

```
1 #ifdef _WIN32
2     #pragma warning(disable:4996)
3     #include <conio.h>
4     #include <direct.h>
5 #elif (defined __linux__ || (defined _AIX)
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 #include "MaticaStvorcova.h"
21 #include "CisloKomplexne.h"
22 #include "MaticaStvorcovaKomplexna.h"
23 //-----
24 TMaticaStvorcovaKomplexna::TMaticaStvorcovaKomplexna()
25 {
26     MaticaKomplexna = NULL;
27 }
28 //-----
29 //Konstruktor dynamicky alokuje pamat pre prvky matice a inicializuje ich
30 TMaticaStvorcovaKomplexna::TMaticaStvorcovaKomplexna(const unsigned n)
31 {
32     unsigned i,j;
33     TCisloKomplexne zero(0,0);
34
35     PocetRiadkov = PocetStlpcov = n;
36
37     MaticaKomplexna = new TCisloKomplexne*[PocetRiadkov];           // pocet [] a * je 3
38     for (i=0;i<PocetRiadkov;i++)
39         MaticaKomplexna[i] = new TCisloKomplexne[PocetStlpcov];   // pocet [] je 3
40
41     for (i=0; i<PocetRiadkov; i++)
42         for (j=0; j<PocetStlpcov; j++)
43             MaticaKomplexna[i][j]=zero;
44 }
45 //-----
46 //Kopirovaci konstruktor
47 TMaticaStvorcovaKomplexna::TMaticaStvorcovaKomplexna(TMaticaStvorcovaKomplexna& X)
48 {
49     unsigned i,j;
50
51     PocetRiadkov = X.PocetRiadkov;
52     PocetStlpcov = X.PocetStlpcov;
53
```

```
54 MaticaKomplexna = new TCisloKomplexne*[PocetRiadkov]; // pocet [] a * je 3
55 for (i=0;i<PocetRiadkov;i++)
56     MaticaKomplexna[i] = new TCisloKomplexne[PocetStlpcov]; // pocet [] je 3
57
58 for (i=0;i<PocetRiadkov;i++)
59     for (j=0;j<PocetStlpcov;j++)
60         MaticaKomplexna[i][j] = X.MaticaKomplexna[i][j];
61 }
62 //-----
63 //Destruktor upratuje alokovanu pamat
64 TMaticaStvorcovaKomplexna::~TMaticaStvorcovaKomplexna()
65 {
66     if (MaticaKomplexna != NULL) {
67         for (unsigned i = 0; i < PocetRiadkov; i++)
68             delete[] MaticaKomplexna[i];
69         delete[] MaticaKomplexna;
70         MaticaKomplexna = NULL;
71     }
72 }
73 //-----
74 TMaticaStvorcovaKomplexna& TMaticaStvorcovaKomplexna::operator=
75 (const TMaticaStvorcovaKomplexna& X)
76 {
77     if (this == &X) return *this;
78
79     for (unsigned i=0;i<PocetRiadkov;i++)
80         for (unsigned j=0;j<PocetStlpcov;j++)
81             MaticaKomplexna[i][j] = X.MaticaKomplexna[i][j];
82
83     return *this;
84 }
85 //-----
86 const TMaticaStvorcovaKomplexna operator+(const TMaticaStvorcovaKomplexna&
87     LavaMatica,const TMaticaStvorcovaKomplexna& PravaMatica)
88 {
89     TMaticaStvorcovaKomplexna VyslMatica(LavaMatica.PocetRiadkov);
90
91     for (unsigned i=0; i<LavaMatica.PocetRiadkov; i++)
92         for (unsigned j=0; j<LavaMatica.PocetStlpcov; j++)
93             VyslMatica.MaticaKomplexna[i][j] = LavaMatica.MaticaKomplexna[i][j]
94             +PravaMatica.MaticaKomplexna[i][j];
95
96     return VyslMatica;
97 }
98 //-----
99 const TMaticaStvorcovaKomplexna operator*(const TMaticaStvorcovaKomplexna&
100     LavaMatica,const TMaticaStvorcovaKomplexna& PravaMatica)
101 {
102     TMaticaStvorcovaKomplexna VyslMatica(LavaMatica.PocetRiadkov);
103     my_class xx;
104     unsigned i,j,k;
105     TCisloKomplexne Suma(0,0);
106
```

```
107     if (LavaMatica.PocetStlpcov!=PravaMatica.PocetRiadkov){
108         cout << "\nMatice sa nedaju nasobit!";
109         xx.my_getch();
110         exit(1);
111     }
112
113     for (i=0; i<LavaMatica.PocetRiadkov; i++)
114         for (j=0; j<PravaMatica.PocetStlpcov; j++){
115             for (k=0; k<LavaMatica.PocetStlpcov; k++) {
116                 Suma = Suma + LavaMatica.MaticaKomplexna[i][k]
117                     * PravaMatica.MaticaKomplexna[k][j];
118             /* Suma.Re = Suma.Re
119                + LavaMatica.MaticaKomplexna[i][k].Re * PravaMatica.MaticaKomplexna[k][j].Re
120                - LavaMatica.MaticaKomplexna[i][k].Im * PravaMatica.MaticaKomplexna[k][j].Im;
121             Suma.Im = Suma.Im
122                + LavaMatica.MaticaKomplexna[i][k].Re * PravaMatica.MaticaKomplexna[k][j].Im
123                + LavaMatica.MaticaKomplexna[i][k].Im * PravaMatica.MaticaKomplexna[k][j].Re;
124             */
125             }
126             VyslMatica.MaticaKomplexna[i][j]=Suma;
127         }
128
129     return VyslMatica;
130 }
131 //-----
132 TMaticaStvorcovaKomplexna& TMaticaStvorcovaKomplexna::operator+=
133     (const TMaticaStvorcovaKomplexna& PravaMatica)
134 {
135     *this = *this + PravaMatica;
136     return *this;
137 }
138 //-----
139 TMaticaStvorcovaKomplexna& TMaticaStvorcovaKomplexna::operator*=
140     (const TMaticaStvorcovaKomplexna& PravaMatica)
141 {
142     *this = *this * PravaMatica;
143     return *this;
144 }
145 //-----
146 istream& operator>>(istream& is, TMaticaStvorcovaKomplexna& X)
147 {
148     my_class xx;
149
150     for (unsigned i=0; i<X.PocetRiadkov; i++)
151         for (unsigned j=0; j<X.PocetStlpcov; j++)
152             is >> X.MaticaKomplexna[i][j];
153     if (is.fail()){
154         cout << "Subor MaticaKomplexna.TXT sa nepodarilo otvorit.";
155         xx.my_getch();
156         exit(1);
157     }
158
159     return is;
```

```
160 }
161 //-----
162 ostream& operator<<(ostream& os, const TMaticaStvorcovaKomplexna& X)
163 {
164     os.setf(ios::fixed, ios::floatfield);
165     os.precision(2);
166     for (unsigned i=0; i<X.PocetRiadkov; i++) {
167         os << "\n ";
168         for (unsigned j=0; j<X.PocetStlpcov; j++)
169             os << X.MaticaKomplexna[i][j] << " ";
170     }
171     os << "\n";
172
173     return os;
174 }
175 //-----
176 void not_child_class::not_child_method()
177 {
178     // TMaticaStvorcovaKomplexna A, zero(0, 0);
179     // A = zero;
180 }
181 //-----
182
183
```