

```
// W1_2.cpp : Defines the entry point for the console application.
//          Timer: Wywołanie konstruktorów i destruktorów
//          przy lokalnej, globalnej i lokalnej-static deklaracji obiektów

#include "stdafx.h"
#include "windows.h" // dla funkcji Sleep - platforma SDK
#include <iostream>
#include <ctime> //nowy styl nagłówek
using namespace std;

class my_timer
{
    clock_t start;
    char str[256];
    char str_id[256];
public:
    my_timer(char *str1); //konstruktor
    ~my_timer(); //destruktor
};

my_timer::my_timer(char *str1)
{
    //konstruktor: wykonuje pierwsze pobranie czasu
    start = clock();

    if(strcpy_s(str_id, sizeof(str_id), str1))
    {
        cout << str1 << endl;
        cout << "zbyt długi wiersz tekstowy" << endl;
        system("pause");
        exit(1);
    }
    cout << "tworzenie obiektu " << str_id << endl;
}

my_timer::~my_timer()
{
    //destruktor: wykonuje drugie pobranie czasu, liczy elapsed time
    clock_t end = clock();
    double duration = (double)(end-start)/CLOCKS_PER_SEC;
    sprintf(str, "niszczenie obiektu %s: trwalosc: %lf s", str_id, duration);
    cout << str << endl;
    MessageBoxA(NULL, str, "KOMUNIKAT", MB_OK);
}

//deklaracja globalna
my_timer t1("t1 deklaracja globalna");

int _tmain(int argc, _TCHAR* argv[])
{
    int count;
    cout << "Start main" << endl << endl;

    for(count=0; count<3; count++)
    {
        cout << "-----\n";
        cout << "icount = " << count << endl;
        //deklaracja lokalna
        my_timer t2("t2 - deklaracja lokalna");

        //deklaracja lokalna, static
        static my_timer t3("t3 - deklaracja lokalna static");

        //Zatrzymaj wykonanie o 1 sekunde
        Sleep(1000);
    }

    cout << endl << "End main" << endl;
    return 0;
}
```