

```

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

```

```

73     TMaticaObdlznikova VyslMatica;
74     my_class xx;
75
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     }
83     VyslMatica.PocetRiadkov=LavaMatica.PocetRiadkov;
84     VyslMatica.PocetStlpcov=PravaMatica.PocetStlpcov;
85     for (i=0; i<LavaMatica.PocetRiadkov; i++)
86         for (j=0; j<PravaMatica.PocetStlpcov; j++){
87             Suma=0;
88             for (k=0; k<LavaMatica.PocetStlpcov; k++)
89                 Suma+=LavaMatica.Matica[i][k]*PravaMatica.Matica[k][j];
90             VyslMatica.Matica[i][j]=Suma;
91         }
92     return VyslMatica;
93 }
94 //-----
95 TMaticaObdlznikova& TMaticaObdlznikova::operator+=
96     (const TMaticaObdlznikova& PravaMatica)
97 {
98     *this = *this + PravaMatica;
99     return *this;
100 }
101 //-----
102 TMaticaObdlznikova& TMaticaObdlznikova::operator*=
103     (const TMaticaObdlznikova& PravaMatica)
104 {
105     *this = *this * PravaMatica;
106     return *this;
107 }
108 //-----
109 istream& operator>>(istream& is, TMaticaObdlznikova& 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
123     return is;
124 }
125 //-----
126 ostream& operator<<(ostream& os,const TMaticaObdlznikova& X)
127 {
128     os.setf(ios::fixed,ios::floatfield);
129     os.precision(2);
130     for (unsigned i=0;i<X.PocetRiadkov;i++) {
131         os << "\n ";
132         for (unsigned j=0;j<X.PocetStlpcov;j++)
133             os << setw(8) << X.Matica[i][j];
134     }
135     os << "\n";
136
137     return os;
138 }
139 //-----
140 void my_class::my_getch() const
141 {
142     #ifdef _WIN32
143         _getch();
144     #else
145         cout << endl;

```

```
146 #endif
147 }
148 //-----
149
150
```