

```

1 // inheritance_simple.cpp
2 //
3 #ifdef _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) || (defined __APPLE__)
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 "MaticaObdlznikova.h"
25 #include "MaticaStvorcova.h"
26
27 #define MAXLINE 255
28 #define MAXLEN 20
29
30 void GetNamesOfIOFiles(const char[], char[]);
31 //-----
32 int _tmain(int argc, _TCHAR* argv[])
33 {
34     TMaticaStvorcova A, B, C;
35     my_class xx;
36     ifstream in;
37     char nameoffile[MAXLINE];
38     int Exponent;
39
40     cout << "\n Tento program nacita z textoveho suboru stvorcovu maticu A,\n"
41          << "z klavesnice prirodzene cislo k a na obrazovku a do textoveho \n"
42          << "suboru vypise k-tu mocninu matice A.\n";
43     GetNamesOfIOFiles("MATICA.TXT", nameoffile);
44     in.open(nameoffile, ios::in);
45     in >> A;
46     in.close();
47
48     cout << "\nZ textoveho suboru bola nacistana matica A:";
49     cout << A;
50     cout << "\nZadajte prirodzene cislo ako exponent matice A!\n Exponent=";
51     do
52         cin >> Exponent;
53     while (Exponent < 0);
54
55     B = A.UmocniMaticu(Exponent); // Overload the function pow(A, Exponent)
56                                 // according to our function UmocniMaticu(Exponent)!!!
57     cout << endl << Exponent;
58     switch (Exponent % 10) {
59     case 1: cout << "-va mocnina matice A je:"; break;
60     case 2: cout << "-ha mocnina matice A je:"; break;
61     case 3: cout << "-tia mocnina matice A je:"; break;
62     default: cout << "-ta mocnina matice A je:"; break;
63     }
64     cout << B;
65
66     C = A * A;
67     cout << "\nMatica C = A * A je:" << C;
68
69     // A += A;
70     // cout << "\nDvojnásobok matice A je:" << A;
71
72     xx.my_getch();
73     return 0;

```

```

74 }
75 //-----
76 void GetNamesOfIOFiles(const char name_of_input_file[], char path_to_input_file[])
77 {
78     char current_path[MAXLINE];
79     current_path[0] = '\0';
80
81 #ifdef _WIN32
82     TCHAR exePath[MAXLINE];
83
84     HMODULE hModule = GetModuleHandle(NULL);
85     if (hModule != NULL) {
86         if (!GetModuleFileName(hModule, exePath, MAXLINE)) {
87             cout << "Nepodarila sa zistit cesta k exe-suboru.\n";
88             exit(1);
89         }
90     }
91     else {
92         cout << "Module handle is NULL.\n" << endl;
93         exit(1);
94     }
95
96     int iii;
97     bool flag = false;
98     for (iii = (int)wcslen(exePath); iii >= 0; iii--) {
99         if (!flag && exePath[iii] == '\\') {
100             current_path[iii + 1] = '\0';
101             flag = true;
102         }
103         if (flag)
104             current_path[iii] = (char)exePath[iii];
105     }
106 #elif (defined __linux__) || (defined __APPLE__)
107     unsigned iii;
108     char line[MAXLINE];
109     FILE* fp;
110     if ((fp = popen("/bin/pwd", "r")) == NULL) {
111         perror("popen error");
112         exit(1);
113     }
114     if (fgets(line, MAXLINE, fp) == NULL) {
115         perror("fgets error");
116         exit(1);
117     }
118     pclose(fp);
119
120     iii = 0;
121     while (line[iii] != '\r' && line[iii] != '\n') {
122         current_path[iii] = line[iii];
123         iii++;
124     }
125     current_path[iii] = '\0';
126 #elif (defined _AIX)
127     unsigned iii;
128     char line[MAXLINE];
129     FILE* fp;
130     if ((fp = popen("user/bin/pwd", "r")) == NULL) {
131         perror("popen error");
132         exit(1);
133     }
134     if (fgets(line, MAXLINE, fp) == NULL) {
135         perror("fgets error");
136         exit(1);
137     }
138     pclose(fp);
139
140     iii = 0;
141     while (line[iii] != '\r' && line[iii] != '\n') {
142         current_path[iii] = line[iii];
143         iii++;
144     }
145     current_path[iii] = '\0';
146 #endif

```

```
147
148     path_to_input_file[0] = '\0';
149     strcat(path_to_input_file, current_path);
150 #if (defined __linux__ || (defined _AIX) || (defined __APPLE__))
151     strcat(path_to_input_file, "/inputs/");
152 #elif (defined _WIN32)
153     strcat(path_to_input_file, "inputs\\");
154 #endif
155     strcat(path_to_input_file, name_of_input_file);
156 }
157 //-----
158
```