

```

1  #ifdef _WIN32
2      #pragma warning(disable:4996)
3      #include <tchar.h>
4      #include <windows.h>
5      #include <conio.h>
6      #include <direct.h>
7  #elif (defined __linux__ ) || (defined _AIX) || (defined __APPLE__)
8      #include <stdlib.h>
9      #include <sys/types.h>
10     #include <sys/stat.h>
11     #include <unistd.h>
12     typedef char _TCHAR;
13     #define _tmain main
14 #endif
15
16 #include <stdio.h>
17 #include <string.h>
18 #include <iostream>
19 #include <iomanip>
20 using namespace std;
21
22 #include "Matica.h"
23 //-----
24 //Konstruktor inicializuje prvky matice
25 TMatica::TMatica()
26 {
27     PocetRiadkov = MAXPOCET;
28     PocetStlpcov = MAXPOCET;
29     for (unsigned i = 0; i<PocetRiadkov; i++)
30         for (unsigned j = 0; j<PocetStlpcov; j++)
31             Matica[i][j] = 0;
32 }
33 //-----
34 TMatica::~TMatica()
35 {
36 }
37 //-----
38 const TMatica operator+(const TMatica& LavaMatica, const TMatica& PravaMatica)
39 {
40     TMatica VyslMatica;
41
42     VyslMatica.PocetRiadkov = LavaMatica.PocetRiadkov;
43     VyslMatica.PocetStlpcov = LavaMatica.PocetStlpcov;
44     for (unsigned i = 0; i<LavaMatica.PocetRiadkov; i++)
45         for (unsigned j = 0; j<LavaMatica.PocetStlpcov; j++)
46             VyslMatica.Matica[i][j] = LavaMatica.Matica[i][j]
47             + PravaMatica.Matica[i][j];
48     return VyslMatica;
49 }
50 //-----
51 const TMatica operator*(const TMatica& LavaMatica, const TMatica& PravaMatica)
52 {
53     TMatica VyslMatica;
54     my_class xx;
55     unsigned i, j, k;
56     float Suma;
57
58     if (LavaMatica.PocetStlpcov != PravaMatica.PocetRiadkov) {
59         cout << "\n Matice sa nedaju nasobit!";
60         xx.my_getch();
61     }
62     VyslMatica.PocetRiadkov = LavaMatica.PocetRiadkov;
63     VyslMatica.PocetStlpcov = PravaMatica.PocetStlpcov;
64     for (i = 0; i<LavaMatica.PocetRiadkov; i++)
65         for (j = 0; j<PravaMatica.PocetStlpcov; j++) {
66             Suma = 0;
67             for (k = 0; k<LavaMatica.PocetStlpcov; k++)
68                 Suma += LavaMatica.Matica[i][k] * PravaMatica.Matica[k][j];
69             VyslMatica.Matica[i][j] = Suma;
70         }
71     return VyslMatica;
72 }
73 //-----

```

```

74  TMatica TMatica::UmocniMaticu(unsigned k)
75  {
76      TMatica Y, PomMatica0, PomMatica;
77      my_class xx;
78
79      if (PocetStlpcov != PocetRiadkov) {
80          cout << "\n Matica sa neda umocnit!";
81          xx.my_getch();
82          exit(1);
83      }
84      Y.PocetRiadkov = Y.PocetStlpcov = PocetRiadkov;
85
86      switch (k) {
87      case 0:
88          for (unsigned i = 0; i<PocetRiadkov; i++) {
89              for (unsigned j = 0; j<PocetRiadkov; j++)
90                  Y.Matica[i][j] = 0;
91              Y.Matica[i][i] = 1;
92          }
93          break;
94      case 1:
95          Y = *this;
96          break;
97      default:
98          PomMatica0 = PomMatica = *this;
99          k--;
100         while (k--) {
101             Y = PomMatica * PomMatica0;
102             PomMatica = Y;
103         }
104     }
105
106     return Y;
107 }
108 //-----
109 istream& operator>>(istream& is, TMatica& X)
110 {
111     my_class xx;
112
113     is >> X.PocetRiadkov >> X.PocetStlpcov;
114     if (is.fail()) {
115         cout << "\n\n Vstupny subor sa nepodarilo otvorit!\n";
116         xx.my_getch();
117         exit(1);
118     }
119     for (unsigned i = 0; i<X.PocetRiadkov; i++)
120         for (unsigned j = 0; j<X.PocetStlpcov; j++)
121             is >> X.Matica[i][j];
122     return is;
123 }
124 //-----
125 ostream& operator<<(ostream& os, TMatica& X)
126 {
127     os.setf(ios::fixed, ios::floatfield);
128     os.precision(2);
129     for (unsigned i = 0; i<X.PocetRiadkov; i++) {
130         os << "\n ";
131         for (unsigned j = 0; j<X.PocetStlpcov; j++)
132             os << setw(8) << X.Matica[i][j];
133     }
134     os << "\n";
135     return os;
136 }
137 //-----
138 void my_class::my_getch() const
139 {
140     #ifdef _WIN32
141         _getch();
142     #else
143         cout << endl;
144     #endif
145 }
146 //-----

```