

---

**Programmieren in Java**<http://proglang.informatik.uni-freiburg.de/teaching/java/2017/>

---

**line-intersection***Schnittpunkt zweier Geraden*

Woche 05 Aufgabe 2/4

Herausgabe: 2017-05-22

Abgabe: 2017-06-02

**Achtung:** beachten Sie unbedingt die allgemeinen Hinweise zur Abgabe auf der Homepage.

Project `line-intersection`Package `lineintersection`

Klassen

Geometry
<pre>public static String lineIntersection(     double a1, double b1     double a2, double b2)</pre>
GeometryTest
<pre>@Test public void lineIntersection() public boolean checkIntersection(     double[] resultPoint,     double a1, double b1,     double a2, double b2,     double deltaX, double deltaY)</pre>

In dieser Aufgabe sollen für vorhandene Funktionen Unit-Tests mit JUnit4 geschrieben werden. Im Paket befindet sich ein vorkonfiguriertes IntelliJ-Projekt. Dort finden Sie die Klasse `lineintersection.Geometry`, die die Implementierung der folgenden Funktion enthält:

```
public static double[] lineIntersection(double a1, double b1, double a2,
                                       double b2)
```

Sie berechnet den Punkt, an dem sich zwei gegebene Geraden schneiden. Die Geraden ergeben sich aus den Gleichungen  $y = a1 * x + b1$  und  $y = a2 * x + b2$ . Das Ergebnis ist ein Array mit zwei Elementen, dass die x-Koordinate als erstes Element und die y-Koordinate als zweites Element enthält. Gibt es keinen Schnittpunkt, so soll `null` zurückgegeben werden.

Benutzen Sie JUnit4 um Tests für die Funktion zu schreiben. Die Tests müssen in der Klasse `lineintersection.GeometryTest` stehen. Außerdem sollen sie in den Tests ein so genanntes *Testorakel* verwenden. Ein Testorakel bestimmt, ob ein berechnetes Ergebnis zu einer gegebenen Eingabe korrekt ist. In dieser Aufgabe ist das Testorakel die oben genannte Funktion `lineintersection.GeometryTest.checkIntersection`. Sie soll `true` zurückgeben, genau dann wenn der Punkt `resultPoint` ein korrekter Schnittpunkt von für die

Geraden  $y = a1 * x + b1$  und  $y = a2 * x + b2$  ist. Da bei `double`-Werten leicht Rundungsfehler auftreten können, erlaubt das Orakel zusätzlich die Angabe der Toleranzwerte `deltaX` und `deltaY`. Die x-Koordinate von `resultPoint` ist demnach korrekt, wenn sie weniger als `deltaX` vom genauen Schnittpunkt abweicht. Analog dazu ist die y-Koordinate von `resultPoint` korrekt, wenn sie weniger als `deltaY` vom genauen Schnittpunkt abweicht.

Die Tests sollen das Orakel nach folgendem Muster verwenden:

---

```
1  assertTrue(checkIntersection(Geometry.lineIntersection(a1, b1, a2, b2),
2                               a1, b1, a2, b2, ...));
```

---

Wird dieses Muster nicht befolgt, gibt es keinen Punkt für Codequalität.

Ihre Testfälle müssen außerdem wieder so beschaffen sein, dass sie für die Methode `lineIntersection` Statement-Abdeckung herstellen. Das heisst, jedes Statement der Methoden muss durch mindestens einen Testfall ausgeführt werden. Sie dürfen hier wieder ausschließlich die Datei `test/lineintersection/GeometryTest.java` bearbeiten. Die Abdeckung kann wieder durch Ausführen von `test/lineintersection/Main.java` betrachten. Bei folgender Ausgabe ist Ihre Abdeckung in Ordnung:

```
Messages:
lineIntersection:intersect
lineIntersection:parallel
```