

Zwischenprüfung Frühjahr 2021



Aufgabenbogen

Mathematisch-technischer Softwareentwickler
Mathematisch-technische Softwareentwicklerin
6511

2

Objektorientierte Modelle und Algorithmen

4 Aufgaben
60 Minuten Prüfungszeit
100 Punkte

Es soll für ein Mathematik-Lernprogramm unter anderem ein Modul entwickelt werden, welches die Lage von Geraden im \mathbb{R}^3 überprüft (siehe Abbildung 2, Lage von Geraden).

Dazu wurden bereits von anderen Entwicklern einzelne Klassen entworfen, die als Klassendiagramm (siehe Abbildung 1, Klassendiagramm) vorliegen. So wurde bereits die Klasse Ebene entworfen, die durch einen Konstruktor aus drei gegebenen Punkten eine Ebene konstruiert und weitere Methoden, wie im Klassendiagramm angegeben, besitzt. Gehen Sie davon aus, dass der Richtungsvektor immer ungleich dem Nullvektor ist.

Sie erhalten von Ihrem Chef die Aufgabe, für die Klasse Gerade die Attribute und Methoden wie in den nachfolgenden Aufgaben beschrieben zu implementieren. Die im Klassendiagramm implementierten Attribute und Methoden stehen für die Aufgaben komplett zur Verfügung.

1. Aufgabe (25 Punkte)

Entwerfen Sie die Klasse Gerade als Klassendiagramm in OOD mit den folgenden Attributen und Methoden:

- Stützvektor und Richtungsvektor
- Parametrisierten Konstruktor mit den Parametern `Vektor stuetzvektor` und `Vektor richtungsvektor`
- Methode, die zum übergebenen Parameter $r \in \mathbb{R}$ den Punkt auf der Geraden $g : \vec{x} = \vec{p} + r \cdot \vec{v}$ berechnet und das Ergebnis zurückgibt.
- Eine Methode, die überprüft, ob ein Punkt auf der Geraden liegt.
- Eine Methode, die überprüft, ob eine als Parameter übergebene Gerade g parallel ist.
- Eine Methode, die überprüft, ob eine als Parameter übergebene Gerade g „echt parallel“ ist.
- Eine Methode zur Überprüfung, ob eine übergebene Gerade windschief zum aktuellen Objekt ist.

Eine Methode zur Berechnung eines Schnittpunktes zwischen zwei Geraden soll später noch realisiert werden und soll hier nicht im Klassendiagramm erwähnt werden.

2. Aufgabe (25 Punkte)

Entwerfen bzw. implementieren Sie die Methode

`istEchtParallel(g: Gerade): boolean`,

die überprüft, ob die übergebene Gerade zu der aktuellen Geraden echt parallel ist. Die Ungenauigkeit von Double kann hier ignoriert werden.

3. Aufgabe (25 Punkte)

Entwerfen bzw. implementieren Sie die Methode

`istWindschief(g: Gerade): boolean`,

die überprüft, ob die beiden Geraden windschief sind. Die Ungenauigkeit von Double kann hier ignoriert werden.

4. Aufgabe (25 Punkte)

Für die Überprüfung zweier Geraden auf Parallelität wird eine Methode der Klasse Vektor benötigt, die zwei Vektoren auf Vielfachheit überprüft. Erweitern Sie die Klasse Vektor so, dass der Vergleich zweier Vektoren auf Vielfachheit eine Abweichung von Epsilon zulässt. Implementieren Sie die Methode

`istVielfaches(v: Vektor, eps: double): boolean`.

Abbildung 1: Klassendiagramm

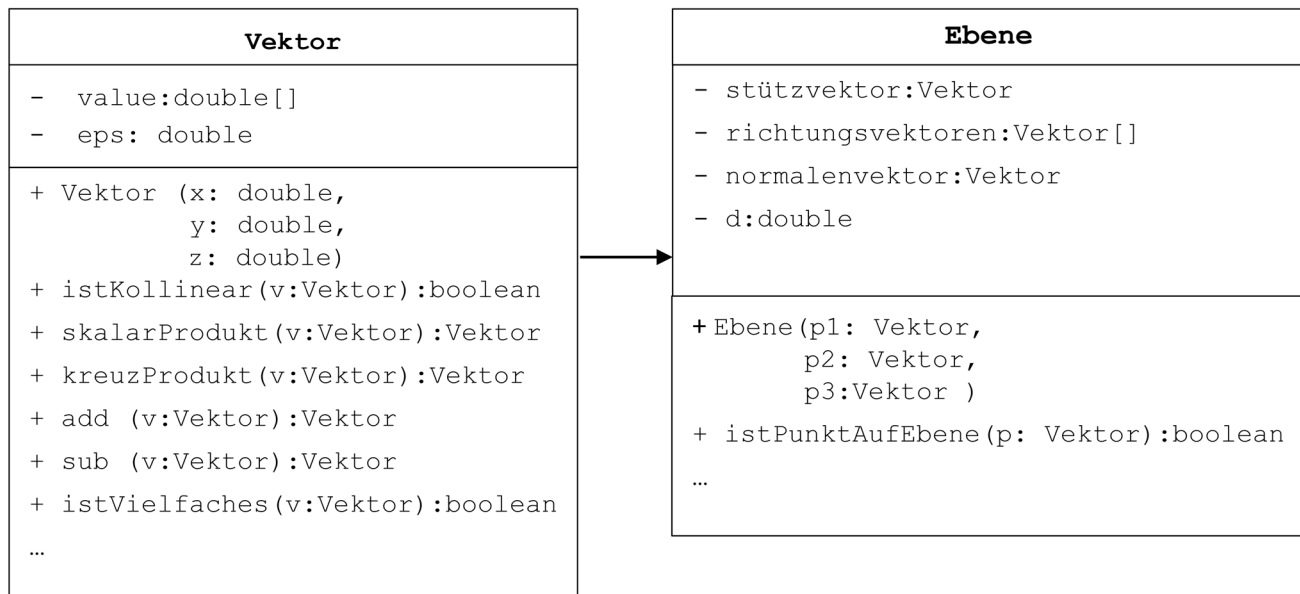


Abbildung 2: Lage von Geraden

Es sei: $g : \vec{x} = \vec{p} + r \cdot \vec{v}_1$ und $h : \vec{x} = \vec{q} + s \cdot \vec{v}_2$

