

```

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 "CisloKomplexne.h"
20 //-----
21 TCisloKomplexne::TCisloKomplexne ()
22 {
23     Re = 0;
24     Im = 0;
25 }
26 //-----
27 TCisloKomplexne::TCisloKomplexne(const float re,const float im)
28 {
29     Re = re;
30     Im = im;
31 }
32 //-----
33 TCisloKomplexne::~TCisloKomplexne ()
34 {
35 }
36 //-----
37 TCisloKomplexne& TCisloKomplexne::operator=(const TCisloKomplexne& X)
38 {
39     if (this == &X) return *this;
40     Re = X.Re;
41     Im = X.Im;
42
43     return *this;
44 }
45 //-----
46 const TCisloKomplexne operator+(const TCisloKomplexne& X,const TCisloKomplexne& Y)
47 {
48     TCisloKomplexne Z;
49
50     Z.Re = X.Re + Y.Re;
51     Z.Im = X.Im + Y.Im;
52
53     return Z;
54 }
55 //-----
56 const TCisloKomplexne operator*(const TCisloKomplexne& X,const TCisloKomplexne& Y)
57 {
58     TCisloKomplexne Z;
59
60     Z.Re = X.Re*Y.Re - X.Im*Y.Im;
61     Z.Im = X.Re*Y.Im + X.Im*Y.Re;
62
63     return Z;
64 }
65 //-----
66 TCisloKomplexne TCisloKomplexne::HermitovskyZdruzene ()
67 {
68     TCisloKomplexne Y;
69
70     Y = *this;
71     Y.Im = -Im;
72
73     return Y;

```

```

74 }
75 //-----
76 istream& operator>>(istream& is, TCisloKomplexne& com)
77 {
78     float x;
79     char c1,c2;
80
81     do {
82         is.get(c1);
83     } while (isspace(c1));
84
85     if (c1=='i') {
86         com.Re=0;
87         com.Im=1;
88         return is;
89     } else
90         is.putback(c1);
91
92     c1 = ' ';
93     is >> x;
94     is.get(c1);
95     if (c1!='i'){
96         com.Re=x;
97         if ((c1!='+')&&(c1!='-'))
98             com.Im=0;
99         else{
100             is >> com.Im >> c2;
101             if (c1=='-')
102                 com.Im=(-1)*com.Im;
103         }
104     }else{
105         com.Re=0;
106         com.Im=x;
107     }
108
109     return is;
110 }
111 //-----
112 ostream& operator<<(ostream& os, const TCisloKomplexne& com)
113 {
114     os.setf(ios::fixed,ios::floatfield);
115     os.precision(2);
116     if ((com.Re!=0)|| (com.Im==0))
117         os << setw(6) << com.Re;
118     else
119         os << " ";
120
121     if (com.Im!=0){
122         if (com.Re!=0)
123             os.setf(ios::showpos);
124         else if (com.Im>0)
125             os << " ";
126         os << setw(8) << com.Im << 'i';
127         os.unsetf(ios::showpos);
128     }else{
129         os << " ";
130     }
131
132     return os;
133 }
134 //-----
135
136

```