

```
// W8_1.cpp : Defines the entry point for the console application.  
//          Przeciążenie operatora []. Przykład tablicy bezpiecznej
```

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

```
class s_array  
{  
    double *arr;  
    int w_size;  
public:  
    s_array(int ndim);  
    ~s_array();  
    double &operator[](int i);  
    int getsize() { return w_size; }  
    void put(double a, int i);  
    double get(int i);  
};
```

```
s_array::s_array(int ndim)  
{  
    w_size = ndim;  
    try  
    {  
        arr = new double [w_size];  
    }  
    catch(bad_alloc aa)  
    {  
        cout << "s_array: Memory allocation error\n";  
        system("pause");  
        exit(1);  
    }  
    memset((void *)arr, 0, w_size*sizeof(double));  
}
```

```
s_array::~s_array()  
{  
    if(arr)  
    {  
        delete [] arr;
```

```

        arr = NULL;
    }
}

double & s_array::operator [] (int i)
/*=====
Przeciazenie operatora []
zwraca referencje na typ double - mozna uzywac jako L-value
i - indeks tablicy
=====*/
{
    if(!arr || i < 0 || i >= w_size)
    {
        cout << "bledny wskaznik do tablicy lub indeks przekrocza granice tablicy\n";
        system("pause");
        exit(1);
    }

    return arr[i];
}

void s_array::put(double a, int i)
{
    if(!arr || i < 0 || i >= w_size)
    {
        cout << "bledny wskaznik do tablicy lub indeks przekrocza granice tablicy\n";
        system("pause");
        exit(1);
    }

    arr[i] = a;
}

double s_array::get(int i)
{
    if(!arr || i < 0 || i >= w_size)
    {
        cout << "bledny wskaznik do tablicy lub indeks przekrocza granice tablicy\n";
        system("pause");
        exit(1);
    }
}

```

```

        return arr[i];
    }

int _tmain(int argc, _TCHAR* argv[])
{
    int ii;
    s_array aa(10);

    //teraz mozna odwolowywac do obiektu klasy s_array tak, jako do zwykłej tablicy
    aa[0] = 100.0;    //tu [] jest uzyty jako L-value
    aa[1] = aa[0]+100.0;

    for(ii=2; ii<aa.getsize(); ii++)
        aa[ii] = 100.0*(ii+1);

    for(ii=0; ii<aa.getsize(); ii++)
    {
        cout << aa[ii] << "\n";
    }
    cout << "\n";

    //To samo, wariant alternatywny
    aa.put(100.0, 0);    //arr[0] <--100
    aa.put(aa.get(0)+100.0, 1);

    for(ii=2; ii<aa.getsize(); ii++)
        aa.put(100.0*(ii+1), ii);

    for(ii=0; ii<aa.getsize(); ii++)
    {
        cout << aa.get(ii) << "\n";
    }
    cout << "\n";

    //Który wariant jest bardziej elegancki i czytelny?

    //Generujemy blad
    cout << aa[11];

    return 0;
}

```

