

Übungen zur Vorlesung
Simulationswerkzeuge
Dr. S. Lang, D. Popović

Besprechung am 05. Mai 2009 in der Übung

ÜBUNG 7 OPENCASCADE STAND-ALONE ANWENDUNG

Wir entwickeln nun ein Stand-alone Programm auf Basis von *OpenCascade*. Das Programm nennen wir `occ_geom`, es soll folgende Funktionalität bieten (Mini-Pflichtenheft):

- Eine CAD-Datei im *IGES*- (Endung `.igs` oder `.iges`), *BREP*- (`.brep`) oder *STEP*-Format (`.step`) kann eingelesen werden.
- Der Programmaufruf soll folgendermaßen aussehen: `occ_geom <cad-file>`, wobei `<cad-file>` eine CAD-Geometrie-Datei in einem der uns bekannten Formate ist.
- Wird das Programm mit falscher Parameteranzahl aufgerufen, soll sich das Programm nach Ausgabe einer Usage-Info beenden.
- Das einzulesende CAD-Format wird über die Endung des Dateinames `<cad_file>` gesteuert.
- Nach Einlesen der CAD-Datei sollen Informationen zur Transformation ausgegeben werden. Verwenden Sie hier wieder `Reader.PrintTransferInfo` für *IGES*-Geometrien mit den Parametern `IFSelect_xxx`. Gibt es für die anderen CAD-spezifischen Reader (*STEP*, *BREP*) eine ähnliche Methode?
- Die Applikation soll in drei Dateien gegliedert werden. Eine Header-Datei `occ_geom.hh` definiert eine Klasse `OCC_GEOM`, die die Benutzerschnittstelle definiert. Die Datei `occ_geom.cxx` implementiert die Methoden. Eine Datei `main.cxx` steuert die Parameterauswertung und verwendet die Klasse `OCC_GEOM` um die Geometrie einzulesen.
- Eine Methode soll `IndexedMaps` aufbauen können. Die `IndexedMaps` speichern eindeutige Referenzen auf Objekte vom Typ `TopAbs_FACE`, `TopAbs_EDGE`, `TopAbs_VERTEX`. Definieren sie sich geeignete Iteratoren zum Iterieren über alle vorhandenen Objekte, Typ `TopAbs_{SOLID,SHELL,FACE,EDGE,VERTEX}` und sorgen Sie in den Maps `TopTools_IndexedMapOfShape` `faces`, `edges`, `vertices` für Eindeutigkeit. Lesen Sie dazu im *Modeling Data Users Guide* (`Modat.pdf`) das Kapitel *Topology* auf S.35ff. Besonders die Beispiele ab S.44ff sollten sich ansehen, bevor Sie diese Methode realisieren.
- Eine Methode, welche die Anzahl der Shapes in den Maps ausgibt.

Orientieren Sie sich an den Implementierungen von `Translate.cxx` (*IESample*) und `MakeBottle.cxx` (Tutorial). Im Projekt-Verzeichnis finden Sie zwei Verzeichnisse:

`occ_geom_skel` und `occ_geom_skel_light`. Beide sind Projekt-Verzeichnisse mit einer Klassendefinition `occ_geom.hh`, einem Teil der Implementierung in `occ_geom.cxx`, einer Haupt-Applikation `main.cxx` und einem Makefile. Das Programmgrüst im ersten Verzeichnis ist etwas schmaler und für erfahreneren C++-Programmierer gedacht, das zweite soll von den weniger Geübten verwendet werden. Das Programm können Sie jeweils mit `make` erzeugen.

Tipps:

- Laden Sie vor Programmstart die Konfigurations-Datei: `source occ_debian.sh`.
- Alle Stellen in den Quelldateien, die Sie bearbeiten und an denen Sie Code einfügen müssen, sind markiert:

```
//----- ...
```

- Die Argumente, die einem Program mitgegeben werden, stehen im Parameter `argv` der Funktion `main`. Der Programmname steht im ersten Eintrag, d.h. `std::cout << argv[0]`; gibt den Programmnamen aus. Die Anzahl der Argumente steht in `argc`.
- Mit der C++-String-Methode `compare(<string>)` können Sie C++-Strings auf Gleichheit überprüfen, mit den String-Methoden `find` und `rfind` einen Substring in einem String finden.
- Unbedingt die *OpenCascade*-Doku lesen, da wird vieles erklärt!