# Getter- u setter-Methoden

Bisher haben wir uneingeschränkt von überall lesend u. schreibend direkt auf die Attribute zugegriffen.

Dies kann zu schwer findbaren Fehlern führen, wenn in den Attributen plötzlich inkonsistente Werte auftreten,  
da kaum nachvollziehbar ist, von wo aus sie geschrieben wurden.

Deshalb wollen wir den Zugriff einschränken, keinen direkten Zugriff von überall mehr zulassen,  
sondern einen "Umweg erzwingen", bei dem wir Regeln aufstellen können.

Z.B. wollen wir verhindern, dass Seitenlängen von Rechteck oder Dreieck oder der Radius eines Kreises negativ wird.

class … {

direkter schreibender Zugriff

class Rechteck {

int laenge … ;

}

… main … {

Rechteck r1 = new Rechteck();

direkter lesender Zugriff

r1.laenge = …;

Sys…ln(r1.laenge);

}

}

Wir unterbinden den direkten lesenden u. schreibenden Zugriff,

indem wir vor das entsprechende Attribut das Schlüsselwort "private" schreiben.

Der Versuch, trotzdem (von außerhalb der Klasse) direkt zuzugreifen, wird mit einer Fehlermeldung verhindert.

class … {

class Rechteck {

**private** int laenge … ;

}

direkter schreibender Zugriff

… main … {

Rechteck r1 = new Rechteck();

direkter lesender Zugriff

r1.laenge = …;

Sys…ln(r1.laenge);

}

}

Wie greifen wir stattdessen indirekt zu?

Lesend über eine getter-Methode, schreibend über eine setter-Methode:

class … {

direkter schreibender Zugriff

class Rechteck {

**private** int laenge … ;

}

… main … {

Rechteck r1 = new Rechteck();

direkter lesender Zugriff

r1.laenge = …;

**in**direkter schreibender Zugriff

setter-Methode

Sys…ln(r1.laenge);

**in**direkter lesender Zugriff

getter-Methode

}

}