

Planungsentwurf zur Unterrichtsvorbereitung zur Java Programmierung eines E-Mail-Clients

Angaben:

Referent:	Johannes Klebert
Universitätsbetreuer:	Prof. Dr. S. Schubert, Dipl.-Ing. St. Freischlad
Mentor:	N.N.
Ort:	N.N.
Datum:	02.05.2006
Zeit:	9:35 – 11:10 Uhr (3. und 4. Unterrichtsstunde)
Curriculare Einbindung:	Java Programmierung
Stundenthema:	Implementierung eines E -Mail-Clients zum Versenden von E-Mails.

1 Getroffene Entscheidungen

1.1 Thematischer Zusammenhang

Art der Stunde: Praxis Stunde im Computer-Raum, Java Programmierung

1.2 Ziele der Doppel-Stunde

a.) Stunden-Lernziel

Die Schülerinnen und Schüler können einen Java basierten SMTP-Client planen und in seinen Grundzügen implementieren.

b.) Teil-Lernziele

Die Schülerinnen und Schüler

L1. kennen die Unterschiede und Gemeinsamkeiten von SMTP und POP3.

L2. können den Ablauf eines Zustandsdiagramms in Java-Quelltext übertragen.

L3. können ein UML-Klassendiagramm zu einem bekannten Problem richtig ergänzen

1.3 Hausaufgaben

Die Schüler sollen als Hausaufgabe einen Quelltextausschnitt aus der Klasse POP3Client bearbeiten, wie Sie es bereits vorher mit einem Quelltextausschnitt aus der Klasse SMTPClient getan haben.

1.4 geplanter Unterrichtsverlauf

Siehe Unterrichtsskizze.

2 Begründung zentraler didaktischer Entscheidungen

2.1 Lerngruppe

Bei der Lerngruppe handelt es sich um einen Informatikkurs der Jahrgangsstufe 11 mit insgesamt 15 Schülern (3 Mädchen, 12 Jungen). Vor den Ferien wurde mit dem Aufbau von Rechnernetzen ein neues Thema begonnen, welches in der vorherigen Stunde vertieft wurde. Die Schüler kennen nun das Prinzip der Kommunikation von Rechnern und Rechnernetzen. Sie haben den korrekten Ablauf des SMTP Protokolls mit Hilfe eines Zustandsdiagramms modelliert und damit, via Telnet, E-Mails verschickt. Als Hausaufgabe hatten die Schüler auf, zu schreiben, welchen Weg eine E-Mail von einer Person zu einer anderen Person nimmt, und welche Informationen dabei weitergegeben werden.

2.2 Darstellung der Entscheidungen

In der Doppelstunde sollen die Schüler an das Programm „Postamt“, mit dem noch in den folgenden Stunden gearbeitet werden soll, herangeführt werden. Der Einstieg in die Stunde erfolgt über die Abfrage der Hausaufgaben der vorherigen Stunde. Hierbei soll noch einmal klar Bezug zum Zustandsdiagramm, ebenfalls aus der vorherigen Stunde hergestellt werden. Dies bereitet gleichzeitig die spätere Bearbeitung des Arbeitsblattes „Quelltext“ vor.

Doch zunächst sollen die Schüler das Arbeitsblatt „Beziehungs-Diagramm“ gelöst werden. Dies dient dazu, dass die Schüler die Anforderungen an ein E-Mail-Programm spezifizieren können. Außerdem wird auf dem Arbeitsblatt noch einmal der Unterschied zwischen SMTP und POP3 betont. Die Lösung der Schüler wird via Folie und OHP gesichert. Dies könnte auch in der Form geschehen, dass die Schüler selbst einzelne Lösungsteile eintragen. Abschließend noch ein Tafelanschrieb mit einer Definition von SMTP und POP3.

Nachdem nun die Schüler bereits eine grobe Vorstellung der Anforderungen an ein E-Mail-Programm haben, soll nun der Einstieg in die Software „Postamt“ gefunden werden. Dazu sollen die Schüler das Arbeitsblatt „Quelltext“ bearbeiten. Dies beinhaltet einen Ausschnitt aus der Klasse SMTPClient. Der vorgelegte Quelltext ist in kleinere Blöcke gegliedert, welche in einer falschen Reihenfolge angeordnet sind. Mit Hilfe des Zustandsdiagramms, welche zu Beginn der Stunde wiederholt wurde, können die Schüler nun den Quelltext in die richtige Reihenfolge bringen. Für die Lösung des Arbeitsblattes haben die Schüler bis zum Ende der Stunde Zeit.

Die erste Stunde ist nun vorbei und die Schüler wechseln in den PC-Raum. Zu Beginn wird nun die Software „Postamt“ via USB-Stick an die Schüler verteilt und die Schüler kopieren diese auf ihre Rechner und importieren sie in ihre Java-Entwicklungs-Umgebung. Nach etwa 5 Minuten sollten alle Schüler PCs mit der Software ausgestattet sein und das Arbeitsblatt „Postamt“ wird ausgeteilt. Bei der Bearbeitung werden die Schüler recht schnell feststellen, dass die Software noch nicht einwandfrei funktioniert. Um die volle Funktionalität des Programms wiederherzustellen, müssen

die Schüler, wie bereits auf dem Arbeitsblatt „Quelltext“ geübt, den Dienst „ausführen()“ sortieren.

Sobald die Schüler ihre Lösung vom Arbeitsblatt in den Quelltext übertragen haben, können sie mit der Bearbeitung vom Arbeitsblatt „Postamt“ fortfahren. Da dieses Arbeitsblatt nur eine stützende Funktion bei der explorativen Erforschung der Software haben soll, ist eine weitere Sicherung der Ergebnisse nicht von Nöten.

3 Lernmaterial

Tafelbilder
Beamer
Computer
Arbeitsblatt

4 Literatur

- [Schwill und Claus, 2001], Schwill, A; Claus, V.; Duden der Informatik. Ein Fachlexikon für Studium und Praxis. Dudenverlag. Mannheim, 2003

- [Schubert und Schwill, 2004] Schubert, S.; Schwill, A.; Didaktik der Informatik. Spektrum Akademischer Verlag. München, 2004.

- www.wikipedia.de

- [Gudjons, 2003] Gudjons, Herbert. Pädagogisches Grundwissen. Überblick - Kompendium - Studienbuch. Verlag Julius Klinkhardt. Regensburg, 2003.

- [Blankertz, 2000] Blankertz, Herwig. Theorien und Modelle der Didaktik. Juventa Verlag. München, 2000.

Unterrichtsskizze

Java Programmierung eines Email-Clients zum Versenden von Nachrichten

<i>Zeit/M in</i>	<i>Phase</i>	<i>Unterrichtsschritt inhaltlich</i>	<i>Methode</i>	<i>Medien</i>
09:35	Begrüßung	Begrüßung, Klärung des weiteren Unterrichtsverlauf	LV	Tafel
09:40	Wiederholung	Wiederholung des Zustandsdiagramms SMTP von C. Dittich vom 25.04 Besprechung der Hausaufgabe. Klärung des Unterschieds SMTP und POP3.	LSG	Tafel
09:50	Erarbeitung 1	Bearbeitung des Arbeitsblattes „UML-Puzzle“	EA/GA(2)	AB
10:00	Sicherung 1	Die von den Schülern gefundene Lösung wird auf einer Folie eingetragen. Tafelanschrieb: SMTP und POP3	LSG	Tafel
10:10	Erarbeitung 2	Bearbeitung des Arbeitsblattes „Quelltext“	EA/GA(2)	AB
10:20	Pause	5min Pause, Schüler wechseln den Raum		
10:25	Installation	Die Schüler installieren die Software „Postamt“ auf ihren PCs. Datentransfer via. USB Stick.	EA/GA(2)	USB/CO
10:30	Motivation	Die Schüler erforschen das Programm „Postamt“ explorativ.	EA/GA(2)	CO
10:35	Erarbeitung 3a	Austeilung & Bearbeitung des Arbeitsblattes „Postamt“	EA/GA(2)	CO
10:40	Motivation	Die Schülern stellen fest, dass das Programm noch nicht vollständig ist.	LSG	Beamer
10:45	Sicherung 2	Die Schülern müssen, um den korrekten Ablauf des Programms zu ermöglichen, den Dienst „ausführen()“ sortieren. Dies haben Sie bereits auf dem AB „Quelltext“ geübt.	EA/GA(2)	CO
10:55	Erarbeitung 3b	Die Schüler setzen die Bearbeitung des Arbeitsblattes „Postamt“ fort.	EA/GA(2)	CO

<i>Zeit/M in</i>	<i>Phase</i>	<i>Unterrichtsschritt inhaltlich</i>	<i>Methode</i>	<i>Medien</i>
11:05		Austeilen & Erklären der Hausaufgabe		
11:10		Ende		

AB: Arbeitsblatt

BE: Beamer

CO: Computer

EA: Einzelarbeit

GA(2): Zweiergruppen

LSG: Lehrer-Schüler-Gespräch

TA: Tafel

GA: Gruppenarbeit

Tafelanschrieb:

SMTP ist ein Protokoll, das zum Austausch von E-Mails in Computer-Netzwerken dient. Es wird dabei zum versenden von E-Mails verwendet. SMTP Server nehmen traditionell Verbindungen auf Port 25 entgegen.

POP3 ist ein Protokoll, über welches ein Benutzer E-Mails von einem E-Mail-Server abholen kann. POP3-Server nehmen Verbindungen für Port 110 entgegen.

Arbeitsblatt „Quelltext“

Der folgende Quelltext ist fehlerhaft. Die SMTP-Befehle werden in einer falschen Reihenfolge ausgeführt. Bitte sortieren Sie die Befehle so, dass eine E-Mail korrekt versendet werden kann. Benutzen Sie das Zustandsdiagramm als Hilfe. (Der fett gedruckte Bereich ist korrekt).

```
public boolean ausfuehren() {
    boolean fehler = false;
// _____

    // Zustand a
    if (!fehler) {
        if (!sendeData()) fehler = true;
    }
// korrekter Zustand: _____

    // Zustand b
    if (!fehler) {
        if (!sendeEmpfaenger()) fehler = true;
    }
// korrekter Zustand: _____

    // Zustand c
    if (!verbinden()) fehler = true;
// korrekter Zustand: _____

    // Zustand d
    if (!fehler) {
        if (!sendeAbsender()) fehler = true;
    }
// korrekter Zustand: _____

    // Zustand e
    if (!fehler) {
        if (!beenden()) fehler = true;
    }
// korrekter Zustand: _____

    // Zustand f
    if (!fehler) {
        while (nachrichtenCode(antwort) != 354) {
            antwort = verbindung.empfangen();
        }

        if (!sendeNachricht()) fehler = true;
    }
// korrekter Zustand: _____

// _____
    return fehler;
}
```

Die korrekte Abfolge lautet also: _____

Arbeitsblatt „Quelltext“ Lösung

Der folgende Quelltext ist fehlerhaft. Die SMTP-Befehle werden in einer falschen Reihenfolge ausgeführt. Bitte sortieren Sie die Befehle so, dass eine E-Mail korrekt versendet werden kann. Benutzen Sie das Zustandsdiagramm als Hilfe. (Der fett gedruckte Bereich ist korrekt).

```

public boolean ausfuehren() {
    boolean fehler = false;
// _____

    // Zustand a
    if (!fehler) {
        if (!sendeData()) fehler = true;
    }
// korrekter Zustand: _____ 5

    // Zustand b
    if (!fehler) {
        if (!sendeEmpfaenger()) fehler = true;
    }
// korrekter Zustand: _____ 4

    // Zustand c
    if (!verbinden()) fehler = true;
// korrekter Zustand: _____ 1

    // Zustand d
    if (!fehler) {
        if (!sendeAbsender()) fehler = true;
    }
// korrekter Zustand: _____ 3

    // Zustand e
    if (!fehler) {
        if (!beenden()) fehler = true;
    }
// korrekter Zustand: _____ 8

    // Zustand f
    if (!fehler) {
        while (nachrichtenCode(antwort) != 354) {
            antwort = verbindung.empfangen();
        }

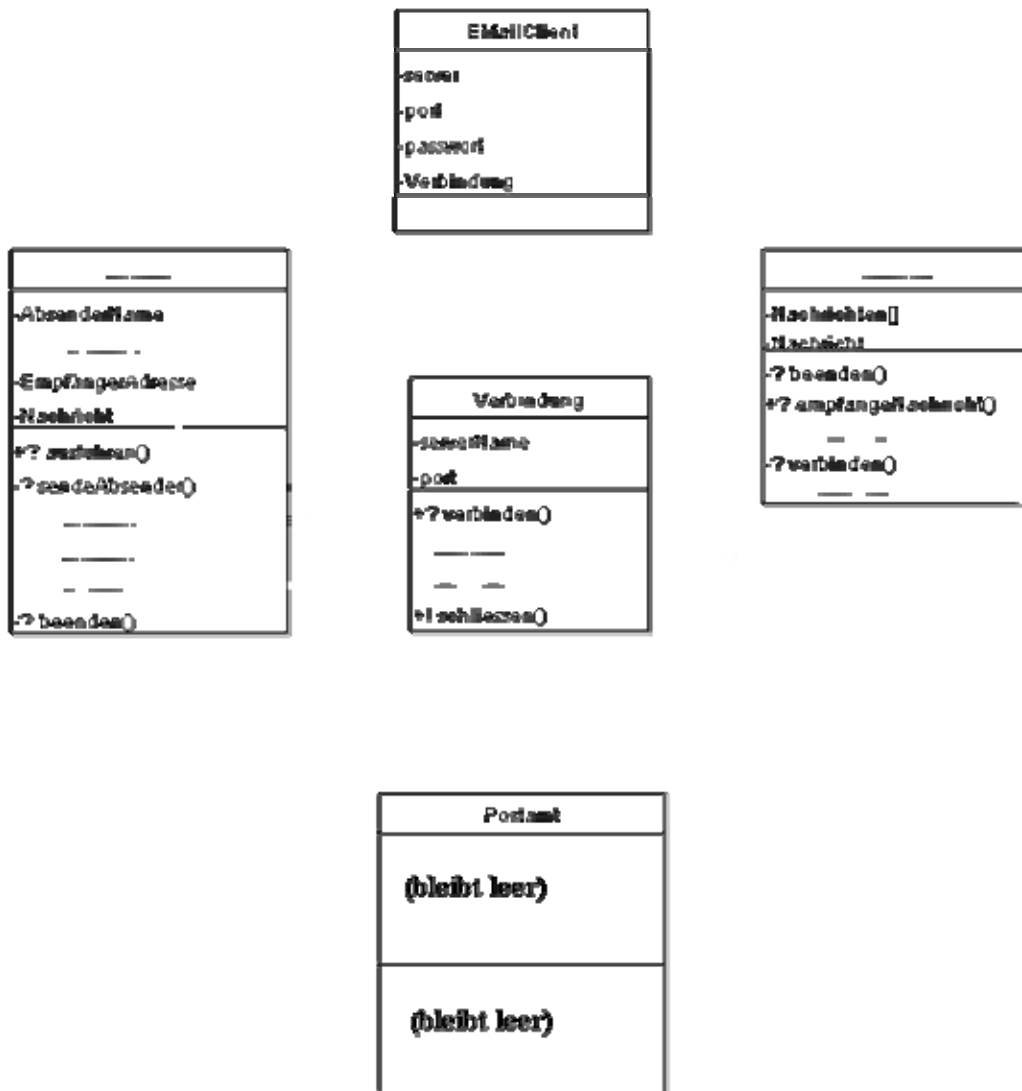
        if (!sendeNachricht()) fehler = true;
    }
// korrekter Zustand: _____ 6

// _____
return fehler;
}

```

Die korrekte Abfolge lautet also: _____ *c d b a f e*

Arbeitsblatt Beziehungs-Diagramm



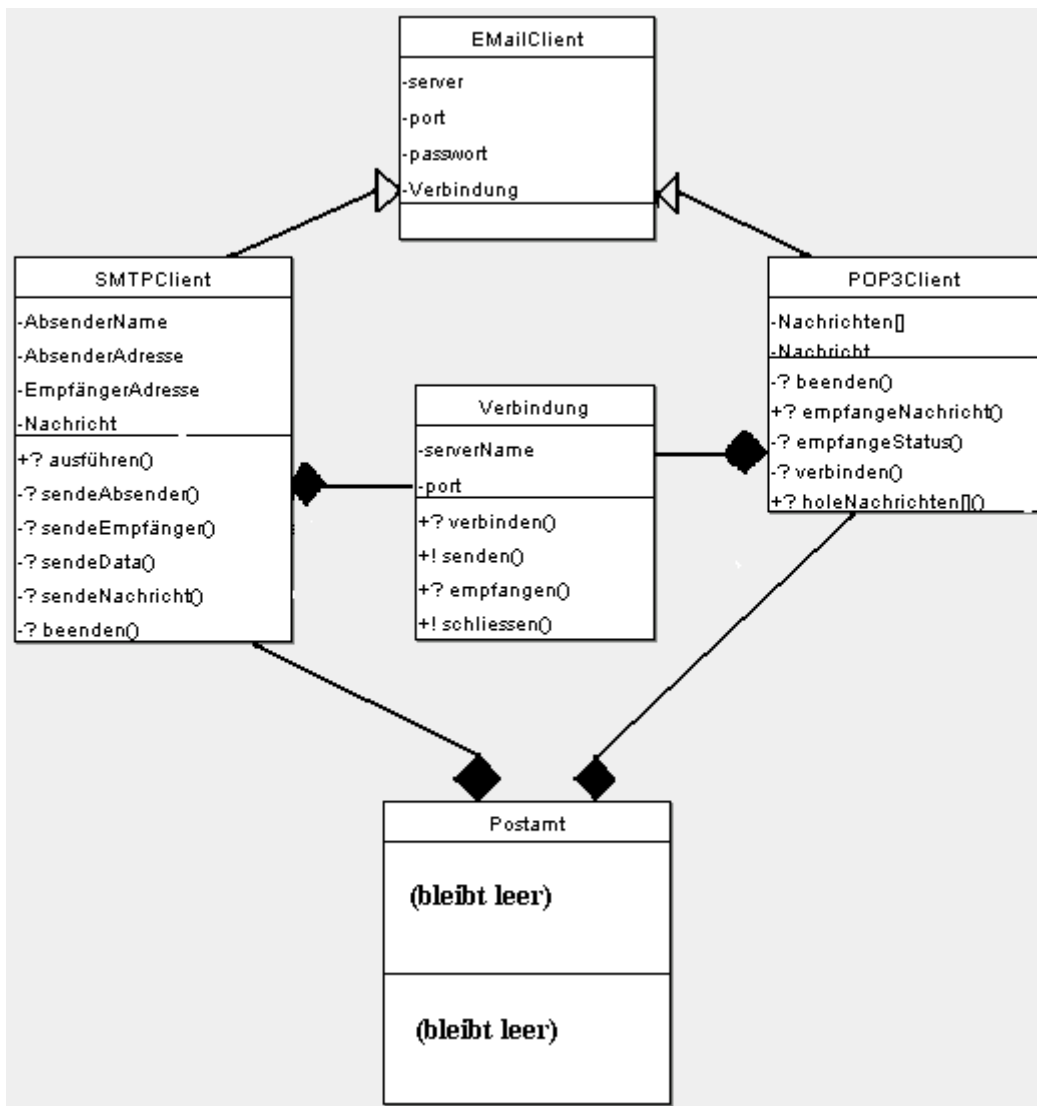
Bitte vervollständigen sie folgendes Beziehungsdiagramm.
 Die fehlenden Wörter stehen unten.
 Ergänzen Sie sinnvoll die „Beziehungspfeile“.

SMTPClient,
 -? empfangenStatus(),
 +! senden(),
POP3Client,
 -AbsenderAdresse,
 -? sendeNachricht(),
 -? sendeData,
 +? empfangen(),
 -? sendeEmpfänger,
 +? holeNachrichten()

Arbeitsblatt Beziehungs-Diagramm - Lösung

Bitte vervollständigen sie folgendes Beziehungsdiagramm.
 Die fehlenden Wörter stehen unten.

Ergänzen Sie sinnvoll die „Beziehungspfeile“.



SMTPClient,
 -? empfangenStatus(),
 +! senden(),
 POP3Client,
 -AbsenderAdresse,
 -? sendeNachricht(),
 -? sendeData,
 +? empfangen(),
 -? sendeEmpfänger,
 +? holeNachrichten()

Arbeitsblatt „Postamt“

1.) Füllen Sie die leeren Felder aus.

SMTP-Server	<input type="text"/>
Port	<input type="text"/>
	<input checked="" type="checkbox"/> Server benötigt Authentifizierung
<hr/>	
POP-Server	<input type="text"/>
Port	<input type="text"/>
	<input checked="" type="checkbox"/> Kopie auf dem Server belassen
<hr/>	
	<input checked="" type="checkbox"/> Nachrichten speichern

- 2.) Benutzen Sie das Programm, um Ihrem Tischnachbarn eine E-Mail zu schreiben.
- 3.) Überprüfen Sie mit Hilfe des Programms, ob Ihr Nachbar Ihnen eine E-Mail geschrieben hat.
- 4.) Was passiert, wenn Sie die Ports von SMTP und POP3 vertauschen?
- 5.) Beobachten Sie die Ausgaben des Programms im Fenster „Kommunikation“ beim versenden einer E-Mail und vergleichen Sie diese mit den Ausgaben des Programms beim abrufen von E-Mails.

Hausaufgabe

Der folgende Quelltext ist fehlerhaft. Die POP3-Befehle werden in einer falschen

Reihenfolge ausgeführt. Bitte sortieren Sie die Befehle so, dass der Client-PC korrekt eine Verbindung mit dem E-Mail-Server herstellen kann.
(Der fett gedruckte Bereich ist korrekt).

```
public boolean ausfuehren() {
    boolean fehler = false;
    int anzahl = 0;
    _____
    // Position a
    if (!fehler) {
        anzahl = empfangestatus();
        if (anzahl == -1) fehler = true;
    }
```

// korrekte Position: _____

```
    // Position b
    if (!verbinden()) fehler = true;
```

// korrekte Position: _____

```
    // Position c
    if (!fehler) {
        if (!authentifizieren()) fehler = true;;
    }
```

// korrekte Position: _____

```
    _____
    if (!fehler) {
        nachrichten = new String[anzahl];
        for ( int i=0; i< anzahl; i++ ) {
            nachrichten[i] = empfangenachricht(i+1);
        }
    }

    //
    if (!fehler) {
        if (!beenden()) fehler = true;
    }

    return !fehler;
}
```

Die korrekte Abfolge lautet also: _____

Hausaufgabe Lösung

Der folgende Quelltext ist fehlerhaft. Die POP3-Befehle werden in einer falschen

Reihenfolge ausgeführt. Bitte sortieren Sie die Befehle so, dass der Client-PC korrekt eine Verbindung mit dem E-Mail-Server herstellen kann.
(Der fett gedruckte Bereich ist korrekt).

```
public boolean ausfuehren() {  
    boolean fehler = false;  
    int anzahl = 0;  
  
    // Position a  
    if (!fehler) {  
        anzahl = empfangenStatus();  
        if (anzahl == -1) fehler = true;  
    }  
}
```

// korrekte Position: _____ 3

```
    // Position b  
    if (!verbinden()) fehler = true;
```

// korrekte Position: _____ 1

```
    // Position c  
    if (!fehler) {  
        if (!authentifizieren()) fehler = true;;  
    }  
}
```

// korrekte Position: _____ 2

```
    if (!fehler) {  
        nachrichten = new String[anzahl];  
        for ( int i=0; i< anzahl; i++ ) {  
            nachrichten[i] = empfangenNachricht(i+1);  
        }  
    }  
  
    //  
    if (!fehler) {  
        if (!beenden()) fehler = true;  
    }  
  
    return !fehler;  
}
```

Die korrekte Abfolge lautet also: _____ *b c a*