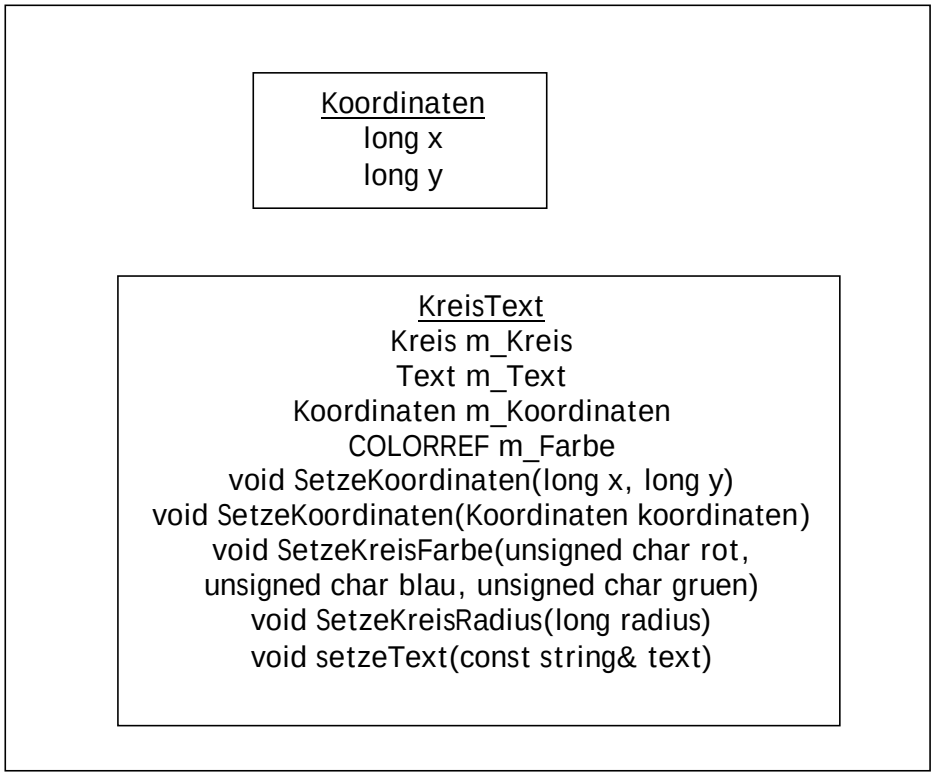
Übung
Zeichnen mit Klassen, Lösung

Analyse

Mit der Analyse verschaffen wir uns einen Überblick über die mögliche Lösung. Wir suchen grob die Klassen, deren Datenelemente und die wichtigsten Methoden. Wir müssen die Datentypen hier noch nicht definieren, genauso können Hilfsfunktionen etc. fehlen.
In der Lösung werden wir nur eine neue Klasse brauchen. Diese Klasse wird sicher zwei Datenelemente besitzen, nämlich einen Kreis und einen Text, da ein Objekt aus einem Text und einem Kreis bestehen soll. Zusätzlich wird die Klasse die Koordinaten und die Farbe des Kreises enthalten. Als Methoden wählen wir vorerst die Möglichkeit die Farbe und den Radius des Kreises zu setzen, möglicherweise sind wir sogar in der Lage den Text je nach Kreisfarbe schwarz oder weiss zu machen. Eine weitere Methode dient zum positionieren des ganzen Gebildes.
Das Analyse-Modell wird also etwa so aussehen :

Design

Im Design verfeinern wir das Analysemodell, wir bestimmen die genauen Datentypen, was in unserem Beispiel nicht so schwierig ist, denn zwei Datenelemente sind vom Datentyp bereits gegeben. Für die Koordinaten könnten wir eine kleine Struktur verwenden, die wir neu definieren. Die Farbe merken wir uns als COLORREF, den Datentyp den Windows für Farben meistens verwendet.
Bei den Methoden können wir bei SetzeKoordinaten(..) als Parameter die x und y Koordinaten oder auch eine Koordinaten-Struktur verwenden. Ein Koordinaten Struktur besteht aus zwei long-Datenelementen. Bei SetzeKreisFarbe wählen wir als Parameter die drei Grundfarben, denn so haben wir die Möglichkeit festzustellen ob der Text weiss oder schwarz sein soll.
Um die beiden Elemente auf den Bildschirm zu bringen müssen wir ZeichenFlaeche::kreisHinzufuegen und ZeichenFlaeche::textHinzufuegen aufrufen. Am besten ist es wir machen eine weitere Methode, die als Parameter die ZeichenFlaeche hat und in der Methode rufen wir die beiden Funktionen dann auf.
Da beide Datenelemente einen Konstruktor mit Parametern haben, machen wir in unserer Klasse auch am besten einen Konstruktor mit Parametern.
Das Designmodell sieht demnach ungefähr so aus:



Implementation

Jetzt endlich sehen wir den Code, den wir brauchen. Es kann auch passieren, dass wir beim implementieren Korrekturen an unserem Analyse-Modell machen müssen.
Hier die Datei

KreisText.h

```
#ifndef KREISTEXT_H // der sogenannte Include-Blocker
#define KREISTEXT_H

#include "Kreis.h"
#include "Text.h"
#include "ZeichenFlaeche.h"
#include <string>


struct Koordinaten
{
	long x;
	long y;
};


class KreisText
{
	public:
		// Zwei Konstruktoren mit Parametern
```

```
            // Der erste nimmt zwei longs als Koordinaten
            KreisText(long x, long y, long kreisRadius);
            // Dieser zweite Konstruktor nimmt als Parameter
            // eine Koordinaten-Struktur
            KreisText(Koordinaten koordinaten, long kreisRadius);

            // hier zwei Funktionen zum setzen der Koordinaten
            // auch wieder einmal mit longs, die andere mit
            // einer Koordinaten-Struktur
            void setzeKoordinaten(long x, long y);
            void setzeKoordinaten(Koordinaten koordinaten);

            void setzeKreisFarbe(unsigned char rot,
                                 unsigned char gruen,
                                 unsigned char blau);

            void setzeKreisRadius(long radius);

            void setzeText(const std::string& text);

            // Hier noch die Hilfsfunktion um die
            // Elemente auf die Zeichenflaeche zu bringen
            void aufZeichenFlaeche(ZeichenFlaeche& zeichenFlaeche);

      private:
            Kreis       m_Kreis;
            Text        m_Text;
            COLORREF    m_Farbe;
            Koordinaten m_Koordinaten;
};


#endif
```

Hier auch gleich die .cpp Datei

KreisText.cpp

```
#include "KreisText.h"

const COLORREF Schwarz = RGB(0,0,0);
const COLORREF Weiss = RGB(255,255,255);

////////////////////////////////////////////////////////////
// Der Konstruktor muss eine Initialisierungsliste
// verwenden, denn die Elemente haben auch Konstruktoren
// mit Parametern
KreisText::KreisText(long x, long y, long kreisRadius)
          :m_Text(x,y), m_Kreis(x, y, kreisRadius)
{
}

////////////////////////////////////////////////////////////

KreisText::KreisText(Koordinaten koordinaten, long kreisRadius)
          :m_Text(koordinaten.x, koordinaten.y),
               m_Kreis(koordinaten.x, koordinaten.y, kreisRadius)
{
}

////////////////////////////////////////////////////////////
```

```
void KreisText::setzeKoordinaten(long x, long y)
{
      m_Kreis.setzeKoordinaten(x, y);
      m_Text.setzeKoordinaten(x, y);
}

////////////////////////////////////////////////////////////

void KreisText::setzeKoordinaten(Koordinaten koordinaten)
{
      m_Kreis.setzeKoordinaten(koordinaten.x, koordinaten.y);
      m_Text.setzeKoordinaten(koordinaten.x, koordinaten.y);
}

////////////////////////////////////////////////////////////

void KreisText::setzeKreisFarbe(unsigned char rot,
                                                unsigned char gruen,
                                                unsigned char blau)
{
      // Hier müssen wir entscheiden ob wir
      // den Text schwarz oder weiss machen
      COLORREF textFarbe = Weiss;

      if(blau+rot+gruen > 400)
      {
            textFarbe = Schwarz;
      }

      if(gruen > blau+rot+150)
      {
            textFarbe = Schwarz;
      }

      m_Kreis.setzeFarbe(RGB(rot, gruen, blau));
      m_Text.setzeFarbe(textFarbe);
}

////////////////////////////////////////////////////////////

void KreisText::setzeKreisRadius(long radius)
{
      m_Kreis.setzeRadius(radius);
}

////////////////////////////////////////////////////////////

void KreisText::setzeText(const string& text)
{
      m_Text.setzeText(text);
}

////////////////////////////////////////////////////////////

void KreisText::aufZeichenFlaeche(ZeichenFlaeche& zeichenFlaeche)
{
      zeichenFlaeche.kreisHinzufuegen(m_Kreis);
      zeichenFlaeche.textHinzufuegen(m_Text);
}

////////////////////////////////////////////////////////////
```