

 <p>bics@LISUM Landesinstitut für Schule und Medien Berlin</p>	<h2 style="text-align: center;">OOP von Anfang an</h2> <p style="text-align: center;">Alexander Dietz, Bernd Kokavec 26. Oktober 2005 Humboldtschule Berlin-Tegel</p>	
----------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

Grafik-Tools

Es werden bei uns unterschiedliche Versionen von Grafiktools im Anfängerunterricht verwendet. Unabhängig vom vorgestellten Beispiel liegen vor:

Für Python:

```

Bearbeiter:      Bernd Kokavec
Datei:           grafik.py
-----
import Tkinter
from Tkconstants import *
import Canvas
-----
class Image:
    /* Bildklasse
--
    def __init__(self,Name):
        /* Name: string : Dateiname des Bildes
--
    def get_Bild(self):
        /* liefert das Bildobjekt für 'image' in Canvas.ImageItem
        /* (Darstellung des Bildes auf einer Zeichenfläche)
--
    def get_Breite(self):
        /* liefert die Bildbreite in Pixeln
--
    def get_Hoehe(self):
        /* liefert die Bildhoehe in Pixeln
--
class TColor:
    /* erste primitive Version mit nur wenigen Farben
    /* Die Farben können über die deutschen Namen oder über
    /* Zahlen abgerufen werden
    /* Hinweis:  "#fff" entspricht "weiss",
    /*          statt "#xxx" kann auch "red", "green" usw. benutzt werden
    /* andere Lösungen mit true-colors sind denkbar
--
    def __init__(self):
        /* transparent, schwarz, blau, gruen, tuerkis, rot, gelb, grau, weiss
--
    def getColor(self,nr):
        /* nr : int : 0 .. 8 für die oben angegebenen Farben
        /* liefert die Farbdarstellung für X
--
    def getFarbnamen(self,nr):
        /* nr : int : 0 .. 8 für die oben angegebenen Farben
        /* liefert den (deutschen) Bezeichner der Farbnummer (s.o.)
--
    def getFarbe(self,wort):
        /* wort : string : ein Element aus den oben angegebenen Farben
        /* liefert die Farbdarstellung fuer X
--
class TFigur:
    /* interne Hinweise:
    /* ZF ist Referenz auf Zeichenfläche, wird später gesetzt

```

```

/* grafObj ist das aktuelle Grafikobjekt
--
def __init__(self):
    /* Alle Grafik-Klassen erben von TFigur. TFigur wird beschrieben durch
    /* folgende Attribute:
    /* X1,Y1 (linke obere Ecke)
    /* X2,Y2 (rechte untere Ecke)
    /* Farbe
    /* Fuellfarbe
--
def setPos(self,ax1,ay1,ax2,ay2):
    /* ax1,ay1 : int :(linke obere Ecke)
    /* ax2,ay2 : int :(rechte untere Ecke)
--
def getXPos(self):
    /* liefert x-Wert der Position der linken oberen Ecke
--
def getYPos(self):
    /* liefert y-Wert der Position der linken oberen Ecke
--
def setFarbe(self,F):
    /* F : string : deutscher Bezeichner (s.o.)
--
def getFarbe(self):
    /* gibt akt. Farbe zurück : string : Farbrepr. für X
--
def setFuellfarbe(self,F):
    /* F : string : deutscher Bezeichner (s.o.)
--
def getFuellfarbe(self):
    /* gibt akt. Füllfarbe zurück : string : Farbrepr. für X
--
def pos_versetzen_um(self,dx,dy):
    /* versetzt die Position des heweiligen Grafikobjektes um dx und dy
--
def zeigen(self):
    /* zeigt das Grafikobjekt auf dem Schirm an
--
def loeschen(self):
    /* löscht das Grafikobjekt auf dem Schirm
--
def entfernen(self):
    /* entfernt das Grafikobjekt aus dem Speicher
--
class TLinie(TFigur):
    /* Klasse Linie
--
class TEllipse(TFigur):
    /* Klasse Ellipse
--
class TKreis(TFigur):
    /* Klasse Kreis
--
def __init__(self):
    /* zus. Attribute sind hier: Radius, x-Mittelpunkt, y-Mittelpunkt
--
def setRadius(self,r):
    /* r : int : Radius
--
def getRadius(self):
    /* liefert aktuelle Radiuslänge
--
def setMPos(self,ax,ay):
    /* ax, ay : int
    /* setzt Mittelpunktskoordinaten
--
class TRchteck(TFigur):
    /* Klasse Rechteck
--

```

```

class TText(TFigur):
    /* Klasse Text zur Beschriftung der Zeichenfläche
--
    def __init__(self):
        /* Attribute sind
        /* Text : string
        /* Schriftart : String (X-Fonts-Bezeichner)
        /* Zeichen-Hoehe : int : default = 10
--
    def setPos(self,ax,ay):
        /* ax, ay : int : Position des ersten Zeichens
--
    def setText(self,Text):
        /* Text : string : auszugebender Text
--
    def setFont(self,Art="*",Grad=10):
        /* Art : string : Font-Name
        /* Grad : int : Zeichengröße
--
class TZeichenblatt:
    /* Zeichenblatt entspricht Canvas. Mit Init wird ein Bild unterlegt
--
    def Init(self,Name):
        /* Init hinterlegt das Bild
        /* Name : string : Dateiname (gif)
--
    def get_Breite(self):
        /* liefert die Bildbreite in Pixeln
--
    def get_Hoehe(self):
        /* liefert die Bildhoehe in Pixeln

```

	<h2 style="text-align: center;">OOP von Anfang an</h2> <p style="text-align: center;">Alexander Dietz, Bernd Kokavec 26. Oktober 2005 Humboldtschule Berlin-Tegel</p>	
-----------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

Für Java:

Die Klasse Zeichenblatt (Datei Zeichenblatt.java)

öffnet ein Fenster, dessen Größe vom verwendeten Hintergrundbild abhängt. Es beinhaltet die Methode zum Schließen des Fensters [X], stellt eine Pausenfunktion (anzugeben in ms) und Methoden zum Einfügen weiterer Grafiken in das Fenster bereit.

Zeichenblatt beerbt Frame und besitzt eine **paint-Methode**, die ggf. überschrieben werden kann, wenn man Zeichenblatt beerbt.

Innerhalb der Grafikumgebung können die üblichen Grafikmethoden (drawOval, fillRect usw.) verwendet werden. Es kann Schrift im Grafikmodus ausgegeben und die Farben gesteuert werden.

```
public class Zeichenblatt extends Frame {

    public Zeichenblatt(String bildname)

    public void pause(int ms)

    public void paint(Graphics page)

    public Image getImage(String name)    // für zeichne Bild

    public void zeichneBild(Image bild, int x, int y)

    public boolean handleEvent(Event ev)    // der Fenster-Zu-Knopp

    // handleEvent METHODE (geerbte System-Methode (AWT) wird hier überschrieben)
```

Gezeichnet wird bei diesem Tool mit den Standard-Java-Methoden (s. Anlage)