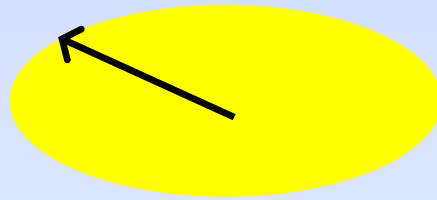
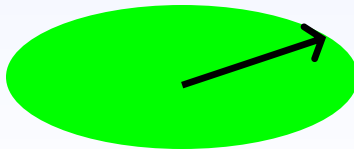


Kreise

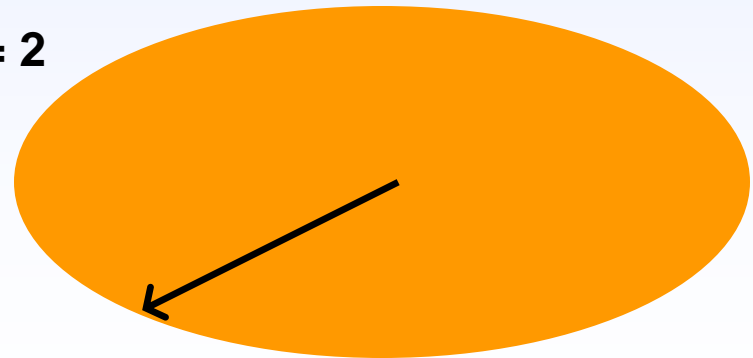


Radius $r = 1,5$

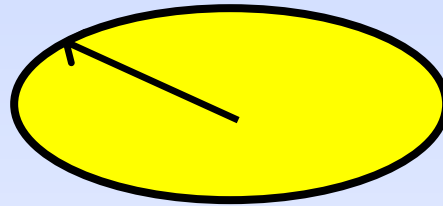
Radius $r = 1$



Radius $r = 2$

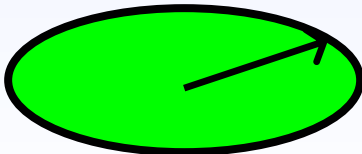


Kreise: Umfang $U=2\pi r$

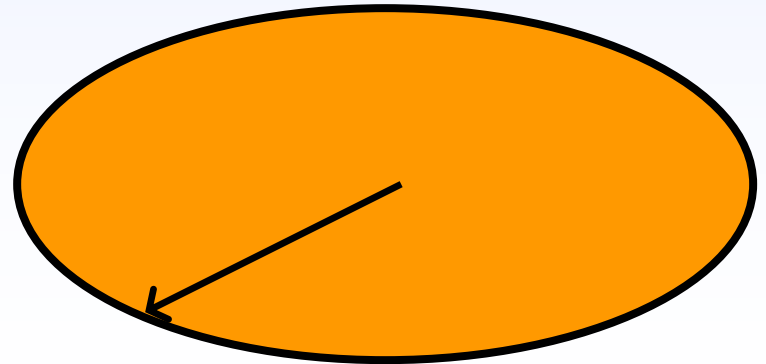


Radius $r = 1,5$
Umfang $U = 9,42$

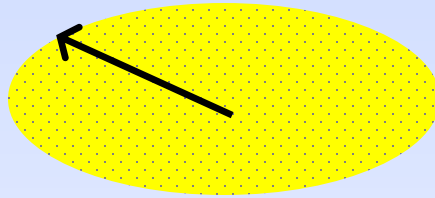
Radius $r = 1$
Umfang $U = 6,28$



Radius $r = 2$
Umfang $U = 12,56$

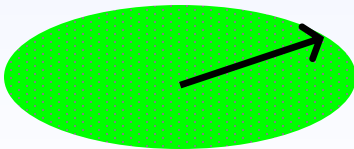


Kreise: Fläche $F=r^2 \pi$

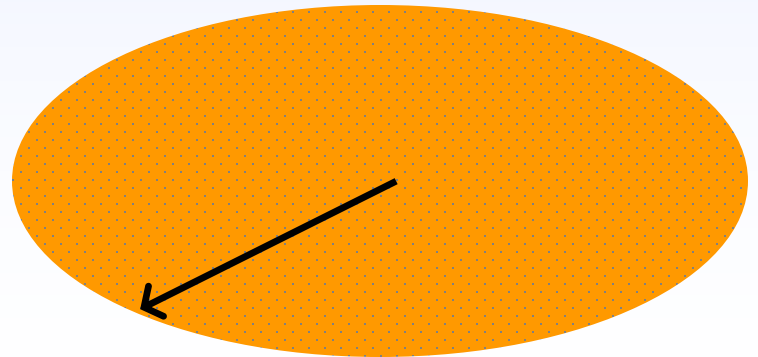


Radius $r = 1,5$
Fläche $F = 7,07$

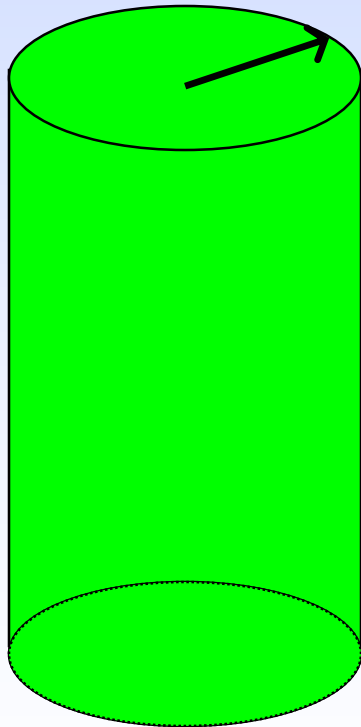
Radius $r = 1$
Fläche $F = 3,14$



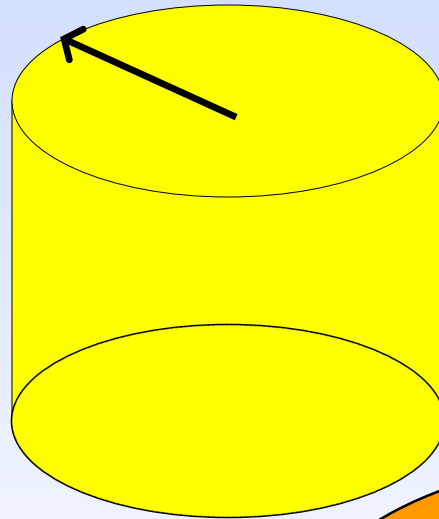
Radius $r = 2$
Fläche $F = 12,56$



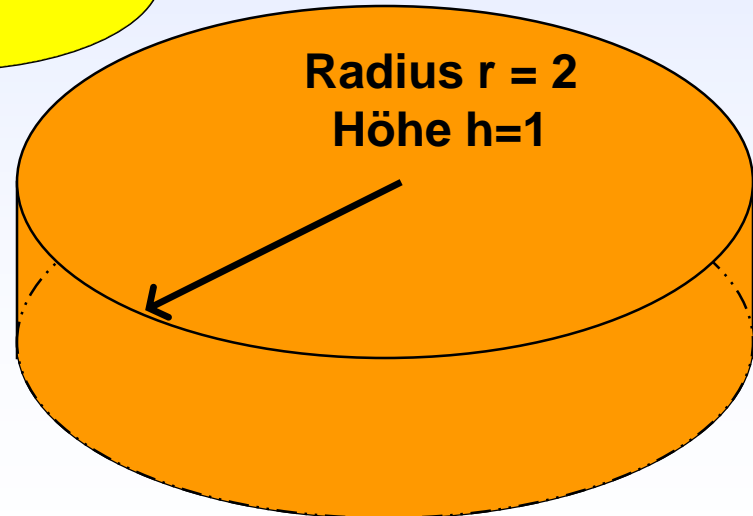
Zylinder: Oberfläche ist $Uh+2F$



Radius $r = 1$
Höhe $h=3$



Radius $r = 1,5$
Höhe $h=2$



Radius $r = 2$
Höhe $h=1$

Programmcode Zylinder

```
class Kreis {  
    ...  
    public double flaeche() {return 3.14*r*r}  
}  
class Zylinder extends Kreis {  
    private double hoehe;  
    ...  
    public double flaeche() {  
        double fl , umf;  
        umf = super.umfang();  
        fl = super.flaeche();  
        return 2*fl + hoehe*umf;  
    }  
}
```

Dynamische Bindung zur Programmlaufzeit 1/2

```
public class ZylinderTest {  
    public static void main(String args[]) {  
        Kreis k[] = new Kreis[10];  
  
        for(int i=0 ; i<=9 ; i++) {  
            if (Math.random() <= 0.5) {  
                k[i] = new Kreis(i+0.5);  
            } else {  
                k[i] = new Zylinder();  
                k[i].setRadius(i+1);  
                ( (Zylinder) k[i]).setHoehe(i+1.5);  
            }  
        }  
        ...  
    }  
}
```

Dynamische Bindung zur Programmlaufzeit 2/2

```
...
System.out.println(
    k[i].getClass().getName() +
    " k[" + i + "]" +
    " mit Radius " + k[i].r +
    " und Flaeche " +
    k[i].flaeche()
);
}
}
}
```

Zusammenfassung

- Dynamische Bindung:
Java ist auch eine Insel Kap. 5.11
- Abstrakte Klassen und abstrakte Methoden:
Java ist auch eine Insel Kap. 5.12