

Übungen - Bildgenierung

Aufgaben 01-03.

Jose Jimenez

Angewandte Informatik
Bergische Universität Wuppertal

October 26, 2022



Table of Contents

1 Aufgabe 01

2 Aufgabe 02

3 Aufgabe 03



Aufgabe 01

Test des Frameworks

`/home/bildgen/cppqt/doc/index.html`

Klassen-Verzeichnis -> UML Klassendiagramm

`Point2D< T >`
`drawPoint()`
`DrawColour()`



Aufgabe 02

Einfache Linien

Wir

- 1 implementieren die Funktion, die die Linien zeichnet.
- 2 verwenden sie in der `maindraw()` Funktion.



Argumentenliste

// zeichnet eine Linie von p1 nach p2

```
void zeichneLinie(Drawing& pic, IPoint2D p1, IPoint2D p2, DrawColour c)
{
```



Aufgabe 02

Einfache Linien

```
void zeichneLinie(Drawing& pic, IPoint2D p1, IPoint2D p2, DrawColour c)
{
    // bestimme Abstand von p1 und p2 und male entsprechend viele
    // Punkte dazwischen

    int len = round(norm(p2 - p1)); //l =  $\|p_2 - p_1\|$ 
```



Aufgabe 02

Einfache Linien

```
// zeichnet eine Linie von p1 nach p2
void zeichneLinie(Drawing& pic, IPoint2D p1, IPoint2D p2, DrawColour c)
{
    // bestimme Abstand von p1 und p2 und male entsprechend viele
    // Punkte dazwischen
    int len = round(norm(p2 - p1)); //l = ||p2 - p1||
    IPoint2D r=(p2-p1);//r = p2 - p1
}
```



Aufgabe 02

Einfache Linien

```
// zeichnet eine Linie von p1 nach p2
void zeichneLinie(Drawing& pic, IPoint2D p1, IPoint2D p2, DrawColour c)
{
    // bestimme Abstand von p1 und p2 und male entsprechend viele
    // Punkte dazwischen
    int len = round(norm(p2 - p1)); //  $l = ||p_2 - p_1||$ 
    IPoint2D r=(p2-p1); //  $r = p_2 - p_1$ 

    double delta;
    for(int k = 0; k <= len; ++k){ // durch die Linie
        delta = k / len;
    }
```


Aufgabe 02

Einfache Linien

```
// zeichnet eine Linie von p1 nach p2
void zeichneLinie(Drawing& pic, IPoint2D p1, IPoint2D p2, DrawColour c)
{
    // bestimme Abstand von p1 und p2 und male entsprechend viele
    // Punkte dazwischen
    int len = round(norm(p2 - p1)); //l = ||p2 - p1||
    IPoint2D r=(p2-p1); //r = p2 - p1

    double delta;
    for(int k = 0; k <= len; ++k){ // durch die Linie
        delta = static_cast<double>(k) / len;  $\Delta = \frac{k}{l}$ 

    }
}
```

Aufgabe 02

Einfache Linien

```
void zeichneLinie(Drawing& pic, IPoint2D p1, IPoint2D p2, DrawColour c)
{
    // bestimme Abstand von p1 und p2 und male entsprechend viele
    // Punkte dazwischen
    int len = round(norm(p2 - p1)); //l = ||p2 - p1||
    IPoint2D r=(p2-p1); //r = p2 - p1

    double delta;
    IPoint2D q;

    for(int k = 0; k <= len; ++k){ // durch die Linie
        delta = static_cast<double>(k) / len;
        q = p1 + delta * r ; //q = p1 + Δr
    }
```

Richtig?

```
void zeichneLinie(Drawing& pic, IPoint2D p1, IPoint2D p2, DrawColour c)
{
    int len = round(norm(p2 - p1)); // l = ||p2 - p1||
    IPoint2D r=(p2-p1); // r = p2 - p1

    double delta;
    IPoint2D q;

    for(int k = 0; k <= len; ++k){ // durch die Linie
        delta = static_cast<double>(k) / len;
        q = p1 + delta * r ; // q = p1 + Δr
        pic.drawPoint(q, c, true);
    }
```

Aufgabe 02

Einfache Linien

Fehler!

```
void zeichneLinie(Drawing& pic, IPoint2D p1, IPoint2D p2, DrawColour c)
{
    int len = round(norm(p2 - p1)); //l = ||p2 - p1||
     $\vec{r}$  IPoint2D r=(p2-p1); //r = p2 - p1

    double delta;
    IPoint2D q;

    for(int k = 0; k <= len; ++k){ // durch die Linie
        delta = static_cast<double>(k) / len;
         $\vec{q}$  q = p1 + delta * r ; //q = p1 + Δr
        pic.drawPoint(q, c, true);
    }
```

Aufgabe 02

Einfache Linien

```
void zeichneLinie(Drawing& pic, IPoint2D p1, IPoint2D p2, DrawColour c)
{
    int len = round(norm(p2 - p1)); // l = ||p2 - p1||
    DPoint2D r;
    r=static_cast<DPoint2D>(p2-p1); // r = p2 - p1

    double delta;
    IPoint2D q;

    for(int k = 0; k <= len; ++k){ // durch die Linie
        delta = static_cast<double>(k) / len;
        q = round(static_cast<DPoint2D>(p1) + delta * r ); // q = p1 + Δr
        pic.drawPoint(q, c, true);
    }
```

Aufgabe 02

Einfache Linien

```
int maindraw(){
    Drawing pic(200, 200);
    pic.show();
    pic.setZoom(1);

    IPoint2D p1, p2;

    while(true)    {
        //etwas, was?
    }

    IOThread::waitForWindow(60); //to close
    return 0;
}
```

Aufgabe 02

Einfache Linien

```
int maindraw(){  
    ...  
  
    while(true)    {  
        if (p1.x < 0 || p1.y < 0 || p2.x < 0 || p2.y < 0)  
            break;  
        zeichneLinie(pic, p1, p2, c); // Color?  
    }  
    ...  
}
```



Aufgabe 02

Einfache Linien

...

//Idee: Just some random numbers

```
default_random_engine generator;  
uniform_int_distribution<int> distribution(0, 255);  
auto dice = bind(distribution, generator);  
  
bind(distribution, generator);  
DrawColour c(dice(), dice(), dice());
```

...

}



Aufgabe 02

Einfache Linien

run!



Aufgabe 03

Einfache Kreise

Quadrat und Kreis

