

```

// W3.cpp : Defines the entry point for the console application.
//      Funkcje zwraca obiekt klasy

#include "stdafx.h"
#include "windows.h"
#include <iostream>
#include <cstdlib>
using namespace std;

#define BAD_VALUE 0xFFFFFFFF

//Zakomentuj nastepna linie kodu, zeby zrealizowac
//inny sposob wykonania programu
//#define EXECUTION_CASE_1

class myclass;           // referencje do klasy myclass przed deklaracja klasy
myclass fun(int ndim);

class myclass
{
    double *ptr;
    int it;
public:
    myclass(int items, char *title);
    ~myclass();
    void put(double a);
    int getnoitems();
    double get(int i);
    void disp(char *title);
};

myclass::myclass(int items, char *title)
{
    ptr = (double *)malloc(items*sizeof(double));
    if(!ptr)
    {
        cout << "blad alokacji pamieci\n";
    }

    it = 0;
    memset((void *)ptr, 0, items*sizeof(double));
    cout << "Konstruktor " << title << "   adres   " << ptr << "\n";
}

```

```

}

myclass::~myclass()
{
    cout << "wywołanie destruktora: ";

    if(ptr)
    {
        cout << "adres " << ptr;
        Sleep(3000);
        free(ptr);
        ptr = NULL;
        cout << " free ptr";
        it = 0;
    }

    cout << "\n";
    MessageBox(NULL, L" ", L"DESTRUKTOR", MB_OK);
}

int myclass::getnoitems()
{
    if(!ptr)
        return -1;

    size_t noitems = _msize(ptr); //rozmiar tablicy
    return (int)noitems/sizeof(double);
}

void myclass::put(double a)
{
    if(ptr)
    {
        int noitems = getnoitems(); //rozmiar tablicy
        if(it >= noitems)
        {
            cout << "tablica jest wypelniona\n";
            return;
        }

        ptr[it] = a;
        it++;
    }
}

```

```

    }
}

double myclass::get(int i)
{
    if(ptr)
    {
        int noitems = getnoitems();
        if(i < 0 || i >= noitems)
            return BAD_VALUE;

        return ptr[i];
    }

    return BAD_VALUE;
}

void myclass::disp(char *title)
{
    cout << title << "\n";
    int noitems = getnoitems();
    int i;
    for(i=0; i<noitems; i++)
        cout << "i = " << i << "\t" << ptr[i] << "\n";
}

int _tmain(int argc, _TCHAR* argv[])
{
#ifdef EXECUTION_CASE_1
    myclass ob = fun(10);    //przy wyjściu z fun będzie wywołany destruktor dla zwolnienia
#else                       //obiektu ret, tworzonego w funkcji fun dla zwracania obiektu.
    myclass ob(10, "main");
    ob = fun(10);
#endif

    ob.disp("main");        // - PAGE FAULT!

    return 0;
}

myclass fun(int ndim)

```

```
{
    myclass ret(ndim, "fun");
    int it;
    for(it=0; it<ndim; it++)
    {
        ret.put(10.0*(it+1));
    }
    ret.disp("fun");
    return ret;
    //tu bedzie wywolany destruktork dla zwolnienia obiektu ret
}
```

```
#undef EXECUTION_CASE_1
```