

```
1 // exponentiate_matrix_classes.cpp
2 //
3 #ifndef _WIN32
4     #pragma warning(disable:4996)
5     #include <tchar.h>
6     #include <windows.h>
7     #include <conio.h>
8     #include <direct.h>
9 #elif (defined __linux__) || (defined _AIX)
10    #include <stdlib.h>
11    #include <sys/types.h>
12    #include <sys/stat.h>
13    #include <unistd.h>
14    typedef char _TCHAR;
15    #define _tmain main
16 #endif
17
18 #include <stdio.h>
19 #include <string.h>
20 #include <fstream>
21 #include <iostream>
22 using namespace std;
23
24 #include "Matica.h"
25
26 #define MAXLINE 255
27 #define MAXLEN 20
28
29 void GetNamesOfIOFiles(const char[], char[]);
30 //-----
31 int _tmain(int argc, _TCHAR* argv[])
32 {
33     TMatica A, B;
34     my_class xx;
35     ifstream in;
36     char nameoffile[MAXLINE];
37     int Exponent;
38
39     cout << "\n Tento program nacita z textoveho suboru stvorcovu maticu A,\n"
40          << "z klavesnice prirodzene cislo k a na obrazovku vypise k-tu \n"
41          << "mocninu matice A.";
42
43     GetNamesOfIOFiles("MATICA.TXT", nameoffile);
44     in.open(nameoffile, ios::in);
45     in >> A;
46     in.close();
47
48     cout << "\n\n Zadajte prirodzene cislo ako exponent matice A!\n Exponent=";
49     do
50         cin >> Exponent;
51     while (Exponent < 0);
52
53     B = A.UmocniMaticu(Exponent);
```

```
54
55     cout << "\n Z textoveho suboru sa nacitala matica A:\n";
56     cout << A;
57     cout << "\n " << Exponent;
58     switch (Exponent % 10) {
59     case 1: cout << "-va mocnina matice A je:\n"; break;
60     case 2: cout << "-ha mocnina matice A je:\n"; break;
61     case 3: cout << "-tia mocnina matice A je:\n"; break;
62     default: cout << "-ta mocnina matice A je:\n"; break;
63     }
64     cout << B;
65
66     xx.my_getch();
67     return 0;
68 }
69 //-----
70 void GetNamesOfIOFiles(const char name_of_input_file[], char path_to_input_file[])
71 {
72     char current_path[MAXLINE];
73     my_class xx;
74     current_path[0] = '\0';
75
76 #ifdef _WIN32
77     TCHAR exePath[MAXLINE];
78
79     HMODULE hModule = GetModuleHandle(NULL);
80     if (hModule != NULL) {
81         if (!GetModuleFileName(hModule, exePath, MAXLINE)) {
82             cout << "Nepodarila sa zistit cesta k exe-suboru.\n";
83             xx.my_getch();
84             exit(1);
85         }
86     }
87     else {
88         cout << "Module handle is NULL.\n" << endl;
89         xx.my_getch();
90         exit(1);
91     }
92
93     int iii;
94     bool flag = false;
95     for (iii = (int)wcslen(exePath); iii >= 0; iii--) {
96         if (!flag && exePath[iii] == '\\') {
97             current_path[iii + 1] = '\0';
98             flag = true;
99         }
100         if (flag)
101             current_path[iii] = (char)exePath[iii];
102     }
103 #elif defined __linux__
104     unsigned iii;
105     char line[MAXLINE];
106     FILE* fp;
```

```
107     if ((fp = popen("/bin/pwd", "r")) == NULL) {
108         perror("popen error");
109         exit(1);
110     }
111     if (fgets(line, MAXLINE, fp) == NULL) {
112         perror("fgets error");
113         exit(1);
114     }
115     pclose(fp);
116
117     iii = 0;
118     while (line[iii] != '\r' && line[iii] != '\n') {
119         current_path[iii] = line[iii];
120         iii++;
121     }
122     current_path[iii] = '\0';
123 #elif (defined _AIX)
124     unsigned iii;
125     char line[MAXLINE];
126     FILE* fp;
127     if ((fp = popen("user/bin/pwd", "r")) == NULL) {
128         perror("popen error");
129         exit(1);
130     }
131     if (fgets(line, MAXLINE, fp) == NULL) {
132         perror("fgets error");
133         exit(1);
134     }
135     pclose(fp);
136
137     iii = 0;
138     while (line[iii] != '\r' && line[iii] != '\n') {
139         current_path[iii] = line[iii];
140         iii++;
141     }
142     current_path[iii] = '\0';
143 #endif
144
145     path_to_input_file[0] = '\0';
146     strcat(path_to_input_file, current_path);
147 #if (defined __linux__) || (defined _AIX)
148     strcat(path_to_input_file, "/inputs/");
149 #elif (defined _WIN32)
150     strcat(path_to_input_file, "inputs\\");
151 #endif
152     strcat(path_to_input_file, name_of_input_file);
153 }
154 //-----
155
```