Institut für Visualisierung
und Interaktive Systeme

Dr. Guido Reina Dr. Michael Krone

1. Übungsblatt zur Vorlesung Computergraphik im WS 2017/18

Abgabe am Donnerstag, 02.11.2017 (12:00)
Besprechung am Montag, 06.11.2016 17:30 - 19:00 Uhr

Aufgabe 1 *Einfache Funktionen und Konsoleneingabe und -ausgabe [1 Punkt]*

In dieser Aufgabe sollen Sie sich mit dem Schreiben einfacher Funktionen und der Aus- und Eingabe über die Konsole vertraut machen. Das Programm soll vom Benutzer auf der Konsole den Radius eines Kreises abfragen, den Kreisumfang berechnen und auf der Konsole ausgeben. Das gegebenen Programskelett enthält bereits den Einstiegspunkt des Programms (die `main()` Funktion) mit einer ersten Textausgabe in der Konsole.

1. Implementieren Sie eine Funktion `circumference`. Die Funktion soll den Kreisumfang für einen gegebenen Radius berechnen. (Sie können näherungsweise $\pi = 3.1415926$ verwenden).

2. Ergänzen Sie in der `main()` Funktion das Einlesen von Eingaben für den Kreisradius auf der Konsole und die Ausgabe des berechneten Wertes.

Aufgabe 2 *Verwendung der STL [3 Punkte]*

In dieser Aufgabe sollen Sie sich mit der Verwendung einiger grundlegender STL Container (`std::array`, `std::vector`, `std::list`) vertraut machen. Ergänzen Sie dazu das Struct `Point3D` und die Klasse `LineStrip3D` im gegebenen Programmskelett. Anschließend ergänzen Sie die `main()` Funktion um ein paar einfache Tests Ihrer Implementierung.

Point3D

1. Ergänzen Sie das Struct `Point3D` um ein Array, das die drei Koordinaten eines Punktes im 3D Raum enthält. Verwenden Sie den für diesen Zweck am geeignetsten STL Container und begründen Sie Ihre Wahl.

2. Implementieren Sie den Constructor und die vorgegebenen Funktionen von `Point3D`.

LineStrip3D

1. Ergänzen Sie die Klasse `LineStrip3D` um ein Array, das beliebig viele Punkte (jeweils vom Typ `Point3D`) eines Linienzuges enthält. Verwenden Sie den für diesen Zweck am geeignetsten STL Container und begründen Sie Ihre Wahl.

2. Implementieren Sie den Constructor, die Funktion `addPoint`, die Funktion `removePoint` und die Funktion `computeLength` der Klasse `LineStrip3D`.

Testprogram

1. Legen Sie in der `main()` Funktion ein Array für beliebig viele Linienzüge an. Verwenden Sie den für diesen Zweck am geeignetsten STL Container und begründen Sie Ihre Wahl. *(Hinweis: Sie können davon ausgehen, dass hinzugefügte Objekte zur Laufzeit nicht mehr entfernt werden.)*

2. Fügen Sie dem Container mehrere Linienzüge hinzu. Geben Sie anschließend die Länge der einzelnen Linienzüge auf der Kommandozeile aus.

3. Sortieren Sie den Container nach der Länge der Linienzüge und geben Sie die Länge der einzelnen Linienzüge erneut auf der Kommandozeile aus, um das Ergebnis der Sortierung zu überprüfen.

Aufgabe 3 *Pointer (Arrays) und RAII [6 Punkte]*

In dieser Aufgabe sollen Sie sich anhand einer einfachen Matrix Implementierung mit Speicherallokation und dem Umgang mit Pointern, sowie der Umsetzung des RAII Konzeptes vertraut machen. Dazu ist ein Programmskelett gegeben in dem Sie die Klasse `MatrixInt` erweitern und verwenden sollen. Das Programm lädt zwei Matrizen aus Dateien, multipliziert diese und schreibt das Ergebnis anschließend in eine dritte Datei.

1. Implementieren Sie den fehlenden Constructor und Destructor der `MatrixInt` Klasse und ergänzen Sie den Multiplikationsoperator `MatrixInt operator*(...);` so, dass eine Matrixmultiplikation stattfindet.

2. Ergänzen Sie in der Funktion `loadMatrix` das Beschreiben der Matrix mit den Werten aus der geladenen Datei. Warum wird in der Funktion `loadMatrix` bzw. `writeMatrix` der Parameter `matrix` mit `&` bzw. `const&` versehen?

3. Nennen Sie einen Vorteil des RAII Konzeptes.

4. Bonus: Welches Verhalten erwarten Sie für den Rückgabewert des Multiplikationsoperators von `MatrixInt`?