```
//Sources:complex.cpp
#include "complex.h"
 1
 2
 3
      #include <iostream>
 4
      #include <cmath>
 5
 6
     bool complex::operator==(const complex & other)
 7
 8
          一般我们比较两个对象是否相等
会比较对象里的所有变量是否相等
若相等则认为两个对象相等
布尔值是c++关键字
 9
10
11
12
13
14
          return ( (real==other.real) && (imag==other.imag) );
15
16
17
      complex operator+(float a, complex c)
18
19
          这是非类里的函数,不需要类解析符
但是访问类里私有变量因此要在类里加友元并声明
重载了加号+运算符
20
21
22
           complex a=(float)c+(complex)b
相当于 a = c.operator+(b);
23
24
           对一个复数实部进行加运算,虚部不变
25
26
27
          complex temp;
28
          temp.real = a + c.real;
29
          temp.imag = c.imag;
30
          return temp;
31
32
33
     complex operator-(float a, complex c)
34
3.5
           这是非类里的函数,不需要类解析符
但是访问类里私有变量因此要在类里加友元并声明
重载了减号-运算符
36
37
38
          complex a=(float)c-(complex)b
相当于 a = c.operator-(b);
对一个复数实部进行减运算,虚部不变
39
40
41
42
43
          complex temp;
          temp.real = a - c.real;
temp.imag = c.imag;
44
45
          return temp;
46
47
48
49
      complex operator*(float a, complex c)
50
51
          这是非类里的函数,不需要类解析符
但是访问类里私有变量因此要在类里加友元并声明
重载了乘号*运算符
52
53
54
          complex a=(float)c*(complex)b
相当于 a = c.operator*(b);
对一个复数实部进行乘运算,虚部和实部分别相乘
55
56
57
58
59
          complex temp;
           temp.real = a * c.real;
60
          temp.imag = a * c.imag;
61
62
          return temp;
63
64
65
      complex operator/(float a, complex c)
66
67
           这是非类里的函数,不需要类解析符
但是访问类里私有变量因此要在类里加友元并声明
重载了除号/运算符
68
69
70
           complex a=(float)c/(complex)b
相当于 a = c.operator/(b);
71
72
73
           对一个复数实部进行除运算,虚部和实部分别相除
74
      * /
75
           if(c.real==0 & &c.imag==0)
76
77
                std::cout<<"Input Error!"<<std::endl;</pre>
78
79
          else
80
81
                complex temp;
               temp.real = a / c.real;
temp.imag = a / c.imag;
82
83
84
                return temp;
```

```
8.5
         }
86
87
88
89
      std::ostream &operator<<(std::ostream &output, complex c)</pre>
90
91
           重载了重定向<<运算符
92
           重定向到标准输出流,由于要改变值,所以做引用传递
93
94
          output<<"The complex is "<<c.real<<" + "<<c.imag<<"i"<<std::endl;
output<<"The complex 's real part is "<<c.real<<" ,its imag part is</pre>
9.5
96
      "<<c.imag<<std::endl;
97
          return output;
98
99
      std::istream &operator>>(std::istream &input, complex &c)
100
101
           重载了重定向>>运算符
102
           重定向到标准输入流,由于要改变值,所以做引用传递
103
104
           std::cout << "Please enter the real part:";</pre>
105
106
          input >> c.real;
           std::cout << "please enter the imaginery part:";</pre>
107
1.08
          input >> c.imag;
109
          return input;
110
      }
111
112
      complex complex::operator+(complex c)
113
114
           重载了加号+运算符
115
116
          complex a=(complex)b+(complex)c
相当于 a = b.operator+(c);
117
           对两个复数进行加运算
118
      * /
119
120
          complex temp;
          temp.real = real + c.real;
121
          temp.imag = imag + c.imag;
122
123
          return temp;
124
125
      complex complex::operator+(float c)
126
127
           重载了加号+运算符
128
           complex a=(complex)b+(float)c
相当于 a = b.operator+(c);
129
130
           对一个复数实部进行加运算,虚部不变
131
132
      * /
133
          complex temp;
134
          temp.real = real + c;
135
          temp.imag = imag;
136
          return temp;
137
      }
138
139
140
      complex complex::operator-(complex c)
141
142
           重载了减号-运算符
143
          complex a=(complex)b-(complex)c
相当于 a = b.operator-(c);
144
145
          对两个复数进行减运算
146
147
      * /
148
          complex temp;
149
          temp.real = real - c.real;
150
          temp.imag = imag - c.imag;
151
          return temp;
1.52
153
      complex complex::operator-(float c)
154
155
156
           重载了减号-运算符
          complex a=(complex)b-(float)c
相当于 a = b.operator-(c);
对一个复数实部进行减运算,虚部不变
1.57
1.58
159
      * /
160
161
          complex temp;
          temp.real = real - c;
temp.imag = imag;
1 62
163
164
          return temp;
165
166
167
      complex complex::operator*(complex c)
```

```
1 68
169
           重载了乘号*运算符
170
          complex a=(complex)b*(complex)c
相当于 a = b.operator*(c):
171
172
                  a = b.operator*(c);
          对两个复数进行乘运算
(a+bi)*(c+di)=(ac-bd)+(bc+ad)i,
173
174
175
          a=real;
176
          b=imag;
177
          c=c.real;
178
          d=c.imag;
179
          temp.real=(real*c.real-imag*c.imag)
180
          temp.imag=(imag*c.real-real*c.imag)
181
182
          if(c.real==0 & &c.imaq==0)
183
184
               std::cout<<"Input Error!"<<std::endl;</pre>
185
186
          else
187
188
               complex temp;
189
               temp.real = (real*c.real-imag*c.imag);
190
               temp.imag = (imag*c.real-real*c.imag);
191
               return temp;
1 92
193
194
      complex complex::operator*(float c)
195
196
1 97
           重载了乘号*运算符
          complex a=(complex)b*(float)c
相当于 a = b.operator*(c);
1 98
199
          对一个复数实部进行乘运算,虚部和实部分别相乘
200
201
      * /
202
          complex temp;
203
          temp.real = real * c;
          temp.imag = imag * c;
204
205
          return temp;
206
      }
207
208
      complex complex::operator/(complex c)
209
210
           重载了除号/运算符
211
          complex a=(complex)b/(complex)c
相当于 a = b.operator/(c);
对两个复数进行/运算
212
213
214
           (a+bi) / (c+di) = ((ac+bd) / (c^2+d^2)) + ((bc-ad) / (c^2+d^2)) i,
215
216
          a=real;
217
          b=imag;
218
          c=c.real;
219
          d=c.imag;
220
          temp.real=(real*c.real-imag*c.imag)/(c.real*c.real+c.imag*c.imag);
221
          temp.imag=(imag*c.real-real*c.imag)/(c.real*c.real+c.imag*c.imag);
222
223
          if(c.real==0 & &c.imag==0)
224
225
               std::cout<<"Input Error!"<<std::endl;</pre>
226
227
          else
228
229
               complex temp;
               temp.real = (real*c.real-imag*c.imag) / (c.real*c.real+c.imag*c.imag);
230
               temp.imag = (imag*c.real-real*c.imag)/(c.real*c.real+c.imag*c.imag);
231
232
               return temp;
233
234
235
      complex complex::operator/(float c)
236
237
           重载了减号/运算符
238
           complex a=(complex)b/(float)c
相当于 a = b.operator/(c);
239
240
          对一个复数实部进行除运算,虚部和实部分别相除
241
242
243
          if(c==0)
244
245
               std::cout<<"Input Error!"<<std::endl;</pre>
246
247
          else
248
249
               complex temp;
250
               temp.real = real / c;
               temp.imag = imag / c;
251
```

```
252
               return temp;
253
254
255
256
     float complex::abs()
257
258
           return sqrt(real*real+imag*imag);
259
260
261
      complex::complex()
262
263
           real=0.0;
264
           imag=0.0;
265
266
      complex::complex(float r,float i)
267
268
269
           real=r;
270
          imag=i;
271
272
273
      void complex::PrintComplex()
274
           std::cout<<"The complex is ("<<real<<") + ("<<imag<<") i"<<std::endl;
std::cout<<"The complex 's real part is "<<real<<" ,its imag part is "<<imag<<std::endl;</pre>
275
276
277
278
279
      void complex::setComplex(float r, float i)
280
           real=r;
281
282
          imag=i;
283
284
285
      float complex::getImag()
286
287
           return imag;
288
289
290
      float complex::getReal()
291
292
           return real;
293
294
295
```