

# Abiturprüfung NRW 2018

## Informatik, Leistungskurs

### Aufgabenstellung:

An der Von-Neumann-Gesamtschule sollen die Termine für den kommenden Elternsprechtag mit Hilfe eines netzbasierten Informatiksystems organisiert werden. Die Termine mit den Lehrkräften können von den Eltern über einen Elternsprechtagserver gebucht werden.

Die buchbaren Termine sind getaktet: Für jeden Termin wird die gleiche, feste Dauer angesetzt. Die buchbaren Termine schließen direkt aneinander an. Die buchbaren Termine jeder Lehrkraft werden mit 0 beginnend durchnummeriert. Diese Nummerierung wird im Folgenden Termin-Nummer genannt. Termine mit gleicher Termin-Nummer finden immer zur selben Uhrzeit statt.

Lehrkraftkürzel: MD Name: Müller		Lehrkraftkürzel: MR Name: Meier		Lehrkraftkürzel: SR Name: Skröer	
TerminNr	Elternteil	TerminNr	Elternteil	TerminNr	Elternteil
0	"Kipp"	0	"Strehl"	0	
1		1		1	"Kipp"
2	"Mertens"	2	"Fehlhaber"	2	"Sprenger"
3		3		3	"Loos"
4	"Frieso"	4	"Karadag"	4	"Hoti"
5		5	"Windhuis"		
6	"Krause"	6			

Tabelle 1: Beispiel für die Terminbelegungen der Lehrkräfte

Die Kürzel der Lehrkräfte sind eindeutig. Die Elternteile erhalten Passwörter und eindeutige Benutzerkennungen. Elternteile können nur für sich selbst Termine bei einer Lehrkraft buchen und nur ihre eigenen Termine bei einer Lehrkraft löschen.

Auf dieser Grundlage wird folgendes Kommunikationsprotokoll vorgeschlagen:

Client sendet an Server	Server sendet an Client
ANMELDEN <Benutzerkennung> <Passwort>	+OK angemeldet <Benutzerkennung> -ERR Fehler bei Anmeldung
ABMELDEN	+OK abgemeldet Nach dem Senden trennt der Server die Verbindung.
BUCHEN <LehrkraftKuerzel> <TerminNr>	+OK Termin gebucht -ERR Terminbuchung nicht moeglich
LOESCHEN <LehrkraftKuerzel> <TerminNr>	+OK Termin geloesch -ERR Termin nicht vorhanden -ERR Terminloeschen verweigert
Anderer Text	-ERR unbekannter Befehl

Tabelle 2: Kommunikationsprotokoll

- a) Im Folgenden ist eine Kommunikation zwischen einem Client und dem Server gegeben. Die Lehrkräfte mit den Namen Mueller (Kürzel: MD) und Meier (Kürzel: MR) existieren in der Datenbasis des Servers.

Der Elternteil Hannelore Wegener mit der Benutzerkennung HWegener und dem Passwort Ru135# möchte folgende Aktionen ausführen:

- i. Frau Wegener möchte sich (erfolgreich) anmelden.
- ii. Sie möchte einen Termin bei Mueller für Termin-Nummer 1 buchen.
- iii. Sie möchte einen Termin bei Meier für Termin-Nummer 4 buchen.  
Dieser Termin ist jedoch schon anderweitig vergeben.
- iv. Sie möchte den Termin bei Meier mit Termin-Nummer 4 löschen.

*Stellen Sie die Kommunikation zwischen Client und Server gemäß den Vorgaben aus Tabelle 2 dar.*

(6 Punkte)

Eine Teilmodellierung in Form eines Implementationsdiagramms finden Sie in Abbildung 1. Die Verwaltung der Termine wird in den Klassen Elternteil und Lehrkraft im Sinne der Datenkapselung intern korrekt gelöst und nicht erläutert. Die Attribute d, h und m der Klasse ElternsprechtagServer sind Bestandteil von Aufgabenteil c).

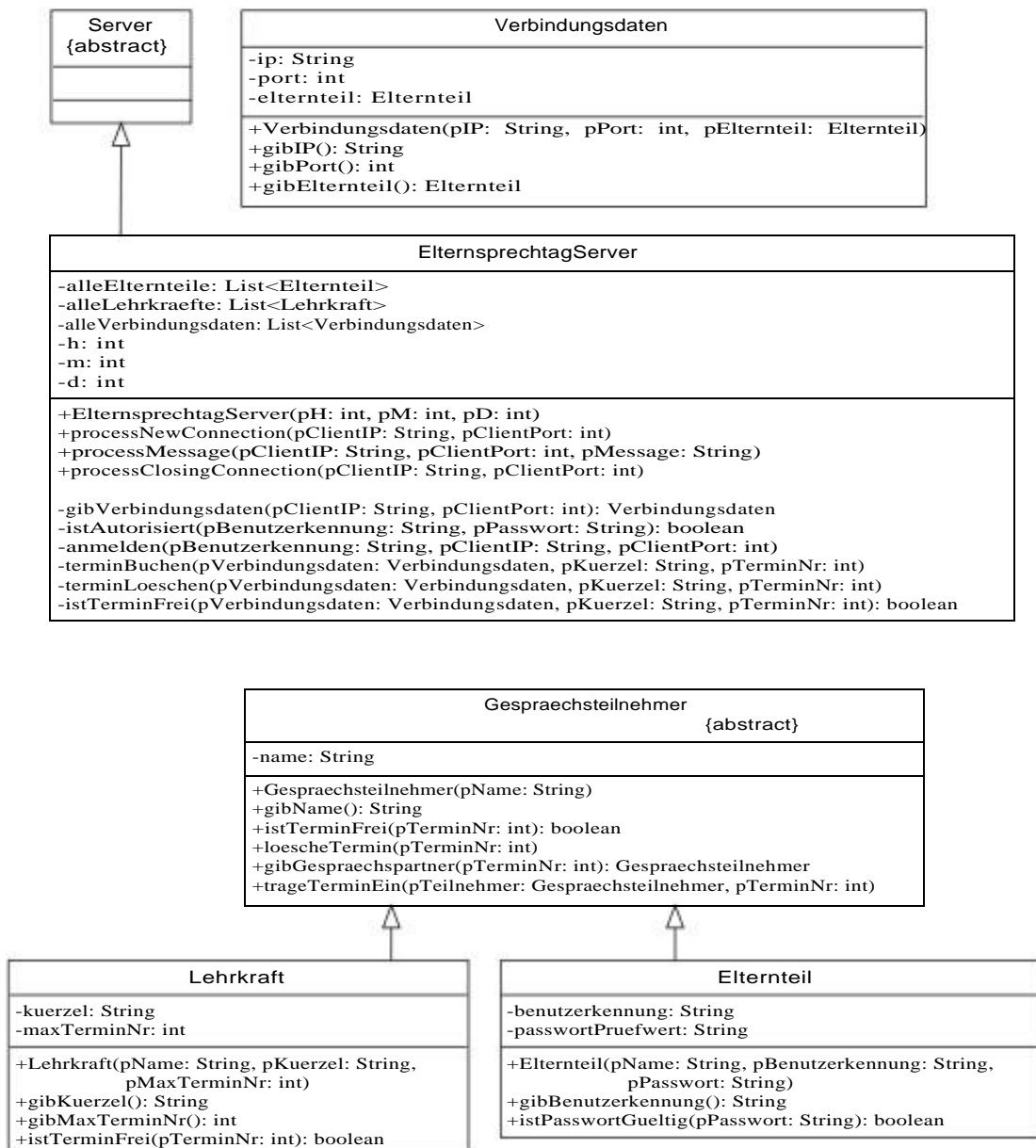


Abbildung 1: Teilmodellierung in Form eines Implementationsdiagramms

b) Analysieren Sie das Implementationsdiagramm aus Abbildung 1 sowie die Dokumentation der Klasse ElternsprechtagServer im Anhang und erläutern Sie die Umsetzung der Anfragen des Clients an den Server aus Tabelle 2 im Modell.

(9 Punkte)

- c) In der Klasse ElternsprechtagServer ist folgende undokumentierte Methode enthalten:

```
1 private String ermittleEtwas(String pParameter) {
2     String ergebnis = "";
3     alleLehrkraefte.toFirst();
4     Lehrkraft aktuell = null;
5     while (alleLehrkraefte.hasAccess()) {
6         if (alleLehrkraefte.getContent()
7             .gibKuerzel().equals(pParameter)) {
8             aktuell = alleLehrkraefte.getContent();
9         }
10        alleLehrkraefte.next();
11    }
12    if (aktuell != null) {
13        for (int i = 0; i <= aktuell.gibMaxTerminNr(); i++) {
14            if (!aktuell.istTerminFrei(i)) {
15                ergebnis = ergebnis
16                    + aktuell.gibGespraechspartner(i).gibName()
17                    + "-";
18                ergebnis = ergebnis + (h + (i * d + m) / 60)
19                    + ":";
20                ergebnis = ergebnis + ((m + i * d) % 60) + "|";
21            }
22        }
23    }
24    return ergebnis;
25 }
```

Beispieldaten zur Analyse der Methode ermittleEtwas:

Wert des Parameters:     pParameter = "MR"

Werte der Attribute:     d = 10, h = 14, m = 40

Die durch das Attribut alleLehrkraefte referenzierten Daten finden sich in Tabelle 1. Gehen Sie in dieser Teilaufgabe davon aus, dass in der Liste mit dem Bezeichner alleLehrkraefte nur die dort genannten Lehrkräfte stehen.

*Analysieren Sie die Methode ermittleEtwas, indem Sie die Methode auf die Beispieldaten anwenden und die Abschnitte von Zeile 3 bis Zeile 10 sowie Zeile 11 bis Zeile 19 erläutern.*

*Geben Sie die Rückgabe der Methode an.*

*Erläutern Sie die Funktionalität der Methode im Sachkontext.*

(13 Punkte)

- d) Die Klasse ElternsprechtagServer soll um eine Methode sendeAusgelastete ergänzt werden. Diese sendet eine Antwort an den entsprechenden Client, in der eine Auswahl der Lehrkräfte aufgelistet ist: Dabei werden die Namen derjenigen Lehrkräfte gesendet, deren Belegungsquote mindestens so hoch ist wie der übergebene Wert des Parameters pQuote. Die Belegungsquote wird berechnet, indem für die jeweilige Lehrkraft die Anzahl der belegten Termine durch die Gesamtzahl ihrer möglichen Termine geteilt wird. Beispielsweise ist die Belegungsquote der Lehrkraft Skröer (siehe Tabelle 1) 0.8.

An den Client soll nur eine einzelne Nachricht gesendet werden, die mit "+OK" beginnt und danach die Namen - durch Kommata getrennt - enthält.

Wird eine Quote kleiner als 0.0 oder größer als 1.0 übergeben, wird nicht gesendet. Verweist der Parameter pVerbindungsdaten auf null, wird ebenfalls nicht gesendet.

Die Methode hat den folgenden Methodenkopf:

```
private void sendeAusgelastete(  
    Verbindungsdaten pVerbindungsdaten, double pQuote)
```

*Implementieren Sie die Methode sendeAusgelastete.*

*Dokumentieren Sie Ihren Quellcode durch geeignete Kommentare.*

(14 Punkte)

- e) Die Schule möchte den bestehenden Elternsprechtagserver für weitere Zwecke verwenden. Sie möchte von sich aus in folgenden Fällen den Elternteilen Nachrichten zukommen lassen:

Erstens sollen die Elternteile nach dem ersten Anmelden einmalig eine Nachricht angezeigt bekommen, die Informationen zum Datenschutz und zur Bedienung enthält.

Zweitens sollen betroffene Elternteile kurzfristig darüber informiert werden, dass Lehrkräfte krank sind und daher ihre Termine nicht wahrnehmen können. Dazu soll eine Nachricht vom Server zu den Clients der Eltern gesendet werden.

*Beurteilen Sie, inwiefern das aktuelle Elternsprechtagssystem geeignet ist, die beiden Erweiterungen umzusetzen.*

(8 Punkte)

## Anhang: Dokumentationen der verwendeten Klassen

### Die Klasse ElternsprechtagServer

Diese Klasse ist Unterklasse der Klasse Server. Objekte dieser Klasse verwalten die Elternteile, die Lehrkräfte und die Verbindungsdaten der Elternteile. Dabei stellt sie neben den geerbten Methoden unter anderem folgende Methoden zur Verfügung:

**Konstruktor** ElternsprechtagServer(int pH, int pM, int pD)

Die übergebenen Werte werden gespeichert. Die Bedeutung von pH, pM und pD in Bezug auf die Attribute sind Bestandteil von Aufgabenteil c).

**Auftrag** void processNewConnection(String pClientIP,  
int pClientPort)

Diese Ereignisbehandlungsmethode wird aufgerufen, wenn sich ein Client mit IP-Adresse pClientIP und Portnummer pClientPort mit dem Server verbunden hat. Der Aufruf der Methode erfolgt nicht synchronisiert.

```
Auftrag void processMessage(String pClientIP,
                           int pClientPort, String pMessage)
```

Diese Ereignisbehandlungsmethode wird aufgerufen, wenn der Server die Nachricht pMessage von dem durch pClientIP und pClientPort spezifizierten Client empfangen hat. Der vom Client hinzugefügte Zeilentrenner wurde zuvor entfernt. Der Aufruf der Methode erfolgt nicht synchronisiert. Die Methode organisiert gemäß dem Protokoll die Antwort des Servers an den Client. Sie sendet dem Client gemäß dem Protokoll eine Nachricht, ob die Anmeldung erfolgreich war.

Auftrag	void processClosingConnection(String pClientIP, int pClientPort)
---------	---

Sofern der Server die Verbindung zu dem durch pClientIP und pClientPort spezifizierten Client trennt, wird diese Ereignisbehandlungsmethode aufgerufen, unmittelbar *bevor* die Verbindungstrennung tatsächlich erfolgt. Wird die Verbindung unvermittelt unterbrochen oder hat der in der Parameterliste spezifizierte Client die Verbindung zum Server unvermittelt getrennt, erfolgt der Methodenaufruf *nach* der Unterbrechung/Trennung der Verbindung. Der Aufruf der Methode erfolgt nicht synchronisiert.

Die gespeicherten Verbindungsdaten zu dem Elternteil werden gelöscht. Dem Client wird die entsprechende Nachricht gesendet.

```
Anfrage Verbindungsdaten gibVerbindungsdaten(
    String pClientIP, int pPort)
```

Das Objekt der Klasse Verbindungsdaten mit den übergebenen Daten wird zurückgegeben. Wenn dieses nicht existiert, wird null zurückgeliefert.

```
Anfrage boolean istAutorisiert(String pBenutzerkennung,  
                                String pPasswort)
```

Die Anfrage liefert true, wenn der Elternteil mit dem Benutzernamen pBenutzerkennung das korrekte Passwort eingegeben hat. Sonst wird false zurückgeliefert.

```
Auftrag void anmelden (String pBenutzerkennung,
                        String pClientIP, int pClientPort)
```

Das Elternteil-Objekt mit der Benutzerkennung pBenutzerkennung wird mit der IP-Adresse pClientIP und der Port-Nummer pClientPort angemeldet. Ein neues Verbindungsdaten-Objekt mit den Informationen pClientIP, pClientPort und der Assoziation auf das zugehörige Elternteil-Objekt wird initialisiert und gespeichert.

```
Auftrag void terminBuchen(Verbindungsdaten pVerbindungsdaten,  
                          String pKuerzel, int pTerminNr)
```

Bei dem zugehörigen Elternteil und der Lehrkraft mit dem Kürzel pKuerzel wird der Termin für die angegebene Termin-Nummer gespeichert. Sind die Verbindungsdaten leer, existiert das übergebene Kürzel nicht oder ist die Termin-Nummer negativ, so geschieht nichts, ansonsten sendet die Methode gemäß dem Protokoll die Antwort an den Client.

```
Auftrag void terminLoeschen(Verbindungsdaten pVerbindungsdaten,  
                           String pKuerzel, int pTerminNr)
```

Bei dem zugehörigen Elternteil und der Lehrkraft mit dem Kürzel pKuerzel wird der Termin für die angegebene Termin-Nummer gelöscht. Sind die Verbindungsdaten leer, existiert das übergebene Kürzel nicht oder ist die Termin-Nummer negativ, so geschieht nichts, ansonsten sendet die Methode gemäß dem Protokoll die Antwort an den Client.

Sind die eingetragenen Gesprächspartner nicht mit den übergebenen Daten identisch, so geschieht nichts, ansonsten sendet die Methode gemäß dem Protokoll die Antwort an den Client.

```
Anfrage boolean istTerminFrei(Verbindungsdaten pVerbindungsdaten,  
                             String pKuerzel, int pTerminNr)
```

Wenn bei dem Elternteil, der zu den Verbindungsdaten gehört, und der Lehrkraft mit dem Kürzel pKuerzel ein Termin für die angegebene Termin-Nummer frei ist, dann wird true und sonst false zurückgegeben.

## Die Klasse **Gespraechsteilnehmer**

Diese abstrakte Klasse dient zur Verwaltung der Termine von Personen. Objekte dieser Klassen verwalten den Namen eines Gesprächsteilnehmers sowie die Gesprächspartner an den entsprechenden Termin-Nummern.

```
Konstruktor  GespraechstTeilnehmer(String pName)
```

Der Name des Gesprächsteilnehmers wird in dem neuen Objekt gespeichert. Anfrage

```
String gibName()
```

Die Methode liefert den gespeicherten Namen.

```
Anfrage    boolean istTerminFrei(int pTerminNr)
```

Die Methode liefert den Wert `true`, wenn an der übergebenen Termin-Nummer kein Gesprächspartner eingetragen ist und die Terminnummer nicht negativ ist. Ansonsten liefert sie `false`.

Auftrag      void loescheTermin(int TerminNr)

Der Termin an der angegebenen Termin-Nummer wird gelöscht.

```
Anfrage      GespraechstTeilnehmer gibGespraechspartner(int TerminNr)
```

Die Methode liefert das Objekt der Klasse `Gespraechsteilnehmer`, die an der Termin-Nummer als Gesprächspartner gespeichert ist. Ist der Termin nicht belegt oder ist die Termin-Nummer negativ, so wird null zurückgegeben.

```
Auftrag      trageTerminEin(Gespraechsteilnehmer pTeilnehmer,
                                     int TerminNr)
```

An der entsprechenden Termin-Nummer wird der übergebene Gesprächsteilnehmer als Gesprächspartner eingetragen, sofern die Termin-Nummer nicht negativ ist.

## Die Klasse Lehrkraft

Diese Klasse dient zur Verwaltung der Lehrkräfte und ihrer Termine. Sie spezialisiert die Klasse `Gespraechsteilnehmer`. Objekte dieser Klasse verwalten das Kürzel der Lehrkraft und die höchste Termin-Nummer.

[illegible]

Die übergebenen Daten werden in dem neuen Objekt gespeichert.

Anfrage      String gibKuerzel()

Die Methode liefert das gespeicherte Kürzel.

Anfrage      `int gibMaxTerminNr()`

Die Methode liefert die Termin-Nummer des letzten Termins, an dem die Lehrkraft für ein Gespräch zur Verfügung steht.

Anfrage      **boolean istTerminFrei(int pTerminNr)**

Die Methode liefert den Wert true, wenn die übergebene Termin-Nummer kleiner oder gleich der maximalen Termin-Nummer und kein Gesprächspartner eingetragen ist, ansonsten liefert sie false.

## **Die Klasse Elternteil**

Diese Klasse dient zur Verwaltung der Elternteile und ihrer Termine. Sie spezialisiert die Klasse Gesprächsteilnehmer. Objekte dieser Klasse verwalten die Benutzerkennung und einen Passwortprüfwert.

**Konstruktor Elternteil(String pName, String pBenutzerkennung,  
String pPasswort)**

Die übergebenen Daten werden in dem neuen Objekt gespeichert. Das Passwort wird mithilfe von kryptografischen Verfahren nicht im Klartext gespeichert.

Anfrage      **String gibBenutzerkennung()**

Die Methode liefert die gespeicherte Benutzerkennung.

Anfrage      **boolean istPasswortGueltig(String pPasswort)**

Die Methode liefert true, wenn das übergebene Passwort korrekt ist, ansonsten liefert es false.

## **Die Klasse Verbindungsdaten**

Diese Klasse dient zur Verwaltung der Verbindungsdaten. Objekte dieser Klassen verwalten die Client-IP, den Client-Port und den zugehörigen angemeldeten Elternteil.

**Konstruktor Verbindungsdaten(String pIP, int pPort,  
Elternteil pElternteil)**

Die übergebenen Daten werden gespeichert.

Anfrage      **String gibIP()**

Die Methode liefert die gespeicherte IP.

Anfrage      **int gibPort()**

Die Methode liefert den gespeicherten Port.

Anfrage      **Elternteil gibElternteil()**

Die Methode liefert eine Referenz auf das gespeicherte Elternteil.