

# Informatik: Semester-Gruppenarbeit „MultiChannel“ Daten übermitteln an Einzelne und Gruppen zu bestimmten Zeit und Kanälen und Ausgabe auf UI.

Studiengang: Informatik

Autor: Stefan Berger

Datum: 23.02.2011

## Inhaltsverzeichnis

<u>Inhaltsverzeichnis.....</u>	<u>2</u>
<u>Inhaltsverzeichnis.....</u>	<u>2</u>
<u>Inhaltsverzeichnis.....</u>	<u>2</u>
<u>Inhaltsverzeichnis.....</u>	<u>2</u>
<u>1 Einleitung.....</u>	<u>3</u>
<u>2 Ziel.....</u>	<u>4</u>
<u>3 Auftrag.....</u>	<u>5</u>
<u>4 Bewertungskriterien und Bewertungsschema.....</u>	<u>6</u>

# 1 Einleitung

Dieser Lernauftrag besteht aus mehreren Teilen. Es werden verschiedene API angewendet, diese müssen in einer Einheit eines Produktes genutzt werden.

Der Auftrag fördert Ihr gesamtheitliches Denken. Aus verschiedenen API müssen sie ein komplexes Ganzes herstellen. Diese Projektarbeit wird in 2-er Gruppen gelöst. 3 Gruppen präsentieren nach der Abgabe ihre Lösung im Plenum (wird am Abgabeabend ausgelost).

## 2 Ziel

Festigung der Grundlagen und objektorientierten Techniken im Umgang mit Design (UML) und Entwicklung. Aufbau und Vertiefung von einzelnen API, damit Sie diese in der Praxis zweckmässig und effizient einsetzen können.

Sie entdecken die Möglichkeiten der OO - Techniken und deuten deren Potential in der Aufgabenstellung an.

### 3 Auftrag

Konzipieren Sie eine Software die Nachrichten auf unterschiedlichen Art versenden / transportieren kann.

Eine Nachricht kann als 'SMS', 'MMS', 'Email', 'Print' versendet werden.

Jede Nachricht kann an einzelne Personen aber auch Gruppen gesendet werden. Jede Nachricht kann zu einem bestimmten Zeitpunkt versendet werden oder auch sofort.

Jede SMS, MMS und Email kann zu einem bestimmten Termin eine Nachricht vorab senden. (Reminder) (Bzw. Termin ist um 18:00 Uhr – Erinnerung ist um 17:00)

Bedenken Sie, welche Faktoren einen Einfluss bei Nachricht haben. Berücksichtigen Sie nur logische Gründe nicht physische Ursachen/Faktoren.

Bei allen 4 Kanälen müssen Validierungen durchgeführt werden, damit der mögliche Transport gewährleistet werden kann.

Überlegen Sie sich die Use Cases zeichnen und beschreiben Sie diese.

Bilden Sie ein Klassendiagramm, welches die Beziehungen und Interaktionen aufzeigt. Erklären Sie, welche Voraussetzungen Sie warum angenommen haben.

Entwickeln Sie ein Test Programm, dass entweder auf User – Eingaben reagiert oder für alle 4 Kanäle mit simulierten Testwerten durchläuft.

Die Eingaben bzw. Defaultwerte und die Resultate der einzelnen Kanäle muss auf der Konsole einsehbar sein.

Das Programm muss erklärt und funktionstüchtig sein.

***(Achtung: das Versenden muss nur als Konsolenmeldung simuliert werden. Für funktionierende SMS, MMS, Emails oder Printaufträge werden keine Punkte gegeben!!)***

Wichtig: Die oben gemachten Angaben lassen bewusst Spielraum für eigene Ideen.

Diese Aufgabe kann smart und effizient mit möglichem Zukunftspotential gelöst werden, das Gegenteil wäre eine einfache, minimalistische ohne jede Wiederverwendbarkeit oder Ausbaufähigkeit im Sinne der Objektorientiertheit.

Es geht ums OO – Denken und Entwickeln, nicht um ein Programmchen das einfach läuft.

Bauen Sie die Software so, dass Sie den Kern der Aufgabe erfasst und im Sinne von OO ausgebaut werden kann.

Die Bewertung der Aufgabe zu 2/3 erfolgt im Stil einer Experteneinschätzung, 1/3 erfolgt im Stile Korrektur von harten Kriterien (Konventionen, UML formale Fehler)

In der Praxis gilt dasselbe: Auch korrekte UML / Klassendiagramme, erfüllte Konventionen können auch in einem völlig unbrauchbaren Programm vorhanden sein, dennoch bleibt das Programm unbrauchbar.

**Abgabe:**

**Nur Source - Code, HTML; PDF (ZIP Datei)**

**Alle anderen Formate werden nicht akzeptiert und entsprechend nicht gewertet:**

### 3.1 Besonderes

Die oben gemachten Angaben lassen bewusst Spielraum für eigene Ideen.

Diese Aufgabe kann smart und effizient mit möglichem Zukunftspotential gelöst werden, das Gegenteil wäre eine einfache und reduzierte Lösung ohne jede Wiederverwendbarkeit oder Ausbaufähigkeit im Sinne der Objektorientierung.

Es geht ums OO – Denken und Entwickeln, nicht um ein Programm das einfach läuft.

Bauen Sie die Software so, dass Sie den Kern der Aufgabe erfasst und im Sinne von OO ausgebaut werden kann.

Codex: Jede Nutzung von fremden Code ist unter Angabe der Quelle zu benennen. Jedes Dokument (Code, UML, Text) muss einen Autor haben.

### 3.2 Gruppenbildung, Gruppendynamik

Gruppenbildung: Wählen Sie Ihre Gruppenmitglieder gut aus. Bedenken Sie das zwischenmenschliche Probleme öfters ein Grund sind, wenn ein Projekt scheitert oder nur zu Teilen fertiggestellt wird.

Bitte planen Sie folgende Möglichkeiten ein und beugen diesen entsprechend vor:

- a) StudentIn steigt aus dem Studium aus (Datensicherstellung – Aufteilung der weiteren Arbeit)
- b) StudentIn ist länger krank → wie wird weitergearbeitet
- c) Es kommt zum Streitfall in der Gruppe, Disharmonie → wie wird weitergearbeitet
- d) Ein Gruppenmitglied macht nichts mehr, aber die ganze Gruppe wird gleich bewertet. Wie wird die Aufgabenverteilung organisiert ?

### 3.3 Tutoriat

Was ist ein Tutoriat ?

- a) keine Pflichtstunden
- b) kein Präsenzunterricht
- c) keine explizite Nachschulung

Sondern:

- a) Gezielte Fragestellung an den Dozenten, die möglicherweise alle Gruppen betreffen könnten
- b) Meinungsaustausch mit Dozenten, Lösungsszenarios besprechen
- c) Vorschläge, Alternativen in Gruppen besprechen.

## 4 Bewertungskriterien und Bewertungsschema

Die Bewertung erfolgt nach verschiedenen Kriterien, die auf die Themen des Kurses:

Kriterium	Punkte
Einhalten der Code Konventionen	6
Übersichtliche Code Gestaltung	6
Verwendete Code Strukturen	6
Allgemeine Funktionalität und Robustheit, Fehlertoleranz	6
Erfüllung der Testfälle	6
Korrekte Verwendung der UML Notation	6
Qualität des Use Cases und Nachvollziehbarkeit in Code	6
Qualität des Klassendiagramm und Nachvollziehbarkeit in Code	6
Objektorientierter Ansatz (Weg der Analyse, Klassendesign, Abstraktion, Anwendung des OO - Paradigmas)	18
Total	64

60+ Punkte	Note 6
50 Punkte	Note 5
40 Punkte	Note 4
30 Punkte	Note 3
20 Punkte	Note 2
0 Punkte	Note 1