```cpp
#ifdef _WIN32
    #pragma warning(disable:4996)
    #include <tchar.h>
    #include <windows.h>
    #include <conio.h>
    #include <direct.h>
#elif (defined __linux__) || (defined _AIX) || (defined __APPLE__)
    #include <stdlib.h>
    #include <sys/types.h>
    #include <sys/stat.h>
    #include <unistd.h>
    typedef char _TCHAR;
    #define _tmain main
#endif

#include <stdio.h>
#include <iostream>
#include <iomanip>
using namespace std;

#include "MaticaObdlznikova.h"
#include "MaticaStvorcova.h"
//-----------------------------------------------------------------------
TMaticaStvorcova::TMaticaStvorcova()
{
}
//-----------------------------------------------------------------------
TMaticaStvorcova::TMaticaStvorcova(TMaticaStvorcova& X)
{
    PocetRiadkov = X.PocetRiadkov;
    PocetStlpcov = X.PocetStlpcov;
    for (unsigned i = 0; i<PocetRiadkov; i++)
        for (unsigned j = 0; j<PocetStlpcov; j++)
            Matica[i][j] = X.Matica[i][j];
}
//-----------------------------------------------------------------------
TMaticaStvorcova::~TMaticaStvorcova()
{
}
//-----------------------------------------------------------------------
TMaticaStvorcova& TMaticaStvorcova::operator=(const TMaticaStvorcova& X)
{
    if (this == &X) return *this;

    PocetRiadkov = X.PocetRiadkov;
    PocetStlpcov = X.PocetStlpcov;
    for (unsigned i = 0; i<PocetRiadkov; i++)
        for (unsigned j = 0; j<PocetStlpcov; j++)
            Matica[i][j] = X.Matica[i][j];

    return *this;
}
//-----------------------------------------------------------------------
TMaticaStvorcova& TMaticaStvorcova::operator=(const TMaticaObdlznikova& X)
{
    if (this == &X) return *this;

    my_class xx;

    if (X.getPocetRiadkov() != X.getPocetStlpcov()) {
        cout << "\nTato obdlznikova matica sa neda previest na stvorcovu!";
        xx.my_getch();
        exit(1);
    }

    PocetRiadkov = X.getPocetRiadkov();
    PocetStlpcov = X.getPocetStlpcov();
    for (unsigned i = 0; i<PocetRiadkov; i++)
        for (unsigned j = 0; j<PocetStlpcov; j++)
            Matica[i][j] = X.getPrvokMatice(i, j);

    return *this;
}
```

```cpp
//---------------------------------------------------------------------------
TMaticaStvorcova TMaticaStvorcova::UmocniMaticu(unsigned k)
{
    TMaticaStvorcova Y, PomMatica0, PomMatica;
    my_class x;

    if (PocetStlpcov != PocetRiadkov) {
        cout << "\n  Matica sa neda umocnit!";
        x.my_getch();
        exit(1);
    }
    Y.PocetRiadkov = Y.PocetStlpcov = PocetRiadkov;

    switch (k) {
    case 0:
        for (unsigned i = 0; i<PocetRiadkov; i++) {
            for (unsigned j = 0; j<PocetRiadkov; j++)
                Y.Matica[i][j] = 0;
            Y.Matica[i][i] = 1;
        }
        break;
    case 1:
        Y = *this;
        break;
    default:
        PomMatica0 = PomMatica = *this;
        k--;
        while (k--) {
            Y = PomMatica * PomMatica0;
            PomMatica = Y;
        }
    }

    return Y;
}
//---------------------------------------------------------------------------
```