

```

1 // inheritance_multiple_new.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 #include <iomanip>
23 using namespace std;
24
25 #include "MaticaObdlznikova.h"
26 #include "MaticaStvorcova.h"
27 #include "CisloKomplexne.h"
28 #include "MaticaStvorcovaKomplexna.h"
29
30 void GetNamesOfIOFiles(const char[], char[]);
31 //-----
32 int _tmain(int argc, _TCHAR* argv[])
33 {
34     my_class xx;
35     ifstream in;
36     char nameoffile[MAXLINE];
37     int Exponent;
38     unsigned m, n;
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 spolu s vysledkami dalsich\n"
43          << "operacii s realnymi a komplexnymi maticami.";
44     GetNamesOfIOFiles("MATICE.TXT", nameoffile);
45
46     in.open(nameoffile, ios::in);
47     in >> m >> n;
48     if (m != n) {
49         cout << "\n Prva nacistana matica nie je stvorcova a neda sa umocnit!";
50         xx.my_getch();
51         exit(1);
52     }
53     TMaticaStvorcova* A = new TMaticaStvorcova(n), * B = new TMaticaStvorcova(n);
54     in >> *A;
55
56     in >> m >> n;
57     if (m != n) {
58         cout << "\n Druha nacistana matica nie je stvorcova a neda sa umocnit!";
59         xx.my_getch();
60         exit(1);
61     }
62     TMaticaStvorcovaKomplexna* KA = new TMaticaStvorcovaKomplexna(n),
63          * KB = new TMaticaStvorcovaKomplexna(n);
64     in >> *KA;
65     in.close();
66
67     cout << "\n Z textoveho suboru boli nacistane matice A a KA:";
68     cout << *A << *KA;
69     cout << "\n Zadajte prirodzene cislo ako exponent matice A!\n      Exponent=";
70     do
71         cin >> Exponent;
72     while (Exponent < 0);
73

```

```

74 *B = A->UmocniMaticu(Exponent); // Try to overload the function pow(A, Exponent)
75 // according to our function
// UmocniMaticu(Exponent)!!!

76 cout << endl << Exponent;
77 switch (Exponent % 10) {
78 case 1: cout << "-va mocnina matice A je:"; break;
79 case 2: cout << "-ha mocnina matice A je:"; break;
80 case 3: cout << "-tia mocnina matice A je:"; break;
81 default: cout << "-ta mocnina matice A je:"; break;
82 }
83 cout << *B;
84
85 *B = *A + *A;
86 cout << "\nMatica B = A + A je:" << *B;
87
88 *B = *A * *A;
89 cout << "\nMatica B = A * A je:" << *B;
90
91 *B = *A;
92 *B += *A;
93 cout << "\nMatica A += A je:" << *B;
94
95 *B = *A;
96 *B *= *A;
97 cout << "\nMatica A *= A je:" << *B;
98
99 *KB = *KA + *KA;
100 cout << "\nMatica KB = KA + KA je:" << *KB;
101
102 *KB = *KA * *KA;
103 cout << "\nMatica KB = KA * KA je:" << *KB;
104
105 *KB = *KA;
106 *KB += *KA;
107 cout << "\nMatica KA += KA je:" << *KB;
108
109 *KB = *KA;
110 *KB *= *KA;
111 cout << "\nMatica KA *= KA je:" << *KB;
112
113 delete KB;
114 delete KA;
115 delete B;
116 delete A;
117
118 xx.my_getch();
119 return 0;
120 }
121 //-----
122 void GetNamesOfIOFiles(const char name_of_input_file[], char path_to_input_file[])
123 {
124     char current_path[MAXLINE];
125     my_class xx;
126     current_path[0] = '\0';
127
128 #ifdef _WIN32
129     TCHAR exePath[MAXLINE];
130
131     HMODULE hModule = GetModuleHandle(NULL);
132     if (hModule != NULL) {
133         if (!GetModuleFileName(hModule, exePath, MAXLINE)) {
134             cout << "Nepodarila sa zistit cesta k exe-suboru.\n";
135             xx.my_getch();
136             exit(1);
137         }
138     }
139     else {
140         cout << "Module handle is NULL.\n" << endl;
141         xx.my_getch();
142         exit(1);
143     }
144
145     int iii;

```

```

146     bool flag = false;
147     for (iii = (int)wcslen(exePath); iii >= 0; iii--) {
148         if (!flag && exePath[iii] == '\\') {
149             current_path[iii + 1] = '\\0';
150             flag = true;
151         }
152         if (flag)
153             current_path[iiii] = (char)exePath[iiii];
154     }
155 #elif (defined __linux__) || (defined __APPLE__)
156     unsigned iiii;
157     char line[MAXLINE];
158     FILE* fp;
159     if ((fp = popen("/bin/pwd", "r")) == NULL) {
160         perror("popen error");
161         exit(1);
162     }
163     if (fgets(line, MAXLINE, fp) == NULL) {
164         perror("fgets error");
165         exit(1);
166     }
167     pclose(fp);
168
169     iiii = 0;
170     while (line[iiii] != '\\r' && line[iiii] != '\\n') {
171         current_path[iiii] = line[iiii];
172         iiii++;
173     }
174     current_path[iiii] = '\\0';
175 #elif (defined _AIX)
176     unsigned iiii;
177     char line[MAXLINE];
178     FILE* fp;
179     if ((fp = popen("user/bin/pwd", "r")) == NULL) {
180         perror("popen error");
181         exit(1);
182     }
183     if (fgets(line, MAXLINE, fp) == NULL) {
184         perror("fgets error");
185         exit(1);
186     }
187     pclose(fp);
188
189     iiii = 0;
190     while (line[iiii] != '\\r' && line[iiii] != '\\n') {
191         current_path[iiii] = line[iiii];
192         iiii++;
193     }
194     current_path[iiii] = '\\0';
195 #endif
196
197     path_to_input_file[0] = '\\0';
198     strcat(path_to_input_file, current_path);
199 #if (defined __linux__) || (defined _AIX) || (defined __APPLE__)
200     strcat(path_to_input_file, "/inputs/");
201 #elif (defined _WIN32)
202     strcat(path_to_input_file, "inputs\\");
203 #endif
204     strcat(path_to_input_file, name_of_input_file);
205 }
206 //-----
207

```