

BİL-142 Bilgisayar Programlama II (C/C++)

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Konular

- Giriş
- Data Hiyerarşisi
- Files (Dosyalar) ve Streams (Kaynaklar)
- Sıralı Dosya Oluşturma
- Sıralı Dosyadan Data Okuma
- Sıralı Dosyaları Güncelleme
- Random-Access (Rastgele Erişimli) File
- Rastgele Erişimli Dosya Oluşturma
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- Rastgele Erişimli Dosyadan Sıralı Okuma

Giriş

- Değişkenler ve diziler hafızada geçici olarak saklanır.
- Dosyalar verilerin sürekli saklanması için kullanılır.
- Bilgisayarlar verileri ikincil depolama cihazları olarak bilinen manyetik disklerde, optik disklerde veya flash memory gibi kalıcı hafıza birimlerinde saklar.

3

Konular

- Giriş
- **Data Hiyerarşisi**
- Files (Dosyalar) ve Streams (Kaynaklar)
- Sıralı Dosya Oluşturma
- Sıralı Dosyadan Data Okuma
- Sıralı Dosyaları Güncelleme
- Random-Access (Rastgele Erişimli) File
- Rastgele Erişimli Dosya Oluşturma
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- Rastgele Erişimli Dosyadan Sıralı Okuma

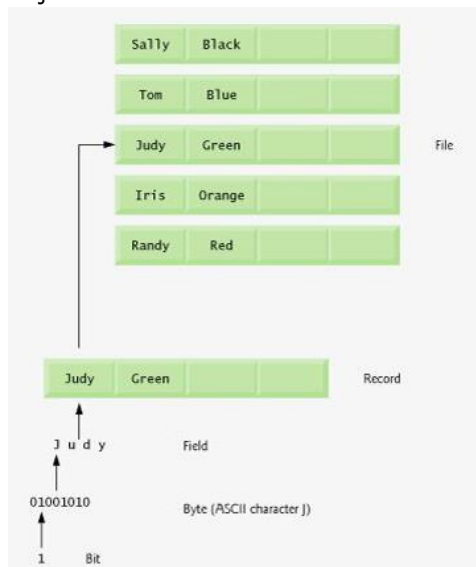
Data Hiyerarşisi

- Bilgisayarın işlediği tüm veriler en sonunda 1 ve 0'lara dönüşür.
- Bilgisayarın desteklediği en küçük veri parçasına **bit** (binary digit) denir.
- 8 bit bir **byte** oluşturur.
- Karakterler bit gruplarıyla gösterilir.
- Alanlar (**fields**) karakter gruplarıyla gösterilir.
- Bilgisayardaki veriler arasında hiyerarşik bir yapı vardır.
- Bir kayıt (**record**) birden fazla alandan oluşabilir.
- Bir dosya (**file**) ise birden fazla kayıttan oluşabilir.

5

Data Hiyerarşisi

- Örnek veri hiyerarşisi



6

Data Hiyerarşisi

- İstenen bir kaydı dosyadan almak için **record key** kullanılır.
- Record key bir kaydı diğer tüm kayıtlardan ayıran alandır.
- Dosya içinde kayıtları organize etmenin farklı yolları vardır. Bunların arasında yaygın kullanılanlardan birisi sıralı dosya (**sequential file**) yapısıdır.
- Veritabanı (**Database**) birden fazla dosyadan oluşur.
- Veritabanını oluşturan ve yöneten programlara **Veritabanı Yönetim Sistemi (DBMS-Database Management Systems)** denir.

7

Konular

- Giriş
- Data Hiyerarşisi
- **Files (Dosyalar) ve Streams (Kaynaklar)**
- Sıralı Dosya Oluşturma
- Sıralı Dosyadan Data Okuma
- Sıralı Dosyaları Güncelleme
- Random-Access (Rastgele Erişimli) File
- Rastgele Erişimli Dosya Oluşturma
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- Rastgele Erişimli Dosyadan Sıralı Okuma

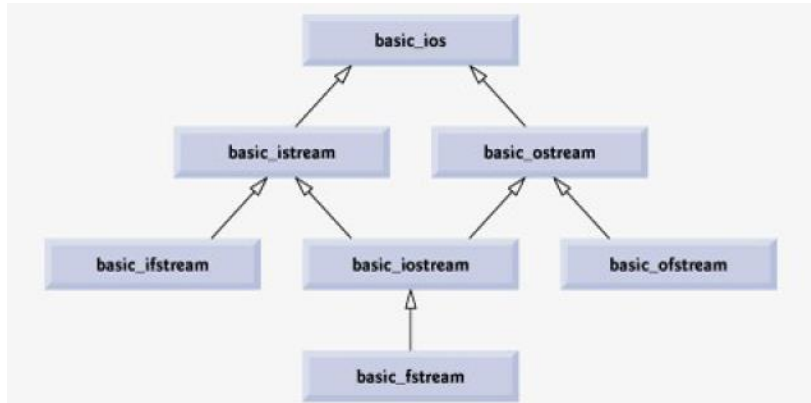
Files (Dosyalar) ve Streams (Kaynaklar)

- C++ her dosyayı byte serileri olarak görür.
- Her dosya, bir dosya sonu işaretçisi (end-of-file marker) ile sonlanır.
- Bir dosya açıldığında bir nesne oluşturulur ve kaynak bu nesneyle ilişkilendirilir.
- Bu nesneyle ilişkilendirilen kaynak (stream) program ile dosya veya cihaz arasında bir iletişim kanalı oluşturur.
- Örneğin `cin` nesnesi klavyeden girişleri okur, `cout` nesnesi ekrana veri gönderir.

9

Files (Dosyalar) ve Streams (Kaynaklar)

- C++ ile dosya işlemleri yapmak için `<iostream>` ve `<fstream>` dosyalarını include etmek gerekir.
- `<fstream>` başlık dosyası dosya giriş ve çıkış işlemleri için gerekli tanımlamaları içerir.



10

Konular

- Giriş
- Data Hiyerarşisi
- Files (Dosyalar) ve Streams (Kaynaklar)
- **Sıralı Dosya Oluşturma**
- Sıralı Dosyadan Data Okuma
- Sıralı Dosyaları Güncelleme
- Random-Access (Rastgele Erişimli) File
- Rastgele Erişimli Dosya Oluşturma
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- Rastgele Erişimli Dosyadan Sıralı Okuma

Sıralı Dosya Oluşturma

- C++ dosya yapısıyla ilgili kısıtlamaya sahip değildir.
- Programcı dosya yapısını kendi istediği gibi oluşturabilir.
- Kayıt (record) yaklaşımı C++ dosyalarında tanımlı değildir.
- Programcının kayıt yapısını ve ona uygun yazma ve okuma işlemlerini oluşturması gerekir.

Sıralı Dosya Oluşturma

```
1 // Fig. 17.4: Fig17_04.cpp
2 // Create a sequential file.
3 #include <iostream>
4 using std::cerr;
5 using std::cin;
6 using std::cout;
7 using std::endl;
8 using std::ios;
9
10 #include <fstream> // file stream
11 using std::ofstream; // output file stream
12
13 #include <cstdlib>
14 using std::exit; // exit function prototype
15
16 int main()
17 {
18     // ofstream constructor opens file
19     ofstream outClientFile( "clients.dat", ios::out );
20
21     // exit program if unable to create file
22     if ( !outClientFile ) // overloaded ! operator
23     {
24         cerr << "File could not be opened" << endl;
25         exit( 1 );
26     } // end if
```

13

Sıralı Dosya Oluşturma

```
27
28     cout << "Enter the account, name, and balance." << endl
29         << "Enter end-of-file to end input.\n? ";
30
31     int account;
32     char name[ 30 ];
33     double balance;
34
35     // read account, name and balance from cin, then place in file
36     while ( cin >> account >> name >> balance )
37     {
38         outClientFile << account << " " << name << " " << balance << endl;
39         cout << "? ";
40     } // end while
41
42     return 0; // ofstream destructor closes file
43 } // end main
```

14

Sıralı Dosya Oluşturma

- Veri girişi ve Notepad ile görüntülenen dosya içeriği

```
c:\m.ali\fileornek\Debug\fileornek.exe
Enter the account, name, and balance.
Enter end-of-file to end input.
? 1020 Ahmet 10000
? 2110 Lale 2300
? 1324 Meryem 234000
? 3421 Mehmet 321000
? ^Z
```

```
clients.dat - Notepad
File Edit Format View Help
1020 Ahmet 10000
2110 Lale 2300
1324 Meryem 234000
3421 Mehmet 321000
```

15

Sıralı Dosya Oluşturma

- Dosyalar **ifstream**, **ofstream**, **fstream** ile açılır.
- ofstream** dosya sadece yazmak için açılmıştır.
- ofstream** constructor'ına iki parametre gönderilmiştir. Dosya adı ve dosya açma modu.
- ios::out** ile dosya varsa truncate edilerek boşaltılır, yoksa yeniden açılır.

Mode	Description
<code>ios::app</code>	Append all output to the end of the file.
<code>ios::ate</code>	Open a file for output and move to the end of the file (normally used to append data to a file). Data can be written anywhere in the file.
<code>ios::in</code>	Open a file for input.
<code>ios::out</code>	Open a file for output.
<code>ios::trunc</code>	Discard the file's contents if they exist (this also is the default action for <code>ios::out</code>).
<code>ios::binary</code>	Open a file for binary (i.e., nontext) input or output.

16

Sıralı Dosya Oluşturma

- Satır 19'da `outClientFile` nesnesi "clients.dat" dosyasıyla ilişkilendirilmiştir.
- `ofstream outClientFile("clients.dat");` ile default olarak yazmak için açılır.
- `ofstream outClientFile;` ile belirli bir dosya açılmaz dosya daha sonra belirlenir.
- Dosya açma işlemi başarılı olursa `outClientFile` true, değilse false döndürür.

17

Sıralı Dosya Oluşturma

- Satır 22-26 dosya açma işleminin başarılı olup olmadığına bakar.
- Satır 36'da aralarında boşlukla girilen değerler sırasıyla değişkenlere aktarılır.
- While döngüsünden Windows'ta ctrl+z, Unix'te ctrl-d ile çıkılır.
- Programdan çıkınca `outClientFile` nesnesinin destructor'ı çağrılır ve açılan "clients.dat" dosyası kapatılır.
- İstenirse `outClientFile.close();` ile dosya doğrudan kapatılabilir.
- Dosya kullanımı bitmişse kapatılması gerekir.

18

Konular

- Giriş
- Data Hiyerarşisi
- Files (Dosyalar) ve Streams (Kaynaklar)
- Sıralı Dosya Oluşturma
- **Sıralı Dosyadan Data Okuma**
- Sıralı Dosyaları Güncelleme
- Random-Access (Rastgele Erişimli) File
- Rastgele Erişimli Dosya Oluşturma
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- Rastgele Erişimli Dosyadan Sıralı Okuma

Sıralı Dosyadan Data Okuma

- Bir dosya `ifstream` nesnesiyle giriş yapmak amacıyla olarak açılır.
- `ifstream` constructor'ı dosya adı ve dosya açma modunu alır.
- Bir dosyanın içeriği değişmeyecekse açma modu `ios::in` olarak girilmelidir.

Sıralı Dosyadan Data Okuma

```
1 // Fig. 17.7: Fig17_07.cpp
2 // Reading and printing a sequential file.
3 #include <iostream>
4 using std::cerr;
5 using std::cout;
6 using std::endl;
7 using std::fixed;
8 using std::ios;
9 using std::left;
10 using std::right;
11 using std::showpoint;
12
13 #include <fstream> // file stream
14 using std::ifstream; // input file stream
15
16 #include <iomanip>
17 using std::setw;
18 using std::setprecision;
19
20 #include <string>
21 using std::string;
22
23 #include <cstdlib>
24 using std::exit; // exit function prototype
25
```

21

Sıralı Dosyadan Data Okuma

```
26 void outputLine( int, const string, double ); // prototype
27
28 int main()
29 {
30     // ifstream constructor opens the file
31     ifstream inClientFile( "c:\\clients.dat", ios::in );
32
33     // exit program if ifstream could not open file
34     if ( !inClientFile )
35     {
36         cerr << "File could not be opened" << endl;
37         exit( 1 );
38     } // end if
39
40     int account;
41     char name[ 30 ];
42     double balance;
43
44     cout << left << setw( 10 ) << "Account" << setw( 13 )
45         << "Name" << " Balance" << endl << fixed << showpoint;
46
47     // display each record in file
48     while ( inClientFile >> account >> name >> balance )
49         outputLine( account, name, balance );
50
51     system("PAUSE");
52     return 0; // ifstream destructor closes the file
53 } // end main
54
55 // display single record from file
56 void outputLine( int account, const string name, double balance )
57 {
58     cout << left << setw( 10 ) << account << setw( 13 ) << name
59         << setw( 9 ) << setprecision( 2 ) << right << balance << endl;
60 } // end function outputLine
```

22

Sıralı Dosyadan Data Okuma

- Ekran çıkışı ve Notepad ile görüntülenen dosya içeriği

```
c:\Documents and Settings\m.ali\My Document
Account Name Balance
1020 Ahmet 10000.00
2110 Lale 2300.00
1324 Meryem 234000.00
3421 Mehmet 321000.00
Press any key to continue . . .
```

```
clients.dat - Notepad
File Edit Format View Help
1020 Ahmet 10000
2110 Lale 2300
1324 Meryem 234000
3421 Mehmet 321000
```

23

Sıralı Dosyadan Data Okuma

- `ifstream inClientFile("clients.dat");` ile dosya default olarak okumak için açılır.
- `!inClientFile` ile dosyanın sorunsuz açıldığı kontrol edilir.
- Satır 48'de her iterasyonda dosyadan bir kayıt (satır) okunur.
- Dosya `istream` ve `ostream` nesnelere dosya pozisyon işaretçisine sahiptir.
- `istream` için `seekg` (seek get), `ostream` için `seekp` (seek put).
- `inClientFile.seekg(0);` ile dosya başına gidilir.

24

Sıralı Dosyadan Data Okuma

- Dosya üzerinde yeniden konumlanmak için ikinci bir parametre daha girilebilir. Bu parametre yön bildirir.

```
// position to the nth byte of fileObject (assumes ios::beg)
fileObject.seekg( n );

// position n bytes forward in fileObject
fileObject.seekg( n, ios::cur );

// position n bytes back from end of fileObject
fileObject.seekg( n, ios::end );

// position at end of fileObject
fileObject.seekg( 0, ios::end );
```

- **ios::cur** bulunulan noktayı, **ios::beg** başlangıcı ve **ios::end** dosya sonunu gösterir.

25

Sıralı Dosyadan Data Okuma

- Dosya üzerinde bulunulan pozisyonu almak için **tellg** ve **tellp** kullanılır.
- Aşağıdaki kod long integer olarak bulunulan noktayı verir.

```
location = fileObject.tellg();
```

26

Konular

- Giriş
- Data Hiyerarşisi
- Files (Dosyalar) ve Streams (Kaynaklar)
- Sıralı Dosya Oluşturma
- Sıralı Dosyadan Data Okuma
- **Sıralı Dosyaları Güncelleme**
- Random-Access (Rastgele Erişimli) File
- Rastgele Erişimli Dosya Oluşturma
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- Rastgele Erişimli Dosyadan Sıralı Okuma

Sıralı Dosyaları Güncelleme

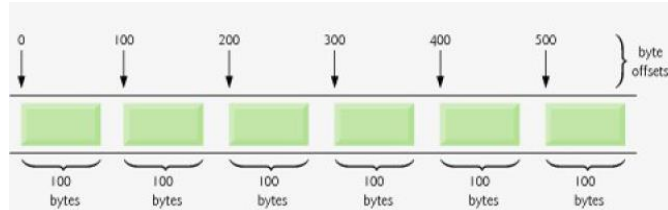
- Sıralı dosyalarda data güncelleme işlemi sırasında diğer datalarda bozulma riski vardır.
- Yeni bilginin boyutu öncekinden büyükse sonraki bilgileri etkileyecektir.
- Aşağıdaki kayıta il adını değiştirmek istersek
1000 **Bursa** 2300
1000 **Istanbul** 2300
yeni il adı devam eden bilginin üzerine taşacaktır.
- Bu tür hataları önlemek için tüm kayıtlar sabit uzunlukta alınabilir.

Konular

- Giriş
- Data Hiyerarşisi
- Files (Dosyalar) ve Streams (Kaynaklar)
- Sıralı Dosya Oluşturma
- Sıralı Dosyadan Data Okuma
- Sıralı Dosyaları Güncelleme
- **Random-Access (Rastgele Erişimli) File**
- Rastgele Erişimli Dosya Oluşturma
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- Rastgele Erişimli Dosyadan Sıralı Okuma

Random-Access (Rastgele Erişimli) File

- Sıralı erişimli dosyalar belirli bir kayda hızlı ulaşım gereken uygulamalarda uygun değildir.
- Bir bilgiye doğrudan erişim Random Access File ile mümkündür.
- Random Access File bir bilgiye doğrudan erişir diğer kayıtlara bakmasına gerek yoktur.
- Bütün kayıtlar için aynı ve sabit boyut kullanmak arama işlemlerini kolaylaştırır.



Konular

- Giriş
- Data Hiyerarşisi
- Files (Dosyalar) ve Streams (Kaynaklar)
- Sıralı Dosya Oluşturma
- Sıralı Dosyadan Data Okuma
- Sıralı Dosyaları Güncelleme
- Random-Access (Rastgele Erişimli) File
- **Rastgele Erişimli Dosya Oluşturma**
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- Rastgele Erişimli Dosyadan Sıralı Okuma

Rastgele Erişimli Dosya Oluşturma

- `ostream` sınıfının `write` üyesi sabit sayıda byte'ı hafızada belirlenen yere yazar.
- Eğer bir dosyayla ilişkilendirilirse dosyadaki pozisyondan itibaren yazar.
- `istream` sınıfının `read` üyesi sabit sayıda byte'ı hafızadan belirlenen yerden okur.
- Eğer bir dosyayla ilişkilendirilirse dosyadaki pozisyondan itibaren okur.
- `outfile << number;` ile integer number dosyaya yazılır.
- Rastgele erişimli dosyalarda byte yazmak için `outfile.write(reinterpret_cast<const char*>(&number), sizeof(number));` kullanılır.

Rastgele Erişimli Dosya Oluşturma

```
1 // Fig. 17.10: ClientData.h
2 // Class ClientData definition used in Fig. 17.12-Fig. 17.15.
3 #ifndef CLIENTDATA_H
4 #define CLIENTDATA_H
5
6 #include <string>
7 using std::string;
8
9 class ClientData
10 {
11 public:
12     // default ClientData constructor
13     ClientData( int = 0, string = "", string = "", double = 0.0 );
14
15     // accessor functions for accountNumber
16     void setAccountNumber( int );
17     int getAccountNumber() const;
18
19     // accessor functions for lastName
20     void setLastName( string );
21     string getLastName() const;
22
23     // accessor functions for firstName
24     void setFirstName( string );
25     string getFirstName() const;
```

33

Rastgele Erişimli Dosya Oluşturma

```
26
27     // accessor functions for balance
28     void setBalance( double );
29     double getBalance() const;
30 private:
31     int accountNumber;
32     char lastName[ 15 ];
33     char firstName[ 10 ];
34     double balance;
35 }; // end class ClientData
36
37 #endif
38
```

34

Rastgele Erişimli Dosya Oluşturma

```
1 // Fig. 17.11: ClientData.cpp
2 // Class ClientData stores customer's credit information.
3 #include <string>
4 using std::string;
5
6 #include "ClientData.h"
7
8 // default ClientData constructor
9 ClientData::ClientData( int accountNumberValue,
10 string lastNameValue, string firstNameValue, double balanceValue )
11 {
12     setAccountNumber( accountNumberValue );
13     setLastName( lastNameValue );
14     setFirstName( firstNameValue );
15     setBalance( balanceValue );
16 } // end ClientData constructor
17
18 // get account-number value
19 int ClientData::getAccountNumber() const
20 {
21     return accountNumber;
22 } // end function getAccountNumber
23
```

35

Rastgele Erişimli Dosya Oluşturma

```
24 // set account-number value
25 void ClientData::setAccountNumber( int accountNumberValue )
26 {
27     accountNumber = accountNumberValue; // should validate
28 } // end function setAccountNumber
29
30 // get last-name value
31 string ClientData::getLastName() const
32 {
33     return lastName;
34 } // end function getLastName
35
36 // set last-name value
37 void ClientData::setLastName( string lastNameString )
38 {
39     // copy at most 15 characters from string to lastName
40     const char *lastNameValue = lastNameString.data();
41     int length = lastNameString.size();
42     length = ( length < 15 ? length : 14 );
43     strncpy( lastName, lastNameValue, length );
44     lastName[ length ] = '\0'; // append null character to lastName
45 } // end function setLastName
46
47 // get first-name value
48 string ClientData::getFirstName() const
49 {
50     return firstName;
51 } // end function getFirstName
52
```

36

Rastgele Erişimli Dosya Oluşturma

```
53 // set first-name value
54 void ClientData::setFirstName( string firstNameString )
55 {
56     // copy at most 10 characters from string to firstName
57     const char *firstNameValue = firstNameString.data();
58     int length = firstNameString.size();
59     length = ( length < 10 ? length : 9 );
60     strncpy( firstName, firstNameValue, length );
61     firstName[ length ] = '\0'; // append null character to firstName
62 } // end function setFirstName
63
64 // get balance value
65 double ClientData::getBalance() const
66 {
67     return balance;
68 } // end function getBalance
69
70 // set balance value
71 void ClientData::setBalance( double balanceValue )
72 {
73     balance = balanceValue;
74 } // end function setBalance
75
```

37

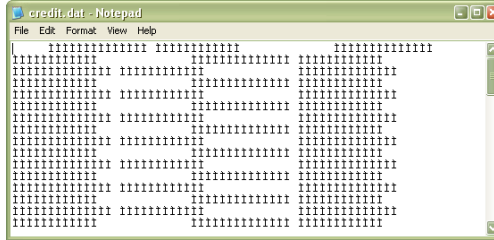
Rastgele Erişimli Dosya Oluşturma

```
1 // Fig. 17.12: Fig17_12.cpp
2 // Creating a randomly accessed file.
3 #include <iostream>
4 using std::cerr;
5 using std::endl;
6 using std::ios;
7
8 #include <fstream>
9 using std::ofstream;
10
11 #include <cstdlib>
12 using std::exit; // exit function prototype
13
14 #include "ClientData.h" // ClientData class definition
15
```

38

Rastgele Erişimli Dosya Oluşturma

```
16 int main()
17 {
18     ofstream outCredit( "credit.dat", ios::binary );
19
20     // exit program if ofstream could not open file
21     if ( !outCredit )
22     {
23         cerr << "File could not be opened." << endl;
24         exit( 1 );
25     } // end if
26
27     ClientData blankClient; // constructor zeros out each data member
28
29     // output 100 blank records to file
30     for ( int i = 0; i < 100; i++ )
31         outCredit.write( reinterpret_cast< const char * >( &blankClient ),
32                         sizeof( ClientData ) );
33
34     system("PAUSE");
35     return 0;
36 } // end main
```



39

Rastgele Erişimli Dosya Oluşturma

- Fig. 17.10 ve Fig.17.11 kayıt formatını belirler (**ClientData**) .
- Program 100 kaydı otomatik olarak oluşturup yazar.
- Program her kaydın sabit boyutta olmasını sağlar.
- Bunun için **setLastName** ve **setFirstName** fonksiyonları kullanılır.
- **ios::binary** ile açılan dosyanın binary formatta olduğu bildirilir.

40

Konular

- Giriş
- Data Hiyerarşisi
- Files (Dosyalar) ve Streams (Kaynaklar)
- Sıralı Dosya Oluşturma
- Sıralı Dosyadan Data Okuma
- Sıralı Dosyaları Güncelleme
- Random-Access (Rastgele Erişimli) File
- Rastgele Erişimli Dosya Oluşturma
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- Rastgele Erişimli Dosyadan Sıralı Okuma

Rastgele Erişimli Dosyaya Rastgele Veri Yazma

- `seekp` ile istenen yere konumlanılır ve `write` ile veri yazılabilir.

```
1 // Fig. 17.13: Fig17_13.cpp
2 // Writing to a random-access file.
3 #include <iostream>
4 using std::cerr;
5 using std::cin;
6 using std::cout;
7 using std::endl;
8 using std::ios;
9
10 #include <iomanip>
11 using std::setw;
12
13 #include <fstream>
14 using std::fstream;
15
16 #include <cstdlib>
17 using std::exit; // exit function prototype
18
19 #include "ClientData.h" // ClientData class definition
20
```

Rastgele Erişimli Dosyaya Rastgele Veri Yazma

```
21 int main()
22 {
23     int accountNumber;
24     char lastName[ 15 ];
25     char firstName[ 10 ];
26     double balance;
27
28     ofstream outCredit( "c:\\\\predit.dat", ios::in | ios::out | ios::binary );
29
30     // exit program if ofstream cannot open file
31     if ( !outCredit )
32     {
33         cerr << "File could not be opened." << endl;
34         exit( 1 );
35     } // end if
36
37     cout << "Enter account number (1 to 100, 0 to end input)\n? ";
38
39     // require user to specify account number
40     ClientData client;
41     cin >> accountNumber;
42
```

43

Rastgele Erişimli Dosyaya Rastgele Veri Yazma

```
43     // user enters information, which is copied into file
44     while ( accountNumber > 0 && accountNumber <= 100 )
45     {
46         // user enters last name, first name and balance
47         cout << "Enter lastname, firstname, balance\n? ";
48         cin >> setw( 15 ) >> lastName;
49         cin >> setw( 10 ) >> firstName;
50         cin >> balance;
51
52         // set record accountNumber, lastName, firstName and balance values
53         client.setAccountNumber( accountNumber );
54         client.setLastName( lastName );
55         client.setFirstName( firstName );
56         client.setBalance( balance );
57
58         // seek position in file of user-specified record
59         outCredit.seekp( ( client.getAccountNumber() - 1 ) *
60             sizeof( ClientData ) );
61
62         // write user-specified information in file
63         outCredit.write( reinterpret_cast< const char * >( &client ),
64             sizeof( ClientData ) );
65
66         // enable user to enter another account
67         cout << "Enter account number\n? ";
68         cin >> accountNumber;
69     } // end while
70
71     system("PAUSE");
72     return 0;
73 } // end main
```

44

Konular

- Giriş
- Data Hiyerarşisi
- Files (Dosyalar) ve Streams (Kaynaklar)
- Sıralı Dosya Oluşturma
- Sıralı Dosyadan Data Okuma
- Sıralı Dosyaları Güncelleme
- Random-Access (Rastgele Erişimli) File
- Rastgele Erişimli Dosya Oluşturma
- Rastgele Erişimli Dosyaya Rastgele Veri Yazma
- **Rastgele Erişimli Dosyadan Sıralı Okuma**

Rastgele Erişimli Dosyadan Sıralı Okuma

- `istream` sınıfındaki `read` fonksiyonu belirlenen kaynakta bulunan pozisyondan belirlenen sayıda byte okur.
- Eğer kaynak bir dosya ise konumlanılan yerden belirlenen boyutta veri okunur.

Rastgele Erişimli Dosyadan Sıralı Okuma

```
1 // Fig. 17.14: Fig17_14.cpp
2 // Reading a random access file sequentially.
3 #include <iostream>
4 using std::cerr;
5 using std::cout;
6 using std::endl;
7 using std::fixed;
8 using std::ios;
9 using std::left;
10 using std::right;
11 using std::showpoint;
12
13 #include <iomanip>
14 using std::setprecision;
15 using std::setw;
16
17 #include <fstream>
18 using std::ifstream;
19 using std::ofstream;
20
21 #include <cstdlib>
22 using std::exit; // exit function prototype
23
24 #include "ClientData.h" // ClientData class definition
25
26 void outputLine( ostream&, const ClientData & ); // prototype
27
```

49

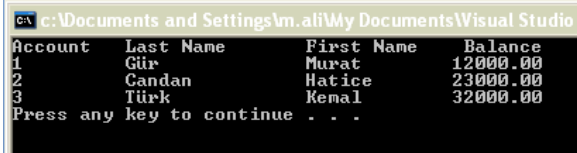
Rastgele Erişimli Dosyadan Sıralı Okuma

```
28 int main()
29 {
30     ifstream inCredit( "c:\\credit.dat", ios::in );
31
32     // exit program if ifstream cannot open file
33     if ( !inCredit )
34     {
35         cerr << "File could not be opened." << endl;
36         exit( 1 );
37     } // end if
38
39     cout << left << setw( 10 ) << "Account" << setw( 16 )
40          << "Last Name" << setw( 11 ) << "First Name" << left
41          << setw( 10 ) << right << "Balance" << endl;
42
43     ClientData client; // create record
44
45     // read first record from file
46     inCredit.read( reinterpret_cast< char * >( &client ),
47                  sizeof( ClientData ) );
48
49     // read all records from file
50     while ( inCredit && !inCredit.eof() )
51     {
52         // display record
53         if ( client.getAccountNumber() != 0 )
54             outputLine( cout, client );
55
56         // read next from file
57         inCredit.read( reinterpret_cast< char * >( &client ),
58                       sizeof( ClientData ) );
59     } // end while
60
61     system("PAUSE");
62     return 0;
63 } // end main
```

50

Rastgele Erişimli Dosyadan Sıralı Okuma

```
65 // display single record
66 void outputLine( ostream &output, const ClientData &record )
67 {
68     output << left << setw( 10 ) << record.getAccountNumber()
69         << setw( 16 ) << record.getLastName()
70         << setw( 11 ) << record.getFirstName()
71         << setw( 10 ) << setprecision( 2 ) << right << fixed
72         << showpoint << record.getBalance() << endl;
73 } // end function outputLine
74
```



The screenshot shows the output of the program in a console window. The output is a table with four columns: Account, Last Name, First Name, and Balance. The data rows are: 1 Gür Murat 12000.00, 2 Candan Hatice 23000.00, and 3 Türk Kemal 32000.00. Below the table, it says 'Press any key to continue . . .'

Account	Last Name	First Name	Balance
1	Gür	Murat	12000.00
2	Candan	Hatice	23000.00
3	Türk	Kemal	32000.00

51

Rastgele Erişimli Dosyadan Sıralı Okuma

- Programda satır 57-58, `sizeof(ClientData)` boyutunda okuma yapar.
- Satır 50'deki `!incredit.eof()` ifadesi dosya sonunu kontrol eder.

52