Dr. Matthias Grabert — Abteilung Angewandte Informationsverarbeitung — 14. Januar 2003
Johannes Mayer — Blatt 11

Objektorientierte Softwareentwicklung mit C++ (WS 2002/2003)

Projekt (75 Punkte)

Wie besprochen werden wir die restlichen Übungsblätter zu einem einzigen zusammenfassen, das dann ein kleines Projekt bildet. Dabei dürfen Sie nun in einem (vielleicht etwas größeren) Team ein etwas umfangreicheres Programm entwickeln. Dabei lernen Sie dann auch die Softwareentwicklung in einem etwas größeren Umfang kennen und können „Teamarbeitsluft schnuppern“.

Finden Sie sich für das Projekt zunächst bis **Donnerstag, 16.1.03,** zu Teams von 5 bis 10 Personen. Dabei dürfen Sie – sofern das mit den Tutoren abgesprochen ist und funktioniert – die Gruppen neu bilden bzw. zwei Tutorien zu einer Gruppe zusammenfassen. Es ist erwünscht, dass die Gruppen etwas größer sind (8 bis 10 Personen). Dies bringt auch den Vorteil mit sich, dass die Aufgaben auf mehr Leute verteilt werden können und somit jeder weniger zu tun hat. ;-)

Suchen Sie sich außerdem bis **Donnerstag, 16.1.03,** ein Projekt heraus und werden Sie sich klar darüber, was das Ergebnis Ihres Projektes sein soll. Dabei sollten Sie ein paar Zeilen zur Dokumentation Ihres Projektes aufschreiben. ;-)

Ein mögliches Projekt muss die folgenden Anforderungen erfüllen:

- Die Konzeption Ihres Programmes muss **objektorientiert** erfolgen, wobei Polymorphie auch ins Spiel kommen sollte. Außerdem soll es ein paar nicht-triviale (!) Klassen geben.
- Das entwickelte Programm muss eine mit **Qt** entwickelte **GUI** besitzen.
- Außerdem soll in Ihrem Programm auch **Persistenz** vorkommen, d. h. die Objekte von mindestens einer Klasse sollen beim Beenden des Programms gesichert und mein Neustart wieder geladen werden. Diese Sicherung kann beispielsweise in eine kleine UNIX-Datenbank erfolgen (jede Zeile, deren Felder durch z. B. Doppelpunkt getrennt sind, enthält einen Datensatz, also ein Objekt).
- Natürliche soll die Programmierung in **C++** erfolgen. ;-) Dabei können Sie die Plattform (also UNIX, Linux, Windows, ...) selbst auswählen. Falls Sie nicht unter UNIX entwickeln, müssen Sie eben sicher stellen, dass Ihre Präsentation klappt. ;-)

Als Projekt können Sie sich entweder ein eigenes wählen, das obigen Anforderungen genügt, oder eines der folgenden Vorschläge:

1. **Grafisches Sokoban** mit Leveleditor, Highscoreliste und Undo-Möglichkeit (beliebig viele Züge rückgängig machen)
2. **Netzwerkspiel** für „Vier gewinnt“ oder Schach: Zwei Spieler können gegeneinander spielen (auch verteilt über’s Netz) – keine Computerlogik erforderlich (!!) – mit Undo-Möglichkeit von beliebig vielen Zügen und Speicherung der Zugfolge, sowie Highscores; evtl. auch mit Spielstand-Editor
3. **Grafische Verkehrssimulation** mit beliebigen rechtwinkligen Straßensystemen, einem Editor für die Straßen und Animation der Simulation

Was muss man also machen, um den **Schein** zu bekommen? Notwendig sind dazu 50 % der gesamten Übungspunkte (inklusive der Punkte für das Projekt). Außerdem muss eine Präsentation und Abnahme des Projektes in der letzten Semesterwoche erfolgreich absolviert werden. Last but not least müssen 50 % der Klausurpunkte erreicht werden. Dafür gibt’s dann den Schein.

Damit Ihnen die Organisation Ihres Projektes leichter fällt bekommen Sie folgenden groben Plan, den Sie **einhalten müssen**. Sie können jedoch auch schneller sein. ;-) Dann bleibt Ihnen mehr Zeit, um sich auf die Klausur vorzubereiten. Diesen Plan sollen Sie selbst verfeinern, wobei die Grobstruktur erhalten bleiben soll. Besprechen Sie Ihren Plan mit Ihrem Tutor.

Teambildung und Projektfestlegung (bis 16.1.2003)

Finden Sie sich zu einem Team von 5 bis 10 Personen zusammen und arrangieren Sie sich mit einem Tutor. Hierzu kann es ganz praktisch sein, zwei (oder mehr) bestehende Tutorien zu kombinieren. Legen Sie sich gemeinsam auf ein Projekt fest und beschreiben Sie dieses mit ein paar Zeilen Text.

Pflichtenheft und Aufgabenverteilung (bis 23.1.2003)

Erstellen Sie ein kleines Pflichtenheft, das eine detaillierte, klare Beschreibung Ihres Projektes (also Problem, Ziel und Funktionalität Ihres Programms) enthält. Modellieren Sie die Lösung des Problems mit Hilfe von Klassen in UML (Klassendiagramme). Spezifizieren Sie die Schnittstellen (d. h. public und protected Methoden und Variablen) Ihrer Klassen. Verteilen Sie die Aufgaben auf die einzelnen Teammitglieder (jeder muss für einen eigenen Teil verantwortlich sein). Entwerfen Sie auch einen eigenen detaillierten Projektplan, der genau festlegt, wer wann was fertig stellen wird. Besprechen Sie Pflichtenheft, Modellierung, Aufgabenverteilung und Plan mit Ihrem Tutor.

Prototyp (bis 30.1.2003)

Schreiben Sie Prototypen Ihrer Klassen/Module. Jedes Gruppenmitglied sollte seine Teile unter Verwendung der Module der Anderen testen können. Diese müssen noch nicht die volle Funktionalität besitzen, aber die Aufrufe müssen bereits möglich sein.

Implementierung, Integration und Test (bis 6.2.2003)

Vervollständigen Sie die Implementierung des Prototypen. Kombinieren Sie die einzelnen Module/Klassen und beseitigen Sie evtl. noch vorhandene Probleme. Testen Sie das komplette Programm und überprüfen Sie die Erfüllung aller Anforderungen.

Präsentation der Projekte und Abnahme (bis 13.2.2003)

Am 11. und 13.2.2003 findet die Präsentation aller Projekte in Vorlesung bzw. Übungen statt. Jede Gruppe hat ca. 10 Minuten Zeit, Ihr Projekt vorzustellen. Außerdem findet noch eine Abnahme durch den Tutor statt, der überprüft, ob das Programm funktioniert und den Anforderungen entspricht. Des Weiteren überprüft der Tutor auch, ob jedes Teammitglied auch sich zumindest mit seinem Teil auskennt. ;-)

P.S.: Hier ist Teamarbeit gefragt ... nicht Einzelkämpfertum! (Kleinere Teams als mit 5 Personen gibt es nicht.)
P.P.S.: Die Punktevergabe erfolgt für das gesamte Team und nicht für jedes Teammitglied einzeln!!!