

```

// W1.cpp : Defines the entry point for the console application.
//          Klasa Stack obsluga steku (LIFO). Straze pliku naglowkowego
//          Dzialania na steku.

#include "stdafx.h"
#include <iostream>
#include "mystack.h"
#include "mystack.h"

using namespace std;

int get_numb_obj();

//definicje funkcji-metod klasy CStack
void CStack::init()
{
    noitems = 0;
}

void CStack::push(double mem)
{
    if(noitems >= MAX_SIZE)
    {
        cout << "stack jest wypelniony\n";
        return;
    }

    arr_stack[noitems] = mem;
    noitems++;
}

double CStack::pop()
{
    if(noitems <= 0)
    {
        cout << "stack jest pusty\n";
        return 0;
    }

    noitems--;
    return arr_stack[noitems];
}

```

```

}

int CStack::getnumb()
{
    return noitems;
}

int _tmain(int argc, _TCHAR* argv[])
{
    int op;
    int item;
    double var;
    CStack st[2]; //tablica obiektow typu CStack

    st[0].init();
    st[1].init();

    while(1)
    {
        cout << "kod dzialania? (0 - push, 1 - pop, 2 - ilosc elementow, 3 - skoncz\n";
        cin >> op;

        switch(op)
        {
            case 0: //push
                item = get_numb_obj(); //ktory obiekt (stek) - 0 lub 1 ?
                if(item >= 0)
                {
                    cout << "wprowadz wartosc\n";
                    cin >> var;
                    st[item].push(var);
                }
                break;

            case 1: //pop
                item = get_numb_obj();
                if(item >= 0)
                {
                    var = st[item].pop();
                }
            }
        }
    }
}

```

```

        cout << "obiekt " << item << "   wartosc var = " << var << "\n";
    }
    break;
case 2: //ilosc elementow
    item = get_numb_obj();
    if(item >= 0)
    {
        cout << "obiekt " << item << "   pozostalo " << st[item].getnumb() << "   elementow\n";
    }
    break;
case 3: //exit
    exit(0);
    break;
default:
    cout << "bledne dzialanie\n";
};
} //end while

return 0;
}

int get_numb_obj()
/*=====
Zwraca numer obiektu lub -1
=====*/
{
    int it;

    cout << "numer obiektu? (0 lub 1)\n";
    cin >> it;
    if(it > 1 || it < 0)
    {
        cout << "bledny numer obiektu\n";
        return -1;
    }

    return it;
}

```