

Vom Haus zum Dorf bzw. zum Windpark

- ① Will man ein Dorf aus mehreren (im Prinzip gleichartigen) Häusern aufbauen, empfiehlt es sich, zunächst den Bauplan (=die Klasse) für ein Haus zu erstellen. Dieses Haus darf aber keinen eigenen Bildschirm haben, weil fürs ganze

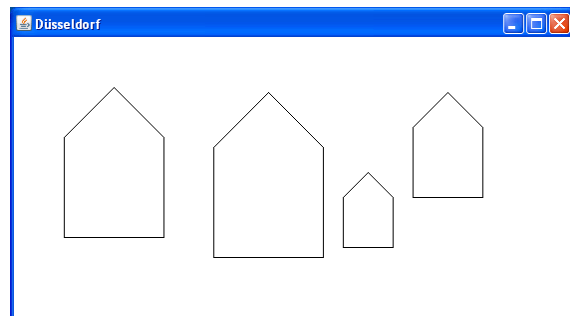
```
import stiftUndCo.*;

public class D_Haus
{
    BuntStift s = new BuntStift();

    public D_Haus (int x, int y, int kante)
    {
        s.bewegeBis (x,y); // linke untere Ecke
        s.runter();
        s.bewegeBis (x+kante, y);
        s.bewegeBis (x+kante, y-kante);
        s.bewegeBis (x+0.5*kante, y-1.5*kante);
        s.bewegeBis (x, y-kante);
        s.bewegeBis (x,y);
    }
}
```

Dorf später mehrere Häuser

auf dem gleichen Bildschirm erscheinen sollen. Außerdem darf die Position und die Größe des Hauses nicht mit konkreten Zahlen (absoluten Werten) festgelegt werden, sondern muss insofern variabel sein, als beim Erzeugen eines weiteren Hauses nach dem gleichen Bauplan eine andere Position (z.B. der linken unteren Ecke) und eine andere Kantenlänge möglich sein muss. Die übrigen Ecken des Hauses müssen relativ zur übergebenen linken unteren Ecke bestimmt werden. Ein Haus macht nur Sinn, wenn man es auch sieht. Deshalb soll es beim Erzeugen sofort gezeichnet werden (und nicht erst warten, bis eine eigene *zeichne*-Methode aufgerufen wird): dazu wird der Konstruktor verwendet – eine Methode mit gleichem Namen wie die Klasse, ohne *void* – die, sofern vorhanden, automatisch beim Erzeugen (d.h. bei *new D_Haus()*) aufgerufen und ausgeführt wird.



Ist der Bauplan fertig, kann man das Haus nicht direkt sehen/testen: Es fehlt die *main*-Methode und ein Bildschirm, auf dem der Stift *s* des Hauses zeichnen kann. Eine kleine Extra-Datei *D_HausTest* hilft. Hier wird ein Bildschirm-Objekt *b* und darauf ein spezielles Haus mit Namen *test* und den angegebenen Maßen erzeugt. *test* ist ein Objekt der Klasse *D_Haus*. Diese Testdatei wird nur während der Entwicklung benötigt, wenn man das Haus sehen will, bevor man das ganze Dorf fertig stellt.

```
import stiftUndCo.*;

public class D_HausTest
{
    public static void main (String[] s)
    {
        Bildschirm b = new Bildschirm();
        D_Haus test = new D_Haus (50,200,100);
    }
}
```

```
import stiftUndCo.*;

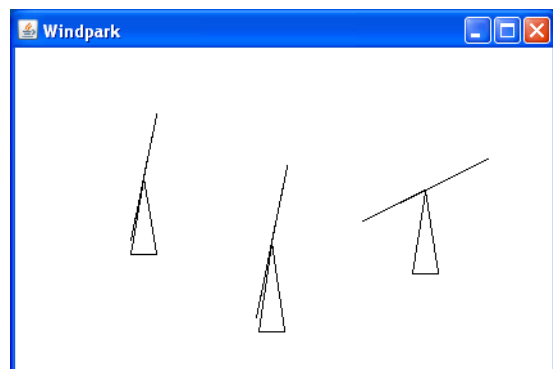
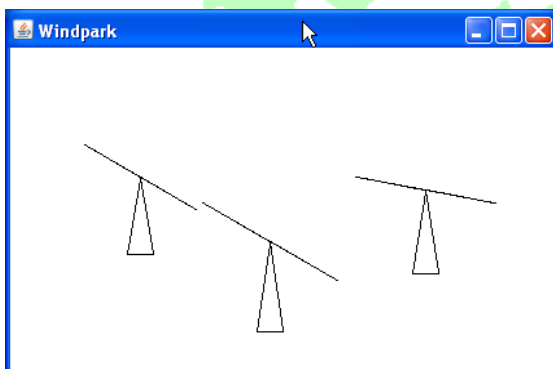
public class D_Dorf
{
    Bildschirm gelände = new Bildschirm(560,280,"Düsseldorf");
    D_Haus haus1 = new D_Haus (50, 200, 100);
    D_Haus haus2 = new D_Haus (200, 220, 110);
    D_Haus haus3 = new D_Haus (400, 160, 70);
    D_Haus haus4 = new D_Haus (330, 210, 50);
}
```

Das Dorf besteht aus einem eigenen Bildschirmfenster, auf/in dem die Häuser aufgebaut werden (dem *gelände*, einem Objekt nach dem Bauplan der von *Stift&Co* bereit gestellten Klasse *Bildschirm*) und vier Hausobjekten. Eine Methode ist nicht nötig, da die Häuser beim Erzeugen direkt gezeichnet werden (s.o.!).

Allerdings kann man das Dorf noch nicht sehen: *D_Dorf* ist lediglich der Bauplan für das Dorf, noch nicht das Dorf selbst. Es muss erst noch ein konkretes Objekt (hier *düsseldorf*) nach diesem Bauplan erzeugt werden. Dazu dient die Startdatei. Weil in der Startdatei Begriffe wie *Bildschirm* oder *BuntStift* nicht ausdrücklich stehen, braucht man hier kein *import stiftUndCo.**. Ein Methodenaufruf (wie etwa *düsseldorf.zeichneAlles()*) ist auch nicht nötig, weil im Bauplan des Dorfes keine Methode steht/nötig war!

```
public class D_DorfStart
{
    public static void main (String[] s)
    {
        D_Dorf düsseldorf = new D_Dorf ();
    }
}
```

- 2 Überlege und begründe:
- Wie viele Buntstifte werden benutzt, um das ganze Dorf zu zeichnen? Wie heißen diese Stifte? Wieso kommt der Computer nicht durcheinander, sondern weiß genau, was er mit welchem Stift zeichnen muss? (Verwende in diesem Zusammenhang den Begriff Kapselung!)
 - Werden alle vier hier gezeigten Baupläne benutzt, wenn die Startdatei gestartet wird? Werden manche Baupläne mehrfach genutzt? Und gibt es Baupläne, die ebenfalls benutzt werden, ohne dass wir sie hier schreiben mussten?
 - Baupläne haben in der Regel zwei Abteilungen: in der ersten Abteilung stehen die Attribute (sozusagen die Materialliste, nämlich die dauerhaft verwendeten Objekte und Variablen) und dann folgen in der zweiten Abteilung eine oder mehrere Methoden (Tätigkeitsbeschreibungen; praktisch das Rezept -- meist bestehend aus Kontrollstrukturen und Anweisungen). Prüfe, ob die vier gezeigten Baupläne immer beide Abteilungen aufweisen.
 - Häuser sollen nur gezeichnet werden, wenn sie mindestens eine Kantenlänge von 30 Pixeln haben. Ändere die richtige Klasse entsprechend ab!
- 3 Unser nächstes Projekt soll nicht nur starr und unverändert angezeigte Bauwerke enthalten, sondern Windmühlen mit rotierenden Flügeln – hier zwei verschiedene Momentbilder:



Entwickle geeignete Baupläne zum stufenweisen Aufbau des Windparks.

Tipp: Ein Strich kann mit einem Buntstift-Objekt *stift* nicht nur zwischen zwei Punkten (x_1, y_1) und (x_2, y_2) mit *stift.bewegeBis (x1,y1); stift.bewegeBis (x2,y2)*; auf den Bildschirm gezeichnet werden. Auch ohne Angabe eines Endpunktes kann ein Strecke von 50 Pixeln Länge mit *stift.bewegeUm (50)*; in die letzte eingestellte Richtung gezeichnet werden. Ein neuer Buntstift zeichnet ohne besondere Angaben nach rechts; mit *stift.dreheUm (10)*; kann aber seine Richtung z.B. um 10° gegenüber der letzten Richtung im Gegenuhrzeigersinn verändert werden. Dabei sind auch negative Angaben wie etwa *stift.bewegeUm (-40)* oder *stift.dreheUm (-10)* möglich!