```
 1 #include <iostream>
 2
 3 #include <cppqt.h>
 4
 5 using namespace std;
 6 //########################################################################
 7 // malt vier Punkte oder zwei Linien
 8 void drawEllipsePoints(Drawing& pic, int x, int y, int xcenter, int ycenter,
 9                        bool filled, int colour = 0)
10 {
11 //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
12   if (!filled)
13     {
14       pic.drawPoint(-x + xcenter,  y + ycenter, colour, true);
15       pic.drawPoint( x + xcenter,  y + ycenter, colour, true);
16       pic.drawPoint(-x + xcenter, -y + ycenter, colour, true);
17       pic.drawPoint( x + xcenter, -y + ycenter, colour, true);
18     }
19   else
20     {
21       pic.drawLine(-x + xcenter,  y + ycenter, x + xcenter,  y + ycenter,
22                    colour, true);
23       pic.drawLine(-x + xcenter, -y + ycenter, x + xcenter, -y + ycenter,
24                    colour, true);
25     }
26 //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
27 }
28
29 //########################################################################
30 // Scan Conversion für Ellipse, reelle Rechnung
31 void drawEllipse_float(Drawing& pic, IPoint2D center, int a, int b, bool filled,
32                        int colour = 0)
33 {
34 //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

```
35   int x     = 0;
36   int y     = b;
37   double a2 = a * a;
38   double b2 = b * b;
39   double d  = b - b2 / a2 - 0.25;
40
41   drawEllipsePoints(pic, x, y, center.x, center.y, filled, colour);
42
43   while (a2 * y > b2 * x)
44     {
45       if (d < 0)
46         {
47           --y;
48           d += 2 * y - b2 / a2 * (2 * x + 3);
49         }
50       else
51         d -= b2 / a2 * (2 * x + 3);
52       ++x;
53       drawEllipsePoints(pic, x, y, center.x, center.y, filled, colour);
54     }
55
56   d = a2 - a2/b2*y*y - x*x - x + a2/b2*(2*y-1) - 0.25;
57
58   while (y >= 0)
59     {
60       if (d < 0)
61         d += a2 / b2 * (2 * y - 3);
62       else
63         {
64           ++x;
65           d -= 2 * x - a2 / b2 * (2 * y - 3);
66         }
67       --y;
68       drawEllipsePoints(pic, x, y, center.x, center.y, filled, colour);
```

```
69      }
70  //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
71  }
72
73  //########################################################################
74  // Scan Conversion für Ellipse, ganzzahlige Rechnung
75  void drawEllipse(Drawing& pic, IPoint2D center, int a, int b, bool filled,
76                   int colour = 0)
77  {
78  //<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
79    int x  = 0;
80    int y  = b;
81    int a2 = a * a;
82    int b2 = b * b;
83    int d = 4 * a2 * b - 4 * b2 - a2;
84
85    drawEllipsePoints(pic, x, y, center.x, center.y, filled, colour);
86
87    while (a2 * y > b2 * x)
88      {
89        if (d < 0)
90          {
91            --y;
92            d += 8 * a2 * y - b2 * (8 * x + 12);
93          }
94        else
95          d -= b2 * (8 * x + 12);
96        ++x;
97        drawEllipsePoints(pic, x, y, center.x, center.y, filled, colour);
98      }
99
100   d = 4*b2*a2 - 4*a2*y*y - 4*b2*x*x - 4*b2*x + 4*a2*(2*y-1) - b2;
101
102   while (y >= 0)
```

```
103     {
104       if (d < 0)
105         d += a2 * (8 * y - 12);
106       else
107         {
108           ++x;
109           d -= 8 * b2 * x - a2 * (8 * y - 12);
110         }
111       --y;
112       drawEllipsePoints(pic, x, y, center.x, center.y, filled, colour);
113     }
114 //>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
115 }
116
117 //########################################################################
118 int maindraw( )
119 {
120   Drawing pic1(200, 200);
121   Drawing pic2(pic1);
122
123   pic1.show( );
124   pic1.setZoom(2);
125   pic2.show( );
126   pic2.setZoom(2);
127
128   IPoint2D center;
129   int a, b;
130   int colour;
131
132   while (true)
133     {
134       cout << "Eingabe von center, a, b, colour: ";
135       cin >> center >> a >> b >> colour;
136       if (center.x < 0 || center.y < 0)
```

```
137         break;
138       pic1.show();
139       drawEllipse(pic1, center, a, b, false, colour);
140       cin.get();
141       cout << "Weiter mit Return" << endl;
142       cin.get();
143       pic2.show();
144       drawEllipse(pic2, center, a, b, true, colour);
145       cout << "Weiter mit Return" << endl;
146       cin.get();
147     }
148
149   cout << endl;
150   IOThread::waitForWindow(60);
151
152   return 0;
153 }
154 /*
155 (100,100) 25 100 0
156
157
158 (100,100) 100 60 70
159
160
161 (10,100) 10 100 100
162
163
164 (-1,-1) 1 1 1
165 */
166
```