```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"
//--------------------------------------------------------------------------
//Konstruktor inicializuje prvky matice
TMaticaObdlznikova::TMaticaObdlznikova()
{
    PocetRiadkov = MAXPOCET;
    PocetStlpcov = MAXPOCET;
    for (unsigned i = 0; i<PocetRiadkov; i++)
        for (unsigned j = 0; j<PocetStlpcov; j++)
            Matica[i][j] = 0;
}
//--------------------------------------------------------------------------
TMaticaObdlznikova::TMaticaObdlznikova(TMaticaObdlznikova& 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];
}
//--------------------------------------------------------------------------
TMaticaObdlznikova::~TMaticaObdlznikova()
{
}
//--------------------------------------------------------------------------
TMaticaObdlznikova& TMaticaObdlznikova::operator=(const TMaticaObdlznikova& 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];

    return *this;
}
//--------------------------------------------------------------------------
const TMaticaObdlznikova operator+(const TMaticaObdlznikova& LavaMatica,
    const TMaticaObdlznikova& PravaMatica)
{
    TMaticaObdlznikova VyslMatica;

    VyslMatica.PocetRiadkov = LavaMatica.PocetRiadkov;
    VyslMatica.PocetStlpcov = LavaMatica.PocetStlpcov;
    for (unsigned i = 0; i<LavaMatica.PocetRiadkov; i++)
        for (unsigned j = 0; j<LavaMatica.PocetStlpcov; j++)
            VyslMatica.Matica[i][j] = LavaMatica.Matica[i][j]
            + PravaMatica.Matica[i][j];
    return VyslMatica;
}
//--------------------------------------------------------------------------
const TMaticaObdlznikova operator*(const TMaticaObdlznikova& LavaMatica,
    const TMaticaObdlznikova& PravaMatica)
{
```

```cpp
 74          TMaticaObdlznikova VyslMatica;
 75          my_class xx;
 76          unsigned i, j, k;
 77          float Suma;
 78
 79          if (LavaMatica.PocetStlpcov != PravaMatica.PocetRiadkov) {
 80              cout << "\n  Matice sa nedaju nasobit!\n";
 81              xx.my_getch();
 82              exit(1);
 83          }
 84          VyslMatica.PocetRiadkov = LavaMatica.PocetRiadkov;
 85          VyslMatica.PocetStlpcov = PravaMatica.PocetStlpcov;
 86          for (i = 0; i<LavaMatica.PocetRiadkov; i++)
 87              for (j = 0; j<PravaMatica.PocetStlpcov; j++) {
 88                  Suma = 0;
 89                  for (k = 0; k<LavaMatica.PocetStlpcov; k++)
 90                      Suma += LavaMatica.Matica[i][k] * PravaMatica.Matica[k][j];
 91                  VyslMatica.Matica[i][j] = Suma;
 92              }
 93          return VyslMatica;
 94      }
 95      //-------------------------------------------------------------------------
 96      istream& operator>>(istream& is, TMaticaObdlznikova& X)
 97      {
 98          my_class xx;
 99
100          is >> X.PocetRiadkov >> X.PocetStlpcov;
101          if (is.fail()) {
102              cout << "\n\n  Vstupny subor sa nepodarilo otvorit.\n";
103              xx.my_getch();
104              exit(1);
105          }
106          for (unsigned i = 0; i<X.PocetRiadkov; i++)
107              for (unsigned j = 0; j<X.PocetStlpcov; j++)
108                  is >> X.Matica[i][j];
109          return is;
110      }
111      //-------------------------------------------------------------------------
112      ostream& operator<<(ostream& os, TMaticaObdlznikova& X)
113      {
114          os.setf(ios::fixed, ios::floatfield);
115          os.precision(2);
116          for (unsigned i = 0; i<X.PocetRiadkov; i++) {
117              os << "\n  ";
118              for (unsigned j = 0; j<X.PocetStlpcov; j++)
119                  os << setw(8) << X.Matica[i][j];
120          }
121          os << "\n";
122          return os;
123      }
124      //-------------------------------------------------------------------------
125      void my_class::my_getch() const
126      {
127      #ifdef _WIN32
128          _getch();
129      #else
130          cout << endl;
131      #endif
132      }
133      //-------------------------------------------------------------------------
134
```