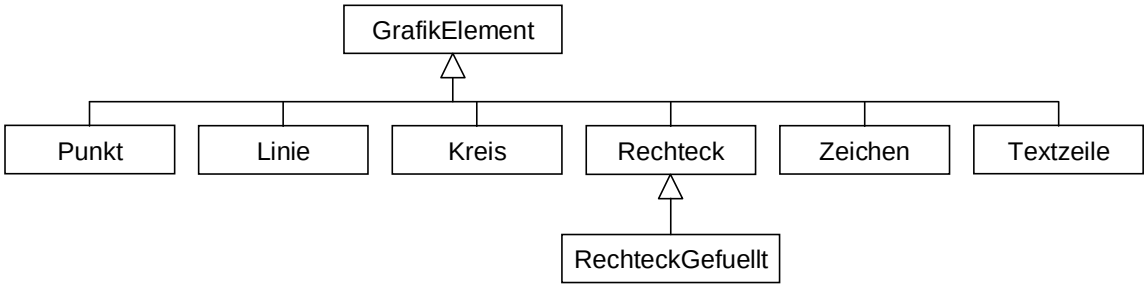
FH Osnabrück
FB Elektrotechnik und Informatik
Prof. Dr.-Ing. B. Lang

Objektorientierte Programmierung
Praktikumsübung 6

Aufgabe (10 Punkte)

Erstellen Sie eine Klassenhierarchie für Grafikobjekte, mit der Punkte, Linien, Kreise, Rechtecke, Zeichen und gefüllte Rechtecke repräsentiert werden. Leiten Sie die Grafikklassen direkt oder indirekt von einer Klasse `GrafikElement` ab. Verwenden Sie, wo sinnvoll, `protected`-Komponenten und/oder Methoden.



Die Klassen sollen neben den geerbten Eigenschaften folgende Informationen enthalten:

`GrafikElement`	Bezugspunkt: `int xs`, `int ys`.
`Punkt`	Farbe: `RGB_Pixel c`.
`Linie`	Endpunkt: `int xe`, `int ye`, Farbe: `RGB_Pixel c`.
`Kreis`	Radius `int radius`, Farbe: `RGB_Pixel c`.
`Rechteck`	Endpunkt: `int xe`, `int ye`, Farbe: `RGB_Pixel c`.
`Zeichen`	Code des Zeichens: `unsigned char ch`, Vordergrundfarbe: `RGB_Pixel fg`, Hintergrundfarbe: `RGB_Pixel bg`.
`Textzeile`	Zeichenkette: `string s`, Vordergrundfarbe: `RGB_Pixel fg`, Hintergrundfarbe: `RGB_Pixel bg`.
`RechteckGefuellt`	Füllfarbe: `RGB_Pixel f`.

Für die Erstellung der Klassenhierarchie gelten folgende Vorgaben:

- Jede Klasse soll einen Default-Konstruktor erhalten.
- Ein weiterer Konstruktor soll die Objekte einer Klasse gemäß Vorgaben initialisieren und dazu auch den Konstruktor der Basisklasse verwenden. Die folgenden Objektdefinitionen zeigen die Reihenfolgen der Konstruktorparameter, die in den Klassen einzuhalten sind:

```
int xs(10), ys(20), xe(37), ye(39), rk(40);
RGB_Color c(255,0,0),f(0,255,0),fg(0,0,255),bg(0,0,0);
unsigned char('\x94');
string s("Hello World");

Punkt           p(xs,ys, c);
Linie           l(xs,ys, xe,ye, c);
Kreis           k(xs,ys, rk, c);
Rechteck        r(xs,ys, xe,ye, c);
Zeichen         z(xs,ys, ch, fg,bg);
Textzeile       t(xs,ys, s, fg,bg);
RechteckGefuellt rg(xs,ys, xe,ye, c,f);
```

- Erstellen Sie für jede Grafikklasse eine virtuelle Methode `void draw(Image&)`, mit der das Grafikobjekt in ein `Image`-Objekt (Siehe Praktikumsübung 4) eingeschrieben werden kann. In der Basisklasse `GrafikElement` soll die `draw` -Methode rein virtuell vereinbart werden.

Mindestpunktzahl für die 6. Praktikumsübung: 5 Punkte

Die `draw`-Methoden der einzelnen Klassen sollen folgende Funktionalität ausführen:

`Punkt`	Punkt an der Position des Bezugspunkts mit der spezifizierten Farbe zeichnen.
`Linie`	Linie beliebiger Steigung vom Bezugspunkt zum Endpunkt in der spezifizierten Farbe zeichnen.
`Kreis`	Kreis um den Bezugspunkt mit dem vorgegebenen Radius in der spezifizierten Farbe zeichnen.
`Rechteck`	Rechteck mit dem Bezugspunkt als eine Ecke und dem Endpunkt als gegenüberliegende Ecke in der spezifizierten Farbe zeichnen.
`Zeichen`	Zeichen mit der spezifizierten Hinter- und Vordergrundfarbe zeichnen. Der Bezugspunkt ist unten links.
`Textzeile`	Zeichenkette aus Zeichen mit der spezifizierten Hinter- und Vordergrundfarbe zeichnen. Der Bezugspunkt ist unten links.
`RechteckGefuellt`	Rechteck wie oben, aber im Inneren gefüllt mit der spezifizierten Füllfarbe, zeichnen.

- In der Basisklasse `GrafikElement` soll der Zuweisungsoperator
  `GrafikElement& operator=(const GrafikElement& r)`
  rein virtuell vereinbart und in den Unterklassen virtuell überladen werden.
  In den Zuweisungsoperatoren der Unterklassen muss überprüft werden, ob das übergebene Objekt `r` der rechten Seite vom Typ der aktuellen Unterklasse ist. Falls nicht, soll eine Ausnahme geworfen werden.
- In der Basisklasse wird eine virtuelle Methode `void add_offset(int,int)` vereinbart und in allen Unterklassen geeignet unterstützt.
  Diese Methode verschiebt den Bezugspunkt eines Objekts um einen angegebenen x,y-Offset, so dass Objekte mit der gleichen Form an verschobener Position dargestellt werden.
- In der Basisklasse wird eine rein virtuelle, konstante Methode `GrafikElement* clone()` vereinbart und in allen Unterklassen unterstützt, mit der eine Kopie des wirklichen, aktuellen Objekts erstellt werden kann.
- Das Zerstören eines Objektes mittels `delete` mit einem `GrafikElement*`-Zeiger soll das wirkliche Objekt korrekt zerstören.

Hinweis: Für das Zeichnen von Linien und Kreisen sind in `Bresenham.cxx` zugehörige Bresenham-Algorithmen gezeigt. Siehe z.B. *http://de.wikipedia.org/wiki/Bresenham-Algorithmus* zur Erläuterung der Algorithmen.

Hinweis: Verwenden Sie beim Zeichnen von Zeichen und Textzeilen zur Umwandlung der Zeichencodes in Rasterdarstellung die Klasse `SimpleFont` aus der letzen Praktikumsübung.

Zum Testen Ihrer Grafik-Klasse steht das Testprogramm `P6_Test.cxx` zur Verfügung. Das Programm verwendet die bekannten Klassen `RGB_Pixel`, `Image` und `BmpWrite`.

☞ **Frage:** Wo werden die Methoden und Operatoren der von Ihnen erstellten Grafik-Klassen im Testprogramm verwendet?