



Şekil 4-3 do ... while döngüsünün çalışması.

Burada önce döngü içine girilir, işlem1, işlem2,... işlemler gerçekleştirilir ve sonra işlemlerin devam edip edilmeyeceği hakkında while (koşul); ile verilir (Şekil 4-3 deki (1) numaralı kısım) sorgulamasına geçilir. Koşulun sorgulanmasından olumlu bir cevap (doğru) alınıyorsa döngüye devam edilir (işlemler bölgesi), olumsuz cevap (yanlış) alınıyorsa döngü sonlandırılır ve (2) bölgesine geçilir.

Örnek 4-5 x değişkenindeki değer 0 dan daha küçük bir değere ulaşana kadar işlem yapan bir program parçası aşağıdaki verilmektedir.

```
int x;
x = 2;
do{
    x=x-1;
    printf("x = %d\n",x);}
while (x>=0)
```

Yukarıdaki program parçasında x tamsayı olarak tanımlanmış ve ilk değeri 2 olarak atanmıştır. Tanımlama ve atama işlemlerinden sonra do...while döngüsü içine girilir ve x değişkenindeki değer 1 eksiltilir. Ekrana printf ile mesaj yazdırıldıktan sonra x>=0 kontrolü yapılarak buradan alınacak cevap doğru olduğu sürece döngü içinde kalınır ve x in değerinin azaltılma işlemine devam edilir. x değişkenindeki değer 0 dan küçük olunca do...while döngüsü dışına çıkarılır.

4.3.1 İç içe do ... while döngülerin kullanılması. (9ncu hafta)

Probleminizi çözmek için iç içe do...while döngüleri kullanmanızı gerektirebilir. Bu aşağıdaki satırlardaki gibi yazılabilir:

```
do{
    işlemler_0;
    do{
        işlemler_1;
    } while (koşul_1);
}
```

```

do{
    İşlemler_2;
} while (koşul_2);
} while (koşul_0);

```

Örnek 4-6 Trigonometrik fonksiyonları sonsuz seriler halinde yazmak mümkündür. Bu işlemi bilgisayara yaptırmak imkansızdır. Hesaplamanın belirli bir yerde kesilmesi (truncate) gerekir. Böylece sonsuz seri sonlu seri şekline getirilir. Örnek olarak $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$ şeklinde veya $\sin x = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}$ şeklinde verilebilir. Son denklemdaki toplam n=1 den sonsuza kadardır. Önce de belirtildiği gibi bilgisayarla seri işlemi sonsuz sayıda yapılamayacağı için işlem bir yerde kesilir. Klavyeden derece cinsinden a değişkenine girilen bir değeri radyana çevirip x değişkeninde saklayan ve bu radyanın sinüs değerini k=1, 2, ... 10 (serinin üst sınır değeri) hesaplayan algoritma ve C/C++ programı aşağıda verilmektedir.

Algoritma 4-3 $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$ ifadesini hesaplayan algoritma.

```

1. Başla
2. int f, i, n=1, k=10;
3. float a, x, t=0;
4. Yaz " Açığı derece cinsinden giriniz: ";
5. a değişkenine değer gir;
6. x=3.14159*a/180.0;
7. n=1;
8. do { f=1; i=1;
    do { f=f*i; i=i+1;
    } while (i <= (2*n-1));
    t=t+pow(-1, n-1)*pow(x, 2*n-1)/(float)f;
    n=n+1;
} while (n<k);
9. Yaz a, t, sin(x);
10. Son

```

Örnek 4-7 Bir biyoloğun incelediği bir bakteri kültürü grubu A ortamında her 60 dakikada bir üremektedir (örneğin 1 iken 2, 2 iken 4 olmaktadır). Aynı bakteri kültürü B ortamında ise her 90 dakikada bir üremektedir. Aşağıdaki C programı 24 saat boyunca bu bakteri kültürünün nüfusunu iki ayrı ortamda ve her 3 saatte bir (180 dakika) ekrana yazmaktadır.

Algoritma 4-4 Bakteri kültürü nüfusunun sayımı.

```

1. Başla
2. A ortamındaki bakteri sayısı değişkeni a ya değer giriniz;
3. B ortamındaki bakteri sayısı değişkeni b ye değer giriniz;

```

4. saat deęişkeni s=0;
5. Eęer (s>1440) ise Git 11
6. s=s+30;
7. Eęer (s%60==0) ise a*=2;
8. Eęer (s%90==0) ise b*=2;
9. Eęer (s%180==0) ise Yaz s/60, " saat sonra A=", a, " B=", b;
10. Git 5
11. Son.

Program 4-6 Bakteri kltr nfusunun sayımı.

```
#include <stdio.h>
#include <conio.h>
int main(){
// A ortamındaki bakteri sayısı deęişkeni
long int a=1,
// B ortamındaki bakteri sayısı deęişkeni
b=1; int s=0; // saat deęişkeni
do { s+=30; // sre artırılır
//60 dk ve katlarında A bakterisi katlanır
if (s%60==0) a*=2;
//90 dk ve katlarında B bakterisi katlanır
if (s%90==0) b*=2;
if (s%180==0)
printf("\n saat %2d , A = %9li , B = %9li ", s/60, a, b);
} while (s<1440); // do...while sonu
getch(); return 0;}
```

rnek 4-8 Bu rnek programda veri.txt isimli dosyadan veriler dosya sonuna gelene kadar okunan veriler yani rakamlar toplanır ve ortalama deęerin hesaplanacağı satıra geilerek elde edilen deęer ort.txt isimli dosyaya yazdırılır. İřlemler bittikten sonra dosyalar kapatılmaktadır. Dosya kapatma iřlemi yapılmazsa dosyalar zarar grebilir. Fakat program istem dıřı bitse veya sonlandırılrsa da, dosyalar derleyiciler veya iřletim sistemi tarafından otomatik olarak kapatılırlar.

Algoritma 4-5 Bakteri kltr nfusunun sayımı.

1. Bařla
2. veri.txt ve ort.txt dosyalarını kullanıma aınız;
3. t=0; i=0;
4. veri.txt den x deęişkenine deęer oku, dosya sonu ise Git 8;
5. t=t+x;
6. i=i+1;
7. Git 4;

8. ort=t/i;
9. Yaz ort;
10. veri.txt ve ort.txt dosyalarını kullanıma kapat;
11. Son.

Program 4-7 Bir dosyadan okunan verilerin işlendikten sonra başka bir dosyaya yazılması

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
int main()
{
    /* kullanılacak veri tipleri tanımlanır */
    int i;
    float x, x_tpl, x_ort;
    /* dosya işlemleri */
    FILE *fin, *fout; // dosyalar işaretçi olarak tanımlanır
    fin = fopen("veri.txt","r"); /* dosya okuma amaçlı açılır */
    fout = fopen("ort.txt","w"); /* dosya yazma amaçlı açılır */
    x_tpl=0;
    i=1;
    while (fscanf(fin,"%f", &x) !=EOF){
        /* dosya sonuna kadar okuma işlemi devam eder
        dosyadan okunan veriler x değişkenine aktarılır */
        x_tpl=x_tpl+x;
        i=i+1; } // while bloğu sonu
    x_ort=x_tpl/ (float) i;
    fprintf(fout,"ortalama deger=%f\n",x_ort);
    // dosyaya veri yazdırılır
    fclose(fin); // dosyalar kapatılır
    fclose(fout); } // dosyalar kapatılır
```

Buradaki örnek veri.txt dosyası aşağıdaki gibi olabilir:

```
1.2
2.3
2.0
1.8
10.0
1.5
3.4
5.3
81.9
```

Örnek 4-9 Aşağıdaki programda #include <fstream.h> satırı ile dosya işlemlerinin yapılacağı programa bildirilir. Buradaki f file anlamındadır. fstream dosyal("test.txt",ios::in); ifadesi ise "test.txt" isimli dosyanın sadece okuma amaçlı (in) kullanacağını göstermektedir. Bu

ifadedeki ios input/output status yani giriş/çıkış durumu anlamındadır. Dosyadaki veriler dosya sonuna ulaşmaya kadar bir döngü içinde okunur.

Program 4-8 C++ da dosyadan veri okuma.

```
#include <fstream.h>
int main(){ char str[50];
    fstream dosyal("test.txt",ios::in);
// ios:in-dosyadan veri okuma
// ios:out-dosyaya veri yazma
// ios:app-dosyaya veri ekleme
// ios:binary-ikili kodda dosya işlemi
    while(dosyal >> str) cout << str ;
    dosyal.close();
    return 0; }
```

Örnek 4-10 Bir cismin zamana bağlı olarak sürat değerleri bir dosyada saklanmaktadır. Cismin ortalama ivmesini (ilk ve son değerlerini kullanarak) hesaplayan C++ programı aşağıda verilmektedir. İlk okuma için burada sayaç kullanılmaktadır. Sayaçın ilk değerine bakılır.

Program 4-9 Ortalama ivme hesabı.

```
// #include "stdafx.h" MS Visual Studio 2005 te eklenecek
#include <iostream>
#include <stdio.h>
#include <conio.h>
#include <string>
int main()
// int _tmain(int argc, _TCHAR* argv[])
// MS Visual Studio 2005 te
// yukarıdaki açıklama satırının başlangıcındaki // siliniz.
{/* kullanılacak veri tipleri tanımlanır */
    int i=1;
    float t=0, t_ilk=0, t_son=0,
        v=0, v_ilk=0, v_son=0, a_ort=0;
    char satir[40];
    * dosya islemleri için tanımlamalar yapılır*/
    FILE *dosya1, *dosya2;
// kullanılacak dosyalar açılır
    dosya1 = fopen("veri.txt", "r");
    dosya2 = fopen("sonuc.txt", "w");
    fscanf(dosya1, "%s", &satir); // dosyadan satır okuma yapılır
    printf("%s ", satir); // ekrana mesaj yazdırılır
    fscanf(dosya1, "%s", &satir);
    printf("%s \n", satir);
```

```

while (!feof(dosya1)) // dosya sonuna kadar okuma yapılır
{
    fscanf(dosya1,"%f %f", &t, &v);
    printf("%d %f %f\n", i, t, v);
    if (i == 1) {t_ilk=t; v_ilk=v;}
    i=i+1;
}

t_son=t;
v_son=v;
a_ort=(v_son-v_ilk)/ (t_son-t_ilk);
fprintf(dosya2,"ortalama deęer = %4.2f \n",a_ort);
// sonuc.txt isimli dosyaya veri yazdırılır
fclose(dosya1); // dosyalar kapatılır
fclose(dosya2); // dosyalar kapatılır
getch();
return 0;
}

```

Yukarıdaki program içindeki açıklama satırlarını okuyunuz. Verilerin bulunduğu veri.txt dosyası aşağıdaki gibi olmalıdır (satırlar arasında boşluk bulunmamalıdır): Aşağıdaki verileri bir metin editöründe yazıp bilgisayara saklayabilirsiniz.

Çizelge 4-1 Zamana bağlı olarak sürat değerleri.

t (saniye)	v (metre/saniye)
0.0	0.0
0.5	1.0
1.0	3.0
1.5	4.5
2.0	7.0
2.5	9.5
3.0	10.5
3.5	12.0
4.0	14.0
4.5	15.0
5.0	17.3

Örnek 4-11 0-1 arasında rasgele olarak üretilen sayıların ortalamasının hesaplanması için aşağıda bir algoritma ve program verilmektedir.

Algoritma 4-6 Rasgele sayı üretimi.

1. Başla
2. i=0;
3. n=10;
4. tpl=0.0;
5. Rasgele sayı üreticini aç;
6. Eğer i>n ise Git 10;
7. x = rand()%15;
8. tpl = tpl + x;

9. Git 6;
10. ort = (float) tpl / (float) n;
11. ort deęişkenindeki deęeri Yaz;
12. Son.

Program 4-10 Rasgele sayıların türetilmesi.

```
/* rasgele olarak uretilen n sayinin ortalamasi hesaplanir*/
#include <stdlib.h>
#include <time.h>
#include <stdio.h>
#include <conio.h>
main(void){
int i, n, tpl, x;
float ort;
n = 10;
randomize();
clrscr();
tpl = 0.0;
for(i=1; i <= n; i++)
{ x = rand()%15;
printf("%d \n", x);
tpl = tpl + x; }
ort = (float) tpl / (float) n;
printf("\n\n toplami %d olan %d adet sayinin \n", tpl, n);
printf(" ortalamasi = %5.2f dir.\n", ort);
getch();}
```

Örnek 4-12 $m=5$ kg lık bir kütleyle sahip bir cisim başlangıç noktasından başlayarak x eksenini boyunca hareket etmektedir. Parçacık üzerine etkiyen net kuvvetler eşit aralıklardaki uzaklığa bağlı olarak aşağıdaki çizelgedeki verilmektedir.

Çizelge 4-2 x-ekseni boyunca hareket eden cisme etkiyen deęişken kuvvet.

x (metre)	F (Newton)
1	27.0
2	28.3
3	36.9
4	34.0
5	34.5
6	34.5
7	46.9
8	48.2
9	50.0
10	63.5
11	13.6
12	12.2
13	32.7
14	46.6
15	27.9

Yukarıdaki çizelgedeki verileri bir dosyadan okuduktan sonra toplam işi $W = \sum_{i=1}^N F\Delta x$ formülüne göre hesaplayan algoritma ve bir C/C++ programı aşağıda verilmektedir.

Algoritma 4-7 Değişken kuvvet için toplam işin hesaplanması.

1. Başla
2. F değerlerini oku
3. dx=1;
4. F*dx değerlerini topla;
5. W değişkenindeki değeri yaz
6. Son

Program 4-11 Değişken kuvvet için toplam işin hesaplanması (C).

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
int main(){
    int i=1;
    float F, m=5.0, x, dx=1.0, W=0.0;
    char satir[40];
    /* dosya islemleri için tanımlamalar yapılır*/
    FILE *dosyal;
    // kullanılacak dosyalar açılır
    dosyal = fopen("veri.dat", "r");
    fscanf(dosyal, "%s", &satir); // dosyadan satır okuma yapılır
    printf("%s ", satir); // ekrana mesaj yazdırılır
    fscanf(dosyal, "%s", &satir);
    printf("%s \n", satir);
    while (!feof(dosyal)) // dosya sonuna kadar okuma yapılır
    {
        fscanf(dosyal, "%f %f", &x, &F);
        printf("%2d %2.2f %2.2f\n", i, x, F);
        W=W+F*dx;
        i++;
    }
    printf("Toplam iş = %f Joule\n", W);
    fclose(dosyal); // dosyalar kapatılır
    getch();
    return 0;
}
```

Yukarıdaki problem için dosyadan okunan veriler üzerinde işlemler yapıldıktan sonra ekrana biçimli çıktı verilmektedir.

Program 4-12 Değişken kuvvet için toplam işin hesaplanması.

```
#include "stdafx.h" // MS Visual Studio 2005 te gerekli
```



```

#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    char satir[80];
    ifstream dosyal; // dosya okuma amaçlı (input) tanımlanır
    dosyal.open ("veri.txt", ios::in);
    float F=0, dx=0, W=0.0, x=0, x1=0;
    int i=1;
    if (!dosyal.good ()) // dosya kontrolü yapılır
    {
        cout << "Hata: dosya açılmıyor\n"; // dosya yoksa
        return 1;
    } else
    {
        dosyal.getline(satir, sizeof (satir));
        cout << satir << " dx(m) W(Joule)" << endl;
        cout << "-----" << endl;
        dosyal >> x1 >> F;
        while (!dosyal.eof()) // dosya sonuna kadar okuma yapılır
        {
            dx=x1-x;
            W=W+F*dx;
            x=x1;
            cout << setw(2) << fixed << setprecision(0) << x1;
            cout << " ";
            cout << setw(5) << fixed << setprecision(1) << F;
            cout << " " << dx << " ";
            cout << F*dx << endl;
            dosyal >> x1 >> F;
        } // while döngüsü sonu
    } // if else sorgusu sonu
    cout << "-----" << endl;
    cout << " Yapılan toplam is : " << W << " Joule" << endl;
    dosyal.close();system("PAUSE");
    return 0;} // ana program sonu

```

Örnek 4-13 Besinlere ait çeşitli bilgiler bir dosyadan okunup kullanıcıya sunulmaktadır. Yani ekrandan gösterilmektedir. Veriler Çizelge 4-3 deki gibidir ve verileri dosyadan okuyacak bir C++ programı aşağıda verilmektedir.

Çizelge 4-3 Çeşitli besinlerin kalori, protein, yağ, VitA ve kalsiyum değerleri.

kalori	Protein	Yağ	VitA	Kalsiyum
(gram)	(gram)	/gram	(IU)	(mgram)

Ispanak (1 bardak)	23	3.0	0.3	8100	93
Şekerli patates (1 kepçe)	160	2.0	1.0	9230	46
Yoğurt	230	10.0	3.0	120	343
Yağlı süt (1 bardak)	85	8.0	0.0	500	302
Pirinç (1 bardak)	178	3.8	0.9	0	18
Beyaz ekme (1 dilim)	65	3.0	1.0	0	24
Karpuz (1 dilim)	110	2.0	1.0	2510	30
Tuna balığı (1 lb)	975	126.8	3.6	0	73
Papatya (1 lg)	156	2.4	0.4	7000	80
Istakoz (1)	405	28.8	266.0	984	190

Program 4-13 Bazı besinlerin kalori, protein, yağ, VitA ve kalsiyum değerlerinin dosyadan okunması.

```
#include "stdafx.h" // MS-studio 2005 te eklenecek
#include <iostream> // klavye-monitor giriş/çıkış
#include <fstream> // dosya işlemi giriş/çıkış
#include <conio.h> // konsol giriş/çıkış
using namespace std;
int main(){ char DosyaAdi[20]; // olası dosya uzunluğu
char satir[80]; // dosyadaki her satır 80 karakter kabul ediliyor
ifstream dosyal; // dosyal tanımlanır
cout << "Dosya adını giriniz:\n";
cin.getline (DosyaAdi, sizeof (DosyaAdi)); // dosya adı okunur
dosyal.open (DosyaAdi, ios::in); // dosya okuma amaçlı açılır
if (!dosyal.good ()) // dosya kontrolü yapılır
{ cout << "Hata: dosya açılmıyor\n";
getch();
return 1;
} else
{ while (!dosyal.eof()) // dosya sonuna kadar okuma yapılır
{ // dosyadan satır okunur
dosyal.getline(satir, sizeof (satir));
// dosyadan okunan satırlar ekrana yazdırılır
cout << satir << endl; }
getch(); // bir tuşa basılması beklenir
dosyal.close (); // dosyal kapatılır
return 0; // işletim sistemine hata yok bilgisi yollanır
} } // if .. else .. blok sonu, program sonu
```

4.4 SORULAR

1. Bir sınıftaki 20 öğrencinin 10 tanesi erkek, 10 tanesi de kızdır. 10 erkek ve 10 kız öğrencinin boy bilgilerini klavyeden (for döngüsü ile) okuyan ve sınıfın boy ortalamasını bu bilgilerden yararlanarak hesaplayan bir algoritma ve C++ programı yazınız.

2. Bir tungsten flamanın elektriksel direnci sıcaklığa bağı olarak $R=R_{20}(1+0.0045(T-20))$ formülündeki gibi değişmektedir. Denklemdaki R_{20} (=10 Ohm) elektriksel direncin 20 °C deki değeri (bu genellikle soğuk direnç olarak adlandırılır), T °C cinsinden sıcaklıktır. Sıcaklığı (T) –20 °C den başlatıp 50°C ye kadar 5°C adımlarla artırarak R elektriksel dirincini hesaplayan bir algoritma ve C++ programını (for döngüsü kullanarak) hazırlayınız.
3. Adınızı ve soyadınızı ekrana alt alta 15 kere yazan bir C++ programı (for döngüsünü kullanarak) hazırlayınız.
4. 10 galon dan 20 galona kadar 1 er galon aralıklarla verilen sıvı ölçüm birimini litre ölçüm birimine dönüştürüp sonuçları ekrana yazan C++ programını yazınız. 1 galon = 3.785 litredir.
5. Saattaki ortalama hızınızı 90 km/saat den 120 km/saat e kadar 2 şer km/saat artırarak Ankara ile İstanbul arasındaki 454 km lik yolu kaç saatte alabileceğinizi hesaplayan ve sonuçları ekrana yazan C++ programını yazınız.