throw-dan sonraki hec bir kod bloku islemir

Stack Unwiding (Exception)

exception atilan yerden(throwdan) tutulan yere(catcha) kecdikde steakde throwdan

evvel yaranan elementleri (Stack Unwiding) ozu silir

#include <iostream>

using namespace std;

class Exception

{

protected:

string text;

string source; // file

int line;

string date;

public:

Exception(string text,string source, int line, string date)

:text(text),source(source),line(line),date(date)

{

}

virtual void PrintMessage() const

{

cout << " ========= ERROR INFO ========= " << endl;

cout << " Error Content : " << text << endl;

cout << " Source : " << source << endl;

cout << " Line Number : " << line << endl;

cout << " Date : " << date << endl;

}

};

class OutOfRangeException :public Exception

{

public:

OutOfRangeException(string text, string source, int line, string date)

:Exception(text, source, line, date)

{

}

void PrintMessage() const override

{

cout << " Error type : " << "OutOfRangeException" << endl;

Exception::PrintMessage();

}

};

class InvalidArgumentException : public Exception

{

public:

InvalidArgumentException(string text, string source, int line, string date)

:Exception(text, source, line, date)

{

}

void PrintMessage() const override

{

cout << " Error type : " << "InvalidArgumentException" << endl;

Exception::PrintMessage();

}

};

template<typename T>

class Array

{

private:

T\* data;

int size;

public:

int GetSize() const

{

return size;

}

Array<T>& operator=(const Array<T>& other) = delete;

Array(const Array<T>& other) = delete; // kimsenin copy assignment (basqa seyler de ola biler) etmesine icaze vermir

Array(int size) throw(InvalidArgumentException) // Bu kod blokunda cixa bilecek exceptionlari gosterir

{

if (size < 0 || size > 1000000)

{

throw new InvalidArgumentException("Size should be between 0 and 1000000",

\_\_FILE\_\_, \_\_LINE\_\_, \_\_DATE\_\_);

}

this->size = size;

data = new T[size]{};

}

T& operator[](int index) throw(OutOfRangeException) // Bu kod blokunda cixa bilecek exceptionlari gosterir

{

if (index < 0 || index >= size)

{

throw new OutOfRangeException("Index should be more than 0 and less than " << size,

\_\_FILE\_\_, \_\_LINE\_\_, \_\_DATE\_\_);

}

return data[index];

}

};

int main()

{

try

{

Array<int> a(-10);

}

catch (const Exception \*ex)

{

ex->PrintMessage();

}

return 0;

}