DN3

Zürcher Hochschule für Angewandte Wissenschaften

Arbeitsblatt: DNET1

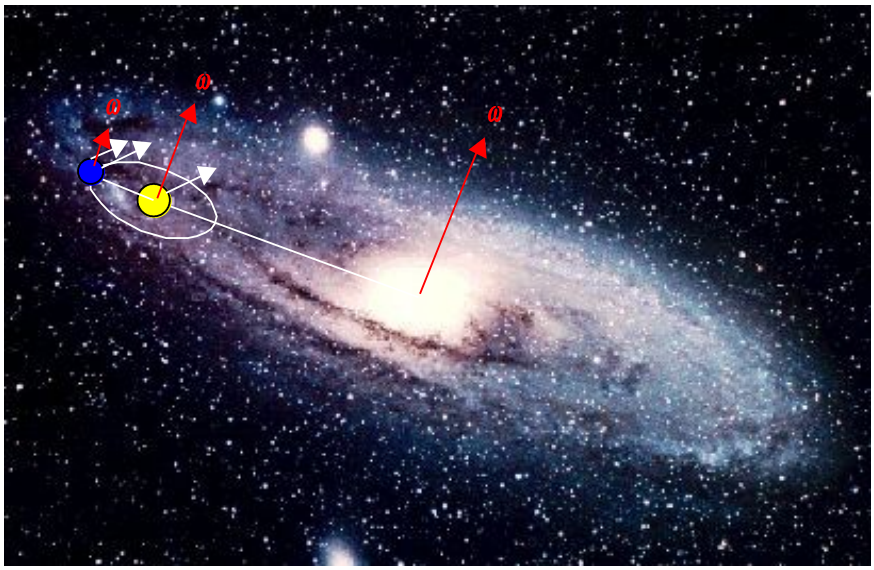
Name: _____ **Kurznamen:** _____

Rasend schnell

Obwohl wir das Gefühl haben in Ruhe zu sein, bewegen wir uns in Wahrheit mit einer rasend schnellen Geschwindigkeit durch das Weltall. Berechnen Sie die Maximalgeschwindigkeit am Äquator, die sich aus folgenden Komponenten zusammensetze:

1. Drehung der Erde um ihre Achse (1 U / Tag; Erdradius: 6370 km).
2. Rotation der Erde um die Sonne (1 U / 365.25 Tage/ Distanz Erde Sonne: 149.6 Millionen Kilometer)
3. Rotation der Sonne um das galaktische Zentrum (1 U / 225 Mio. Jahre / Distanz Sonne gal. Zentrum 25.000 Lichtjahre (1 Lichtjahr = 9.46×10^{12} km.)

Um die Geschwindigkeiten zu berechnen, verwendet man vorzugsweise Vektoren bzw. Vektorrechnung. Diese lässt sich einfach in C# einführen.



Bemerkungen: Die Distanzen sind nicht massstabsgetreu. Bei diesem Bild handelt es sich nicht um unsere eigene Galaxie, von der es noch keine Aussenaufnahmen gibt – zumindest nicht von irdischen Beobachtern.

Aufgabe 1

Entwickeln Sie ein Paket für die Vektorrechnung in 3 Dimensionen mit double Koeffizienten in C#. Es sollen folgende Operationen unterstützt werden:

- Erzeugung i.e. Konstruktor mit den drei Koeffizienten
- +, - koeffizientenweise
- * Vektorprodukt
- * Multiplikation eines Vektors mit einem Skalar und Skalar mit einem Vektor (i.e. beide Reihenfolgen der Operanden)
- Division durch einen Skalar.
- Die Koeffizienten des Vektors mit Hilfe von Indexern mit dem Bereich [0..2] sollen gelesen und gesetzt werden können.
- Konversionen: Wird der Vektor (explizit) einem Skalar zugewiesen, dann soll die euklidische Norm des Vektors berechnet werden. Wird dem Vektor (implizit) ein Skalar zugewiesen, dann soll der Vektor (x,0,0) erzeugt werden.
- Die ToString Methode soll so überschrieben werden, dass die einzelnen Komponenten des Vektors ausgegeben werden.: z.B. "[1.0,2.0,3.0]"
- ==, !=, Equals (und GetHashCode): Die Vektoren sollen auf Gleichheit überprüft werden können, wobei die Überprüfung koeffizientennweise erfolgen soll (Vergleich von einzelnen double Skalaren siehe Hinweis).

Hinweis

- Nehmen Sie das Gerüst Vector.cs
- Vektorprodukt: a x b

$$\vec{a} * \vec{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} * \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{pmatrix}$$

Test: Bsp:

```
Vector a = new Vector(1,2,3);
Vector b = new Vector(4,5,6);
Vector c = a * b;
Console.WriteLine(c);
```

sollte [-3.0 6.0 -3.0] ergeben

- Vergleich von Double Werten.

```
private static bool nearlyEqual(double a, double b)    {
  const double epsilon = 1E-10; // depends, usually parameter
  const double MinNormal = 2.2250738585072014E-308d;
  double absA    = Math.Abs(a);
  double absB    = Math.Abs(b);
  double diff    = Math.Abs(a - b);
  if (a.Equals(b)) {
    // shortcut, also handles infinities
    return true;
  }
  else if (a == 0|| b ==0 || absA +absB <  MinNormal) {
    // a or     b is zero or both are extremely  close to it
    // relative error is less meaningful here
    return diff < (epsilon * MinNormal);
  }
```

```
  else { // use relative error
    return diff / (absA + absB) < epsilon;
  }
}
```

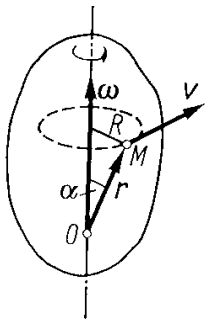
Abgabe:

Praktikum: DN 3.1

Datei: Vector.cs

Aufgabe 2

Mit Hilfe der Vektoren lässt sich die Maximalgeschwindigkeit mittels der Eulergleichung berechnen. Vereinfachend kann angenommen werden, dass die Ebene der Ekliptik, Galaxie und Erddrehung parallel verlaufen. Diese würden es zwar erlauben, rein skalar zu rechnen. Unsere Vektor Formel liefert jedoch das korrekte Resultat auch für den Fall, dass die ω und r Vektoren nicht parallel zueinander stehen. Geben Sie den Wert in km*s^{-} an.



Hinweis

- Nehmen Sie das Gerüst Space.cs
- r soll in x-Richtung und ω in y-Richtung zeigen
- Einheiten: r in km und ω in Rad pro Sec.

(Eulergleichung) $\vec{v} = \vec{\omega} \times \vec{r}$, Winkelgeschwindigkeit in Rad.

$\omega = \frac{d\varphi}{dt}$ wobei Winkelgeschwindigkeit in Rad: $2 \pi / T_{umlauf}$

Abgabe:

Praktikum: DN 3.2

Datei: Space.cs