Grafische Datenverarbeitung

Übung 1

Liebe TeilnehmerInnen der Grafische Datenverarbeitung,

in dieser Übung geht es darum, ein paar Hilfsklassen zu schreiben, die uns im weiteren Verlauf des Semesters noch von Nutzen sein werden. Für diese Übung haben Sie **7 Tage** Zeit. In Zukunft wird es auch Übungen mit 14 Tagen geben.

1. Implementieren Sie eine Klasse *Float4*, welche einen 4D Vektor repräsentiert. Dieser Vektor beinhaltet 4 *float* Werte in einem Array. Realisieren Sie die folgenden Methoden:

   - **Standardkonstruktor:** Setzt alle Werte auf 0
   - **Kopierkonstruktor:** Kopiert die Werte des übergebenen Vektors
   - **Get:** Liefert den Wert mit dem übergebenen Index zurück
   - **Add:** Addiert den aktuellen und den übergebenen Vektor und gibt das Resultat als neuen Vektor zurück
   - **AddAndSet:** Addiert den aktuellen und den übergebenen Vektor und speichert das Resultat im aktuellen Vektor
   - Sub: Siehe *Add* nur Subtraktion
   - **SubAndSet:** Siehe *AddAndSet* nur Subtraktion
   - **Mul:** Multipliziert alle Werte des aktuellen Vektors mit einem übergebenen *float* Wert und gibt das Resultat als neuen Vektor zurück
   - **MulAndSet:** Multipliziert alle Werte des aktuellen Vektors mit einem übergebenen *float* Wert und speichert das Resultat im aktuellen Vektor
   - **Dot:** Berechnet das Skalarprodukt zwischen dem aktuellen Vektor und dem übergebenen Vektor und gibt das Resultat zurück
   - **Cross:** Berechnet das Kreuzprodukt zwischen dem aktuellen Vektor und dem übergebenen Vektor und gibt das Resultat als neuen Vektor zurück
   - **CrossAndSet:** Berechnet das Kreuzprodukt zwischen dem aktuellen Vektor und dem übergebenen Vektor und speichert das Resultat im aktuellen Vektor
   - **GetLength:** Gibt die Länge des aktuellen Vektors zurück
   - **GetNormalized:** Normalisiert den aktuellen Vektor und gibt das Resultat als neuen Vektor zurück
   - **Normalize:** Normalisiert den aktuellen Vektor und speichert das Resultat im aktuellen Vektor

   Hinweise: Wer es sich zutraut, kann eine *template* Klasse *Vector4* statt *Float4* implementieren und *float* durch einen *template* Parameter ersetzen. Außerdem können Sie Operatoren verwenden.

2. Implementieren Sie eine Klasse *Matrix4x4*, welche eine 4x4 Matrix repräsentiert. Diese Matrix beinhaltet 16 *float* Werte in einem Array. Realisieren Sie die folgenden Methoden:

   - **Standardkonstruktor:** Setzt die Einheitsmatrix
   - **Kopierkonstruktor:** Kopiert die Werte der übergebenen Matrix
   - **Add:** Addiert zwei Matrizen und gibt das Resultat als neue Matrix zurück
   - **AddAndSet:** Addiert zwei Matrizen und speichert das Resultat in der aktuellen Matrix
   - **Mul:** Multipliziert zwei Matrizen und gibt das Resultat als neue Matrix zurück
   - **MulAndSet:** Multipliziert zwei Matrizen und speichert das Resultat in der aktuellen Matrix

- **Mul:** Multipliziert die aktuelle Matrix mit einem Vektor und gibt das Ergebnis als neuen Vektor zurück.

Hinweise: Wer es sich zutraut, kann auch hier mit *template* und Operatoren arbeiten.

3. Testen Sie Ihre Klassen und bauen Sie in beide Klassen Debugausgaben ein.

4. Finden Sie heraus, was die Basis eines Koordinatensystems ist.