

```
1 #include <algorithm>
2 #include <iostream>
3
4 #include <cppqt.h>
5
6 using namespace std;
7
8
9 void drawLine(Drawing& pic, IPoint2D p1, IPoint2D p2, int colour = 0)
10 {
11     // zeichnet eine Linie von p1 nach p2
12     int dx = p2.x - p1.x;
13     int dy = p2.y - p1.y;
14     int x = p1.x;
15     int y = p1.y;
16
17     // Sonderfälle zuerst
18     if (dy == 0)
19         // Linie W-O oder O-W
20         for (x = min(p1.x, p2.x); x <= max(p1.x, p2.x); ++x)
21             pic.drawPoint(x, y, colour, true);
22     else if (dx == 0)
23         // Linie N-S oder S-N
24         for (y = min(p1.y, p2.y); y <= max(p1.y, p2.y); ++y)
25             pic.drawPoint(x, y, colour, true);
26     else
27     {
28         double m = static_cast<double>(dy) / dx;
29
30         if (m > 0 && m <= 1 && p1.x < p2.x)
31         {
32             // Linie WSW-ONO // schnelle Richtung: x, y positiv
33             int mm = 2 * dy;
34             int halb = dx;
```

```
35     int eins = 2 * halb;
36     int D = 0;
37     pic.drawPoint(x, y, colour, true);
38     for (++x; x <= p2.x; ++x)
39     {
40         D += mm;
41         if (D > halb) // NO
42         {
43             ++y;
44             D -= eins;
45         }
46         pic.drawPoint(x, y, colour, true);
47     }
48 }
49 else if (m > 0 && m <= 1 && p1.x > p2.x)
50 {
51     // Linie ONO-WSW                // schnelle Richtung: -x, y negativ
52     int mm = -2 * dy;                // Vorzeichen getauscht
53     int halb = -dx;                  // Vorzeichen getauscht
54     int eins = 2 * halb;
55     int D = 0;
56     pic.drawPoint(x, y, colour, true);
57     for (--x; x >= p2.x; --x)        // Vorzeichen getauscht
58     {
59         D += mm;
60         if (D >= halb) // SW
61         {
62             --y;                    // Vorzeichen getauscht
63             D -= eins;
64         }
65         pic.drawPoint(x, y, colour, true);
66     }
67 }
68 else if (m < 0 && m >= -1 && p1.x < p2.x)
```

```
69     {
70         // Linie WNW-OSO                // schnelle Richtung: x, y negativ
71         int mm = 2 * dy;
72         int halb = dx;
73         int eins = 2 * halb;
74         int D = 0;
75         pic.drawPoint(x, y, colour, true);
76         for (++x; x <= p2.x; ++x)
77             {
78                 D -= mm;                // Vorzeichen getauscht
79                 if (D > halb) // SO
80                     {
81                         --y;            // Vorzeichen getauscht
82                         D -= eins;
83                     }
84                 pic.drawPoint(x, y, colour, true);
85             }
86     }
87     else if (m < 0 && m >= -1 && p1.x > p2.x)
88     {
89         // Linie OSO-WNW                // schnelle Richtung: -x, y positiv
90         int mm = -2 * dy;                // Vorzeichen getauscht
91         int halb = -dx;                // Vorzeichen getauscht
92         int eins = 2 * halb;
93         int D = 0;
94         pic.drawPoint(x, y, colour, true);
95         for (--x; x >= p2.x; --x)        // Vorzeichen getauscht
96             {
97                 D -= mm;                // Vorzeichen getauscht
98                 if (D >= halb) // NW
99                     {
100                         ++y;
101                         D -= eins;
102                     }

```

file:///home/jimenez/Teaching/Bildgenerierung/WS2324/Uebungen/02/linien.cc

```
137         if (D >= halb) // SW
138             {
139                 --x;
140                 D -= eins;
141             }
142         pic.drawPoint(x, y, colour, true);
143     }
144 }
145 else if (p1.y < p2.y) // && m < -1
146 {
147     // Linie SSO-NNW           // schnelle Richtung: y, x negativ
148     int mm = 2 * dx;
149     int halb = dy;
150     int eins = 2 * halb;
151     int D = 0;
152     pic.drawPoint(x, y, colour, true);
153     for (++y; y <= p2.y; ++y)
154     {
155         D -= mm;
156         if (D > halb) // NW
157         {
158             --x;
159             D -= eins;
160         }
161         pic.drawPoint(x, y, colour, true);
162     }
163 }
164 else
165 {
166     // Linie NNW-SSO           // schnelle Richtung: -y, x positiv
167     int mm = -2 * dx;
168     int halb = -dy;
169     int eins = 2 * halb;
170     int D = 0;
```

file:///home/jimenez/Teaching/Bildgenerierung/WS2324/Uebungen/02/linien.cc

```
205
206     cout << endl;
207     IOThread::waitForWindow(60);
208
209     return 0;
210 }
211
212 /*
213 ( 0,100) (199,100) 127
214 (100, 0) (100,199) 200
215 ( 29, 29) (171,171) 50
216 ( 29,171) (171, 29) 25 */
217
```