

```
// W_16_0.cpp : Defines the entry point for the console application.  
// Error & message system based on exceptions CPP
```

```
#include "stdafx.h"  
#include <iostream>  
#include <fstream>
```

```
using namespace std;
```

```
class CExcept
```

```
{
```

```
public:
```

```
enum MY_MESSAGES
```

```
{
```

```
MY_EXCPT_ALLOC_MEM,
```

```
MY_EXCPT_OPEN_FILE,
```

```
MY_EXCPT_WARN_1,
```

```
MY_EXCEPT_TOT_NUMB
```

```
}my_mess;
```

```
private:
```

```
int pos; //wskazuje do numeru wierszy tekstowego w tabeli  
//tab_err lub tab_warn
```

```
static char *tab_mess[MY_EXCEPT_TOT_NUMB];
```

```
public:
```

```
CExcept::CExcept(enum MY_MESSAGES my_mess);
```

```
void fatal_error();
```

```
void send_mess();
```

```
};
```

```
CExcept::CExcept(enum MY_MESSAGES my_mess)
```

```
/*=====
```

```
Konstruktor klasy
```

```
=====*/
```

```
{
```

```
pos = my_mess;
```

```
if(tab_mess[my_mess][0] == 'E')
```

```
{
```

```
//błąd fatalny
```

```
throw *this;
```

```
}
```

```
else
```

```
{
```

```

        //komunikat
        send_mess();
    }
}

void CExcept::fatal_error()
/*=====
Obsługujemy błąd fatalny
=====*/
{
    cout << tab_mess[pos] << endl;
    system("pause");
    exit(1);
}

void CExcept::send_mess()
/*=====
Obsługujemy komunikat
=====*/
{
    cout << tab_mess[pos] << endl;
}

//Inicjowanie statycznych składowych klasy jako zmiennych globalnych
//Pierwszy symbol jest symbolem sterującym: E - error (błąd fatalny -
//wykonanie programu należy przerwać), W - warning (komunikat albo błąd
//nie fatalny - program wykonuje następne instrukcje kodu)
char *CExcept::tab_mess[] =
{
    "W Memory allocation error",      //MY_EXCPT_ALLOC_MEM
    "E Open file error",              //MY_EXCPT_OPEN_FILE
    "W Warning foo"                   //MY_EXCPT_WARN_1
};

int _tmain(int argc, _TCHAR* argv[])
{
    int n = 100000;
    double *ptr = NULL;
    ifstream file;
    try
    {
        //wysyłamy komunikat na monitor - skoku do bloku catch

```

```

        //nie wykonujemy
        CExcept ob_mess(CExcept::MY_EXCPT_WARN_1);

        //allocation of memory
        try
        {
            ptr = new double [n*n*n];
        }
        catch(bad_alloc)
        {
            //traktujemy to jako błąd nie fatalny - skoku do bloku catch
            //nie wykonujemy
            CExcept ob(CExcept::MY_EXCPT_ALLOC_MEM);
        }

        if(ptr)
            delete [] ptr;
        ptr = NULL;

        //próbujemy otworzyć plik, którego nie istnieje
        file.open("my_file.txt", ios_base::in);
        if(!file.is_open()) //sprawdzamy, czy plik jest otwarty
            CExcept ob(CExcept::MY_EXCPT_OPEN_FILE); //generujemy wyjątek - błąd fatalny
        else
            file.close();
    }
    catch(CExcept a)
    {
        a.fatal_error();
    }

    system("pause");
    return 0;
}

```