

turmVerschieben(Anzahl n, Stapel quelle, Stapel hilf, stapel ziel):

start

wenn  $n > 1$  dann

turmVerschieben(Anzahl  $n-1$ ,  
Stapel quelle, Stapel ziel, Stapel hilf)

wenn\_ende

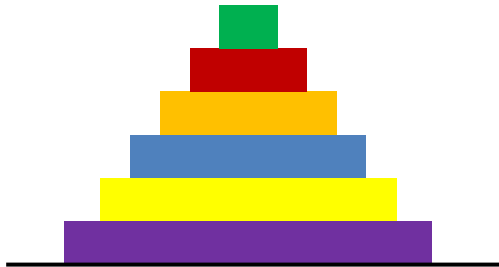
Versetze Scheibe von quelle nach ziel

wenn  $n > 1$  dann

turmVerschieben(Anzahl  $n-1$ ,  
Stapel hilf, Stapel quelle, Stapel ziel)

wenn\_ende

ende



Quelle



Ziel



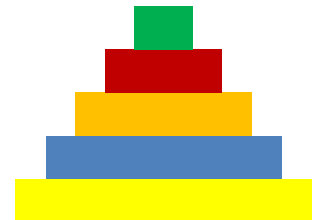
Hilf



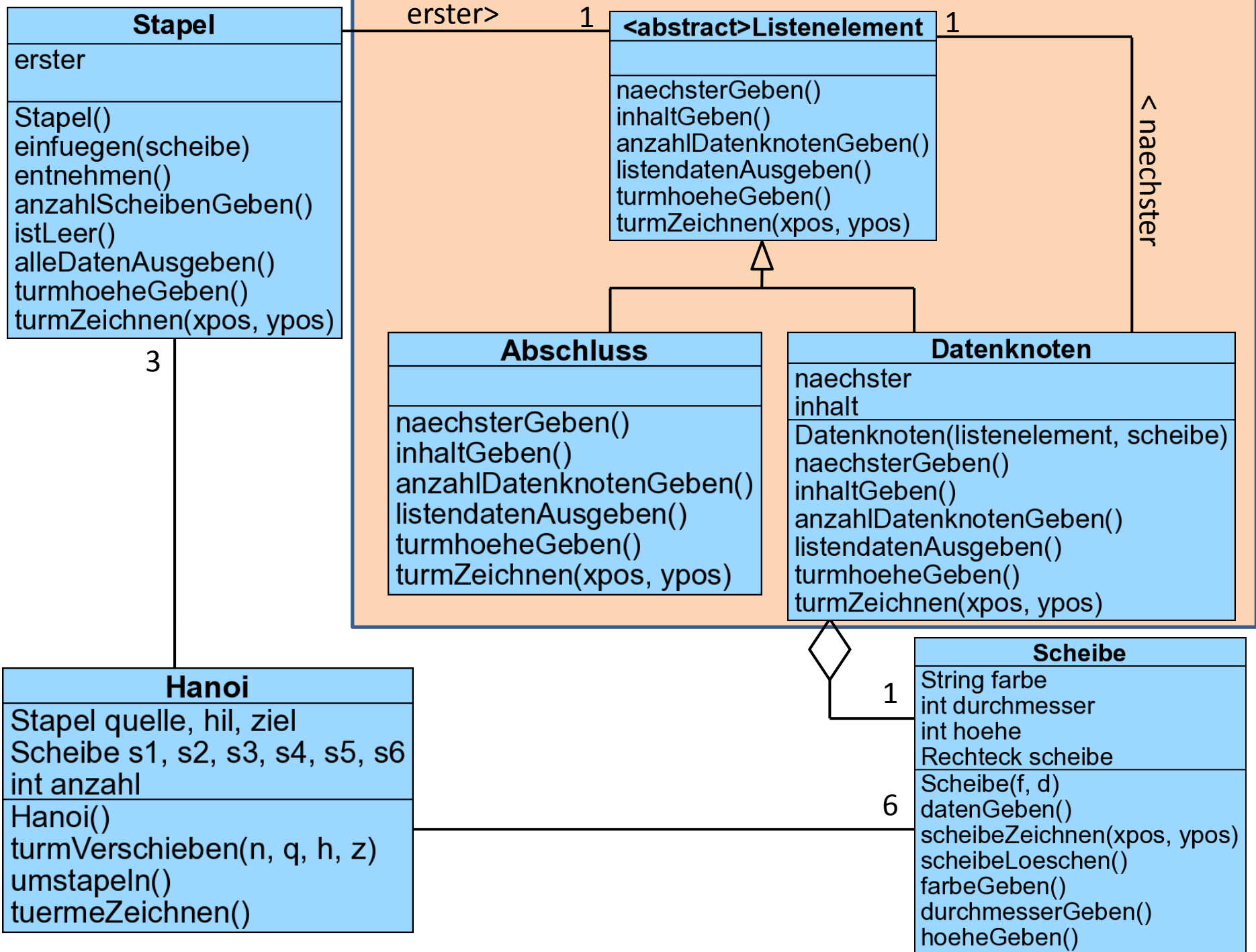
Quelle



Ziel



Hilf



```
public class Scheibe {
    private String farbe;
    private int durchmesser;
    private int hoehe;
    Rechteck scheibe;

    public Scheibe(String f, int d){
        farbe = f;
        durchmesser = d;
        hoehe =15;
        scheibe = new Rechteck();
    }

    public String datenGeben(){
        return "Farbe: "+ farbe + " Durchmesser: " + durchmesser;
    }

    public void scheibeZeichnen(int xpos,int ypos){
        scheibe.setzePosition(xpos, ypos);
        scheibe.setzeGroesse(durchmesser, hoehe);
        scheibe.setzeFarbe(farbe);
        scheibe.sichtbarMachen();
    }
}
```

...

```
public class Stapel{
    ...
    public void einfuegen(Scheibe s){
        Datenknoten neuerKnoten = new Datenknoten(erster, s);
        erster = neuerKnoten;
    }

    public Scheibe entnehmen(){
        Scheibe aktuelleScheibe = erster.inhaltGeben();
        erster = erster.naechsterGeben();
        return aktuelleScheibe;
    }
    ...
    public int turmhoeheGeben(){
        return erster.turmhoeheGeben();
    }

    public void turmZeichnen(int xpos, int ypos){
        erster.turmZeichnen(xpos, ypos-turmhoeheGeben());
    }
}
```

```
public Hanoi(){  
    anzahl= 6;  
  
    s1 = new Scheibe("rot", 120);  
    s2 = new Scheibe("gruen", 100);  
    s3 = new Scheibe("cyan", 80);  
    s4 = new Scheibe("orange", 60);  
    s5 = new Scheibe("blau", 40);  
    s6 = new Scheibe("gelb", 20);  
  
    quelle = new Stapel("Quelle");  
    quelle.einfuegen(s1);  
    quelle.einfuegen(s2);  
    quelle.einfuegen(s3);  
    quelle.einfuegen(s4);  
    quelle.einfuegen(s5);  
    quelle.einfuegen(s6);  
  
    hilf = new Stapel("Hilf");  
    ziel = new Stapel("Ziel");  
    tuermeZeichnen();  
}
```

```
public void tuermeZeichnen(){  
    quelle.turmZeichnen(100, 150);  
    hilf.turmZeichnen(250, 150);  
    ziel.turmZeichnen(400, 150);  
}
```



```
public void turmVerschieben(int n, Stapel q, Stapel h, Stapel z){  
    if (n>1) {turmVerschieben(n-1, q, z, h);}
  
    Scheibe s =q.entnehmen();  
    z.einfuegen(s);
  
    if (n>1) {turmVerschieben(n-1, h, q, z);}
}
```

Füge an geeigneter Stelle eine Ausgabe wie folgt ein:  
"Farbe rot, Durchmesser 20 von Quelle nach Hilf"  
Erweitere dazu die Klasse Stapel um ein Attribut name.

```
public void turmVerschieben(int n, Stapel q, Stapel h, Stapel z){  
    if (n>1) {turmVerschieben(n-1, q, z, h);}  
    Scheibe s =q.entnehmen();  
    System.out.println( s.datenGeben() + " von "  
                        + q.nameGeben()+ " nach " + z.nameGeben());  
    z.einfuegen(s);  
  
    if (n>1) {turmVerschieben(n-1, h, q, z);}  
}
```

Scheibe löschen und Türme zeichnen lassen!

```

public void turmVerschieben(int n, Stapel q, Stapel h, Stapel z){
    if (n>1) {turmVerschieben(n-1, q, z, h);}
    Scheibe s =q.entnehmen();
    System.out.println(s.datenGeben() + " von " + q.nameGeben()+
        " nach " + z.nameGeben());
    s.scheibeLoeschen();
    z.einfuegen(s);
    tuermeZeichnen();
    ← Hier noch Warten einfügen!
    if (n>1) {turmVerschieben(n-1, h, q, z);}
    try { Thread.sleep(300);}
    catch (InterruptedException e){}
}

```

- Zähler einbauen
- Auf mehr bzw. variable Scheibenzahl erweitern
- Manuelles Scheiben verschieben

## Anzahl der Züge:

Bei einer Scheibe:  $2^1 - 1 = 1$  Zug

Bei zwei Scheiben:  $2^2 - 1 = 3$  Züge

Bei drei Scheiben:  $2^3 - 1 = 7$  Züge

Bei n Scheiben:  $2^n - 1$  Züge

Behauptung:

Zum Umschichten eines Turmes mit  $n$  Scheiben sind  $2^n - 1$  Züge nötig.

Beweis durch Induktion nach  $n$ :

- $n = 1$ : 1 Scheibe von Quelle nach Ziel  $\rightarrow$  1 Zug

- Annahme:

Die Behauptung gilt für alle natürlichen Zahlen von 1 bis  $n-1$ .

Zeige, dass sie dann auch für  $n$  gilt.

Zum Umschichten der oberen  $n-1$  Scheiben von Quelle nach Hilf sind entsprechend der Annahme  $2^{n-1} - 1$  Züge nötig.

Zum Umlegen der größten Scheibe von Quelle nach Ziel ist ein Zug nötig.

Zum Umschichten der oberen  $n-1$  Scheiben von Hilf nach Ziel sind  $2^{n-1} - 1$  Züge nötig.

Insgesamt:  $2^{n-1} - 1 + 1 + 2^{n-1} - 1 = 2 \cdot 2^{n-1} - 1 = 2^n - 1$

Aufrufsequenz für turmVerschieben (tV):

$$\begin{aligned} \text{tV}(2, q, h, z) &= \text{tV}(1, q, z, h); \text{umlegen}(q, z); \text{tV}(1, h, q, z) \\ &= \text{umlegen}(q, h); \text{umlegen}(q, z); \text{umlegen}(h, z) \end{aligned}$$

$$\begin{aligned} \text{tV}(3, q, h, z) &= \text{tV}(2, q, z, h); \text{umlegen}(q, z); \text{tV}(2, h, q, z) \\ &= \text{tV}(1, q, h, z); \text{umlegen}(q, h); \text{tV}(1, z, q, h); \\ &\quad \text{umlegen}(q, z); \\ &\quad \text{tV}(1, h, z, q); \text{umlegen}(h, z); \text{tV}(1, q, h, z); \\ &= \text{umlegen}(q, z); \text{umlegen}(q, h); \text{umlegen}(z, h); \\ &\quad \text{umlegen}(q, z); \\ &\quad \text{umlegen}(h, q); \text{umlegen}(h, z); \text{umlegen}(q, z); \end{aligned}$$