

m 6
a 7
A 20
B 10
C 78

```

Fonksiyon alt programından herhangi bir değer ana programa gelmemesi için void deyimi fonksiyon alt programı tanımlanırken kullanılır.

Örnek 5-5 Bir malın katma değer vergisi (KDV) hariç fiyatını hesaplayan bir program aşağıda verilmektedir. Programda, klavyeden malın satış fiyatı ve KDV oranı girildikten sonra $fiyat/(1.0+oran)$ formülünden KDV siz fiyatı belirlenmektedir.

Program 5-5 Bir malın KDV hariç fiyatı.

```

/* fonksiyon kullanarak fiyatın KDV siz halini hesaplar */
float kdv(float fiyat, float oran);
void main(){float fiyat, oran, kdv_siz;
printf(" fiyatı giriniz :");
scanf("%f", &fiyat);
printf("\n kdv oranını giriniz :");
scanf("%f", &oran);
kdv_siz=kdv(fiyat, oran);
printf("\n\n KDV siz degeri : %10.3f\n",kdv_siz);
printf("\nbir tusa basınız");
getch();}

```

```
/* alt program */
float kdv(float fiyat, float oran)
{ return (fiyat/(1.0 + oran));}
```

Program çalıştırılınca ekranda aşağıdaki mesajlar görüntülenir:

```
fiyati giriniz :100
kdv oranini giriniz :.18

KDV siz degeri : 84.75
```

Örnek 5-6 Klavyeden girilen iki sayıdan büyük olanını ekrana yazan bir program aşağıda verilmektedir. Programda, klavyeden girilen iki çift duyarlı sayıyı önce değişkenlere aktarır ve sonra fonksiyon çağrılarak buradaki $sayi=x>y?x:y$; satırı ile x değişkenindeki değer y değişkenindeki değerden büyük ise x değişkenindeki değer ekrana yazdırılır, değilse (burada ?) y değişkenindeki değer x e aktarılmasını sağlar.

Program 5-6 Klavyeden girilen iki sayıdan büyüğünün ekrana (fonksiyon kullanılarak) yazdırılması.

```
#include <iostream>
#include <iomanip>
#include <conio.h>
using namespace std;
double buyuk (double x, double y); // fonksiyon tanımlanır
int main () {
double x; // kullanıcının girdiği birinci rakam
double y; // kullanıcının girdiği ikinci rakam
double enbuyuk; // a place to store the larger of x and y
cout << "İki sayı giriniz : ";
cin >> x >> y;
enbuyuk = buyuk (x, y);
//fonksiyonu çağrılır & büyük değer değişkene aktarılır
cout << enbuyuk << endl;
getch();
return 0;
} // program sonu
/* iki sayıdan büyük olanı bulunur */
double buyuk (double x, double y) {
double sayi;
sayi = x > y ? x : y;
return sayi; }
```

Örnek 5-7 Görelilik teorisine göre cisimler ışık hızına ulaşırlarsa kütleleri enerjiye (ışığa) dönüşür ($E=mc^2$). Aşağıdaki algoritma ve programda fonksiyon kullanılarak maddenin ışığa dönüşümü hesaplanmaktadır.

Algoritma 5-1 Maddenin enerjiye çevrilmesi.

1. Başla
2. m değişkenine kütle değeri giriniz,
3. $c = 2.99792e8$; /* metre/s */
4. $e = m * c * c$;
5. e değişkenindeki değeri Yaz
6. Son

Program 5-7 Maddenin enerjiye dönüşümü.

```
/* maddenin enerjisi hesaplanır :  $E = m c^2$  */
#include <math.h>
float c = 2.99792e8; /* metre/s */
float E(float m, float c) {return m * c * c;}
int main ()
{ float m;
  printf(" m kutlesinin sahip olabilecegi enerji hesaplanır. \n ");
  printf("\n m kutlesinin degerini kilogram cinsinden giriniz : ");
  scanf("%f", &m);
  printf("\n\n kutle kilogram enerjiye donusturulurse = ");
  printf("%e", E(m, c), " Joule elde edilir.\n\n");
  return 0;}
```

Örnek 5-8 windows.h içerik dosyasındaki gotoxy fonksiyonunu kullanarak imlecin metin ekranında istenilen yerde olmasını sağlayan bir program aşağıda verilmektedir. gotoxy fonksiyonu imleci belirtilen satır ve sütuna gitmesini sağlar.

Program 5-8 gotoxy fonksiyonunun kullanılması.

```
#include <iostream>
#include <windows.h>
using namespace std;
void gotoxy ( short x, short y )
{
  COORD coord = {x, y};
  SetConsoleCursorPosition ( GetStdHandle ( STD_OUTPUT_HANDLE ), coord );
}
int main ()
{
  char tip0;
  char tip1;
  char tip2;
  char tip3;
  cout << "   Maddenin bazı fiziksel özellikleri           \n";
  cout << "   -----\n";
```

```

cout << " İletken [ ] Kristal [ ] Manyetik [ ] Katı [ ] \n";
cout << " -----\n";
// ekranın yatay doğrultusu (soldan sağa) y eksenini,
// ekranın düşey doğrultusu (yukarıdan aşağı) x eksenidir.
// açılan ekranda ilk satır 0, ilk sütun 0 dır.
gotoxy(12, 2); cin >> tip0;
gotoxy(26, 2); cin >> tip1;
gotoxy(41, 2); cin >> tip2;
gotoxy(52, 2); cin >> tip3;
return 0;
}

```

Programın çalıştırılması sonucu ekran görüntüsü aşağıda gibidir:

```

Maddenin bazı fiziksel özellikleri
-----
İletken [X] Kristal [X] Manyetik [X] Katı [X]
-----
Process exited with return value 0
Press any key to continue . . .

```

Özet olarak, fonksiyonlar, tekrar kullanılabilen program parçalarıdır. Programlar, fonksiyonların kullanılması ile daha sade bir görünüm almakta, kolay kontrol/takip edilebilir ve daha hızlı olmaktadır. Bir programı bir ekip hazırlarken yapılan iş bölümü (fonksiyonların hazırlanması) ile programlar, daha hızlı ve kısa sürede tamamlanabilir. Bir fonksiyon çağrılırken, parantez içindeki argümanlar veya parametreler fonksiyon alt programın içinde kullanılabilirler. Alt programdaki değişkenlerin sakladıkları değerler diğer bellek alanlarına transfer edilemez. Bir alt program içinden başka bir alt program çağrılabilir. void deyimi ile fonksiyon alt programından, bu alt programın çağrıldığı yere herhangi bir değer dönmemesi sağlanabilir. main fonksiyonu (veya ana programı) sadece 1 tane olur ve başka yerlerden çağrılmaz. Her fonksiyon alt programı } ile kapatılmadan önce dönecek değeri gösteren return deyiminin kullanılması gerekmektedir. return deyiminde fonksiyonun çağrıldığı yere dönecek değer yazılmalıdır. Kullanılan fonksiyonlara isim verilirken C veya C++ yazım kurallarına dikkat edilmelidir.

5.2 SORULAR

1. Vektör bileşenleri klavyeden girilen **A** ve **B** vektörleri arasındaki $C=A+B$, $D=A-B$ ve $E=2A-B$ (vektörel) işlemlerin her birini ayrı fonksiyonlarda yapan ve sonuçları ekrana yazan bir C++ programı hazırlayınız.
2. x-y koordinat sisteminde verilen bir P_1 noktasına ait konum bilgileri $P_1(x_1, y_1)$ şeklindedir. Klavyeden girilen bir α açısına göre x-y koordinat sistemini döndürüp bu noktanın yeni koordinat

değerlerini $P_2(x_2, y_2)$ şeklinde $x_2=x_1\cos\alpha+y_1\sin\alpha$ ve $y_2=-x_1\sin\alpha+y_1\cos\alpha$ fonksiyonları ile hesaplayan bir C++ programını fonksiyon alt programını kullanarak hazırlayınız.

3. $a=1 \text{ m/s}^2$ ve $b=1 \text{ m/s}$ (ivme ve sürat) olmak üzere konum fonksiyonu $x(t) = \frac{1}{2}a \cdot t^2 + 2 \cdot b \cdot t$ fonksiyonu şeklinde olan bir cisme ait hız değerlerini $t=0$ ncı saniyeden başlamak üzere $\Delta t=1 \text{ s}$, 0.5 s ve 0.1 saniye aralık değerleri olarak $t=2$ nci saniyeye kadar hesaplayan bir C++ programı hazırlayınız (ayrıca programınızı $v_{\text{ort}} = \frac{x_{\text{son}} - x_{\text{ilk}}}{\Delta t}$ hesaplayacak şekilde düzenleyiniz).

C/C++ Ders Notları-A.Ü. Fizik Mühendisliği Bölümü