Дәріс 14. Ерекшеліктерді өңдеу және файлдармен жұмыс (C++)

1. Ерекшеліктер (Exceptions) дегеніміз не?

Ерекшелік (exception) – бағдарлама орындалу кезінде пайда болатын қате немесе күтпеген жағдай.
Мысалы:

- Нөлге бөлу
- Жад жетіспеуі
- Файл табылмауы
- Индекстің шектен шығуы

Егер мұндай жағдайлар өңделмесе – бағдарлама тоқтап қалады
C++ тілінде ерекшеліктерді өңдеу үшін try, throw, catch қолданылады.

2. try / throw / catch құрылымы

Негізгі синтаксис:

```
try {
   // Қате пайда болуы мүмкін код
   throw <мән>;   // ерекшелік тастау
}
catch (<тип> e) {
   // ерекшелікті өңдеу
}
```

Мысал:

```
#include <iostream>
using namespace std;

int main() {
   try {
      int a = 5, b = 0;
      if (b == 0)
         throw "Нөлге бөлуге болмайды!";
      cout << a / b;
   }
   catch (const char* msg) {
      cout << "Қате: " << msg << endl;
   }
   return 0;
}
```

Нәтиже:

Қате: Нөлге бөлуге болмайды!

3. Бірнеше catch блоктарын қолдану

Әртүрлі типтегі ерекшеліктерді бөлек өңдеуге болады:

```
try {
   throw 404;
}
catch (int e) {
   cout << "Қате коды: " << e << endl;
}
catch (...) {
   cout << "Белгісіз қате!" << endl;
}
```

catch (...) – кез келген типтегі ерекшелікті ұстайды (жалпы өңдегіш).

4. Өз ерекшеліктеріңді жасау (custom exceptions)

Бағдарламашы өз класы арқылы ерекшелік жасай алады:

```
#include <iostream>
#include <stdexcept>
using namespace std;

class MyException : public exception {
public:
   const char* what() const noexcept override {
      return "Менің өз ерекшелігім орын алды!";
   }
};

int main() {
   try {
      throw MyException();
   }
   catch (const exception& e) {
      cout << "Қате: " << e.what() << endl;
   }
}
```

Нәтиже:

Қате: Менің өз ерекшелігім орын алды!

5. Файлдармен жұмыс негіздері

C++ файлдармен жұмыс істеу үшін **ағындар (streams)** қолданады:

Класс	Мақсаты
ifstream	Файлдан оқу (input file stream)
ofstream	Файлға жазу (output file stream)
fstream	Екі жақты (оқу және жазу)

Файлдармен жұмыс жасау үшін <fstream> кітапханасын қосамыз.

6. Файлға жазу (ofstream)

```
#include <iostream>
#include <fstream>
using namespace std;

int main() {
   ofstream file("data.txt"); // файл ашу

   if (!file) {
      cout << "Файл ашылмады!" << endl;
      return 1;
   }

   file << "Сәлем, әлем!" << endl;
   file << "Бұл файлға жазылған мәтін." << endl;

   file.close(); // файлды жабу
   cout << "Мәлімет файлға жазылды." << endl;

   return 0;
}
```

data.txt файлына мәтін жазылады.

7. Файлдан оқу (ifstream)

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main() {
   ifstream file("data.txt");

   if (!file) {
      cout << "Файл табылмады!" << endl;
```

```
        return 1;
    }

    string line;
    while (getline(file, line)) {
        cout << line << endl;
    }

    file.close();
    return 0;
}
```

Файлдан жол-жолмен оқиды.

8. fstream – оқу және жазу бір мезетте

```
#include <iostream>
#include <fstream>
using namespace std;

int main() {
    fstream file("numbers.txt", ios::in | ios::out | ios::trunc);

    if (!file) {
        cout << "Файл ашылмады!" << endl;
        return 1;
    }

    // Жазу
    file << "12345\n";

    // Курсорды басына жылжыту
    file.seekg(0);

    // Оқу
    string data;
    file >> data;
    cout << "Файлдан оқылды: " << data << endl;

    file.close();
    return 0;
}
```

ios::in – оқу, ios::out – жазу, ios::trunc – файлды тазалап жазу.

Файл ашу режимдері (file open modes)

Түйін	Мағынасы
ios::in	Оқу үшін ашу
ios::out	Жазу үшін ашу
ios::app	Соңына қосу
ios::trunc	Файлды тазалау
ios::binary	Екілік режим

Қысқаша қорытынды

Түсінік	Мағынасы
try / catch	Ерекшеліктерді өңдеу механизмі
throw	Ерекшелік тастау
custom exception	Өз ерекшелігіңді жасау
ifstream / ofstream / fstream	Файлмен жұмыс істеуге арналған ағындар
ios::in / ios::out	Файл ашу режимдері

Артықшылықтары:

- Қателерді қауіпсіз өңдейді
- Бағдарламаның тұрақтылығын арттырады
- Файл арқылы тұрақты деректермен жұмыс жасауға мүмкіндік береді
- Деректерді сақтау және қайта оқу жеңілдейді

Есте сақтау:

- Әрқашан файл ашылғанын тексер! (if (!file))
- Файлды қолданған соң close() әдісімен жап.
- Ерекшеліктерді өңдемесең – бағдарлама тоқтайды.