

### **Implementierung des 3D-Vektors als Klasse oder als Struktur?**

Der 3D-Vektor ist als Struktur implementiert, weil Strukturen eine bessere Leistung bieten, wenn wir kleine Sammlungen von Werttypen haben, die wir gruppieren möchten. Da der 3D-Vektor aus drei Komponenten besteht, die gruppiert werden müssen, um ein neues Vektorobjekt zu instanzieren, wird struct als geeignet für den besonderen Anwendungsfall angesehen.

### **Soll der Vektor nach seiner Erzeugung unveränderlich sein oder sollen sich seine Komponenten nachträglich ändern lassen?**

Alle primitiven Datentypen wie z. B. int, double, string usw. in C# sind unveränderbar. Da ich einen neuen Datentyp erstelle, habe ich mich dafür entschieden, ihn mit den primitiven Datentypen konsistent zu machen, indem ich ihn unveränderlich mit nur lesendem Zugriff mache. Auf diese Weise ist es möglich zu verhindern, dass die Komponenten eines gegebenen Vektors verändert werden. Daher ist das Vektorobjekt in diesem Projekt unveränderlich und jedes Mal, wenn der Vektor eine definierte Vektoroperation durchläuft, wird ein resultierender Vektor als neues Vektorobjekt erzeugt.

### **In welcher Art und Weise erfolgt die Prüfung auf Gleichheit und Ungleichheit? Überladen Sie die entsprechenden Operatoren oder nicht?**

Zwei Vektoren werden als gleich angesehen, wenn sie gleich lang sind und die gleiche Richtung haben. Um die Gleichheit oder Ungleichheit der gegebenen Vektoren zu prüfen, überlade ich die entsprechenden Operatoren.

### **Wie wird der Hashcode eines Vektors bestimmt?**

Für jedes Vektorobjekt wird ein eindeutiger Hash-Code generiert, indem eine benutzerdefinierte Hash-Code-Generatormethode verwendet wird.