

C++ Übungsblatt 3: Vererbung

Aufgabe 3-1

Fortsetzung zu 1-1

Auf der nächsten Seite (oder auf GitHub¹) finden Sie eine Implementierung der Klasse *Ellipse*, mit der es möglich ist, gefüllte Ovale in der Konsole zu zeichnen. Der Einfachheit halber wurde auf einen Rahmen, wie bei *Rectangle* aus 1-1 der Fall war, verzichtet. Testen Sie diese aus.

- a) Wie vielleicht auffällt, gibt es Ähnlichkeiten zur *Rectangle* Klasse – beide verwenden *width* und *height* für ihre Größe und beide verwenden die Methode *draw()* um das Konstrukt zu zeichnen.

Da es gut möglich ist, noch mehr dieser Formen zu implementieren, die die gleichen Grundeigenschaften besitzen, ist es durchaus sinnvoll, diese über einer Elternklasse zu abstrahieren.

- Erstellen Sie hierzu eine Klasse *Shape*. Diese soll die Attribute *width*, *height* sowie *fill* beinhalten, die über den Konstruktor gesetzt werden können. Zusätzlich sollen alle Kindklassen, die von *Shape* erben, die Funktion *draw()* implementieren.
- Lassen Sie die Klassen *Rectangle* und *Ellipse* von *Shape* erben. Entfernen Sie dabei auch sämtliche Attribute, die bereits in *Shape* vorhanden sind.
- Erstellen Sie anschließend die Klassen *Triangle* und *Line*, die jeweils ein Dreieck und eine diagonale Linie zeichnet. Ihnen steht dabei frei, noch weitere geometrische Formen hinzuzufügen.

- b) Erstellen Sie eine Klasse *Canvas*, mit der sich ein Bild aus einzelnen Shapes zeichnen lässt.

- Im Konstruktor soll lediglich die Größe der Zeichnung angegeben werden
- Über die Methode *add_shape* soll ein Objekt, das von *Shape* erbt, zusammen mit einer Koordinate übergeben werden können
- Über die Methode *clear* soll die Zeichnung gelöscht werden
- Über die Methode *draw* soll die gesamte Zeichnung ausgegeben werden

Hinweis: Es müssen durchaus Änderungen an *Shape* vorgenommen werden, um das Zeichnen von überlappenden Formen zu ermöglichen.

- c) Um die Verwirrung nun komplett zu machen, lassen Sie *Canvas* von *Shape* erben und versuchen Sie, ein Bild aus mehreren *Canvas* zu machen.

¹ <https://github.com/jassler/programming1-cpp-solutions>

Anerkennung an Xinux für seine Implementierung einer Ellipse
<https://stackoverflow.com/a/41745421/7669319>

```
1  #include <iostream>
2  #include <vector>
3  #define M_PI 3.14159265358979323846
4
5  class Ellipse
6  {
7  private:
8      int width;
9      int height;
10
11     char fill = '#';
12
13 public:
14     Ellipse(int _width, int _height, char _fill = '#')
15     : width(_width), height(_height), fill(_fill)
16     {}
17
18     void draw() {
19         double center_x = width / 2.0;
20         double center_y = height / 2.0;
21
22         double width_sq = center_x * center_x;
23         double height_sq = center_y * center_y;
24
25         for(int y = 0; y < height; ++y) {
26             for(int x = 0; x < width; ++x) {
27                 double d_x = (double) x - center_x;
28                 double d_y = (double) y - center_y;
29
30                 if(d_x * d_x / width_sq + d_y * d_y / height_sq <= 1) {
31                     std::cout << fill;
32                 } else {
33                     std::cout << ' ';
34                 }
35             }
36             std::cout << '\n';
37         }
38
39     } // void draw
40
41     double area() {
42         return M_PI * width * height / 4;
43     }
44 };
```