

Objektorientierte Programmierung mit C++ (WS 2010)

Dr. Andreas F. Borchert, Tobias Brosch
Institut für Angewandte Informationsverarbeitung
Universität Ulm
Blatt 2: Abgabetermin 03. November 2010

1 A UML drawer

In diesem Übungsblatt sollen einige UML-Diagramme erstellt werden. Eine Möglichkeit, dies am Rechner zu erstellen, ist MetaUML. Manual: http://sourceforge.net/projects/metauml/files/metauml_manual/0.2.5/metauml_manual_0.2.5.pdf/download. Zusätzlich gibt es einige einfache Beispiele auf der Vorlesungshomepage. Mit im Paket ist ein `Makefile`, mit welchem Ihr aus der `.mp`-Datei eine `.pdf`-Datei erzeugen könnt. Hierzu einfach das Programm `make` aufrufen. Die pdf-Datei könnt Ihr an Eurem lokalen Rechner in unserem Pool

- mit `xpdf` betrachten. Mit `xpdf name.pdf &` startet Ihr `xpdf` im Hintergrund als Job. Die aktiven Jobs könnt Ihr mit dem Programm `jobs` anzeigen lassen. Mit `kill %2` könnt Ihr den Hintergrundjob mit der Nummer 2 beenden, mit `fg %1` könnt Ihr den Job mit der Nummer 1 wieder in den Vordergrund holen.
- wenn Ihr auf `theseus` arbeitet, auf Eurem lokalen Rechner kopieren, indem Ihr eine Shell auf Eurem Rechner auf macht. In diese lokale Shell (ohne `ssh`-Verbindung!) gebt Ihr

```
scp login@theseus.mathematik.uni-ulm.de:tmp/uml/test.pdf ./
```

ein, um die Datei `test.pdf` im Verzeichnis `tmp/uml` in Eurem Heimatverzeichnis auf `theseus` in euer derzeitiges, lokales Verzeichnis zu kopieren.

Ihr könnt aber natürlich auch gerne jedes andere euch vertraute Programm verwenden, welches nach PDF exportieren kann (auf unseren Rechnern z.B. `Xfig`).

2 Ein einfacher Blog

Wir betrachten einen einfachen Web-Blog, der einen Namen hat. Dieser hat beliebig viele Artikel, die von einem oder mehreren Autoren verfasst werden können. Jedem Artikel können beliebig viele Tags zugewiesen werden, die eine Kategorie angeben. Zusätzlich kann jeder Artikel beliebig oft kommentiert werden, wobei sich jeder Kommentar auf genau einen Artikel beziehen soll.

Was soll unser Blog sonst können:

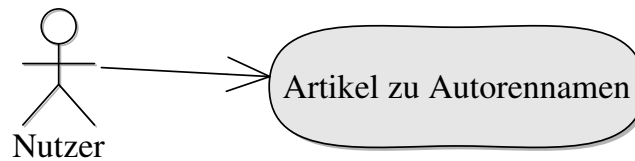
- Die neuesten Artikel anzeigen.
- Uns zu einem Autorennamen alle Artikel liefern.
- Zu einem Tag alle Artikel liefern.
- Uns zu einem Autor dessen Blog-Id ausgeben können.
- Ein Autor mit Koautoren möchte gerne einen Artikel schreiben können.
- Wir möchten gerne wissen, wann ein Artikel geschrieben wurde, was sein Text ist, wer seine Autoren sind, welche Tags er hat und welche Kommentare zu ihm geschrieben wurden.

2.1 Klassendiagramm

Überlegt euch, welche Daten vorgehalten werden müssen, wie diese geschickt durch Klassen repräsentiert werden können und welche Methoden die einzelnen Klassen haben müssen, um die gewünschten Aufgaben ausführen zu können. Zeichnet ein UML-Klassendiagramm, welches die Klassen, die Navigationsmöglichkeiten und die Multiplizitäten aufführt.

2.2 Use-Case Diagramm

Ergänzt folgendes Diagramm um einen Actor Autor und mindestens einen weiteren Use-Case.



2.3 Sequenz-Diagramm

Erstellt mit einem Programm Eurer Wahl (auf unseren Rechnern gibt es z.B. Xfig) ein Sequenz-Diagramm des Beispiel Use-Cases aus Aufgabe 2.2.

Submitted die UML-Diagramme im pdf-Format wie am Ende des Blattes angegeben.

3 Klassen

Auf der Vorlesungshomepage findet Ihr ein Paket mit einer Klasse `Blog`. Kopiert euch diese in einen neuen Ordner (z.B. mit `wget`). Entpackt, kompiliert und führt die Tests dieser Klasse anschließend aus:

```
theseus$ls
Blog.cpp      Blog.h        main.cpp
```

```

theseus$g++ -c Blog.cpp
theseus$g++ -c main.cpp
theseus$g++ Blog.o main.o -o main
theseus$./main
Blog number 0 with name 'Mein□erster□Blog' constructed
Blog number 1 with name 'Mein□zweiter□Blog' constructed
Id 1. Blog: 0
Name 2. Blog: 'Mein□zweiter□Blog'
Blog destructed
Blog destructed
theseus$

```

Aufgabe: Lest die Kommentare in den Dateien durch, versucht euch die Bedeutung einigermaßen klar zu machen und spielt ein wenig mit dem Source-Code. Versucht anschließend (ohne copy&paste) eine zweite, einfache Klasse, zu schreiben. Keine Sorge, wenn euch hier und da noch etwas unbekannt vorkommt. Diese Aufgabe soll euch lediglich eine erste Chance bieten, ein wenig mit Objekten zu spielen, und die Ähnlichkeit zu Java zu erkennen.

Submitted die sechs Dateien `blog.pdf`, `usecase.pdf`, `seq.pdf`, `main.cpp`, `MyClass.h` und `MyClass.cpp`, welche auf theseus kompiliert und ausgeführt werden können bzw. mit einem PDF-Reader betrachtet werden können.

```
submit cpp 2 blog.pdf usecase.pdf seq.pdf main.cpp MyClass.h MyClass.cpp
```

Falls Ihr nicht alle Dateien submitten wollt, könnt Ihr auch nur einen Teil der Dateien angeben. Achtet jedoch bitte darauf, bei Eurer letzten Submission alle Dateien die Ihr submitten möchtet mit anzugeben!

4 FAQ

- Geht es euch in den Übungen zu schnell oder zu langsam? Würdet Ihr gerne noch etwas zu einem bestimmten Thema hören?
- → Schreibt Eurem Übungsleiter eine kurze und freundliche Mail! Er wird versuchen, darauf in den nächsten Übungen einzugehen.
- Nächste Woche am 03.11.2010 ist Vorlesung. Daher wird die Übung lediglich in den letzten 15 Minuten stattfinden.

Viel Spaß!