

Allgemeine Hinweise: Fügen Sie Kommentare hinzu, sobald Sie es für sinnvoll halten. Der Code auf dem main-Branch wird als Abgabe gewertet. Abgaben, bei denen die Kompilierung fehlschlägt, werden mit 0 Punkten bewertet. Wenn Sie bei einer Aufgabe nicht weiterkommen, fragen Sie gerne in den Tutorien oder in Teams nach. Die Designentscheidungen in den Übungsaufgaben entsprechen aus Gründen der Vereinfachung nicht immer denen, die man in der Realität antreffen würde.

1 3D-Vektoren (17 P)

Implementieren Sie in der Datei `src/main/scala/week03/geometry/Vector3D.scala` eine Klasse `Vector3D`, die einen unveränderlichen reellwertigen Vektor im dreidimensionalen Raum repräsentiert. Nutzen Sie die unveränderlichen Brüche aus der Vorlesung als Vorlage. Ein `Vector3D` speichert drei unveränderliche Variablen `x`, `y`, und `z` vom Typ `Double`.

Wenn `u` und `v` Variablen vom Typ `Vector3D` sind und `d` eine Variable vom Typ `Double` ist, sollen folgende Operationen möglich sein:

- (a) `u + v` (Vektoraddition)
- (b) `u - v` (Vektordifferenz)
- (c) `-u` (Vektornegation)
- (d) `u * d` (Skalarmultiplikation rechts)
- (e) `d * u` (Skalarmultiplikation links)
- (f) `u.dot(v)` berechnet das Standardskalarprodukt $\langle u, v \rangle$ zweier Vektoren als `Double`, das wie folgt definiert ist:

$$\langle u, v \rangle = u.x \cdot v.x + u.y \cdot v.y + u.z \cdot v.z$$

- (g) `u.cross(v)` berechnet das Kreuzprodukt $u \times v$ zweier Vektoren, das wie folgt definiert ist:

$$u \times v = \begin{pmatrix} u.y \cdot v.z - u.z \cdot v.y \\ u.z \cdot v.x - u.x \cdot v.z \\ u.x \cdot v.y - u.y \cdot v.x \end{pmatrix}$$

- (h) `u.length` gibt die euklidische Norm ("Länge") $|u|$ des Vektors `u` zurück, die wie folgt definiert ist:

$$|u| = \sqrt{u.x^2 + u.y^2 + u.z^2}$$

In Scala können Sie mit `math.sqrt(x)` die Quadratwurzel von `x` berechnen.

- (i) `u.normalized` gibt den Vektor zurück, der in dieselbe Richtung wie `u` zeigt, aber Länge 1 hat.
- (j) `u.isParallel(v)` gibt zurück, ob `u` und `v` parallel sind. `u` und `v` müssen dazu nicht in dieselbe Richtung zeigen.
- (k) `u.isOrthogonal(v)` gibt zurück, ob `u` senkrecht auf `v` steht.

Schreiben Sie außerdem einen sinnvollen `Scaladoc`-Kommentar für jede der Funktionen!

WICHTIG: Die Tests für diese Aufgabe wurden auskommentiert, um Ihnen den Einstieg in das Blatt zu erleichtern. Bitte entfernen Sie die Kommentarzeichen `/*` und `*/` vor der Abgabe aus den Tests.

2 Variablen vs Funktionen (4 P)

In Ihrer `Vector3D`-Implementierung können Sie `length` entweder mittels `val` als Variable oder mittels `def` als Methode implementieren. Beide Varianten haben Vor- und Nachteile.

- (a) Probieren Sie beide Varianten aus und überzeugen Sie sich davon, dass beide Varianten funktionieren.
- (b) Nennen Sie mindestens einen Vorteil und mindestens einen Nachteil der `val`-Variante.
- (c) Warum kommt es zu einem Laufzeitfehler, wenn man `normalized` als `val` deklariert?

Schreiben Sie Ihre Lösung als Kommentar unter den Code der `Vector3D`-Klasse.

3 Probleme durch Veränderlichkeit (4 P)

Schauen Sie sich das Programm in `src/main/scala/week03/rational/problem.scala` an. Dieses Programm nutzt die veränderlichen Brüche aus der letzten Vorlesung. Das Programm legt einen `ArrayBuffer` an (verhält sich wie eine Liste in Python) und fügt diesem dann zweimal den Bruch $\frac{2}{3}$ hinzu. Auf den ersten Bruch im Buffer wird anschließend 5 addiert. Bei der Ausgabe fällt allerdings auf, dass sich beide Brüche verändert haben.

- (a) Wie kann es passieren, dass auch der zweite Bruch im Buffer plötzlich einen anderen Wert hat?
- (b) Verändern Sie den Code so, dass die Addition nur mit dem ersten Bruch stattfindet.

Schreiben Sie Ihre Lösung als Kommentar in den Code.