

Klausur Programmierung II

Datum:

Name: Vorname:

Matr.-Nr.: letzter Versuch: Ja Nein

Bitte zuerst den Kopf dieses Blattes vollständig und leserlich ausfüllen. Lassen Sie bei der Bearbeitung der Aufgaben die Blätter zusammen geheftet. Bitte schreiben Sie die Lösungen unter den jeweiligen Aufgabentext. Benutzen Sie kein eigenes Papier (außer zum Vorschreiben).

Viel Erfolg!

Aufgabe	Mögliche Punkte	Erstkorrektur	Zweitkorrektur
Aufgabe 1 : Zweidimensionale Arrays	8		
Aufgabe 2: Exceptions	19		
Aufgabe 3: Datei Ein-/Ausgabe	19		
Aufgabe 4: GUI 1	28		
Aufgabe 5: GUI 2	26		
Punkte Gesamt	100		
Note			

Aufgabe 1: Zweidimensionale Arrays

In der Klasse *Stundenplan* werden die Doppelstunden einer Woche (Mo-Sa) in einem zweidimensionalen Array vom Typ *Doppelstunde* verwaltet (siehe Blatt 1). Das Stundenplanraster sieht ANZAHL_TAGE und ANZAHL_DOPPELSTUNDEN pro Tag vor. Schreiben Sie den Konstruktor für die Klasse *Stundenplan* (siehe Blatt 2). Hier soll der Stundenplan zunächst mit neu erzeugten *Doppelstunde*-Objekten gefüllt werden, wobei alle Doppelstunden eines Tages in einer Zeile des Arrays eingetragen werden sollen. (8 Punkte)

```
public class Stundenplan {
    public static int ANZAHL_DOPPELSTUNDEN = 7;
    public static int ANZAHL_TAGE = 6;
    private Doppelstunde[][] woche;

    public Stundenplan() {

    }
    // hier steht der Rest der Klasse ....
}
```

Aufgabe 2: Exceptions

a) Schreiben Sie die Methode `setDoppelstunde(int tag, int nr, Doppelstunde ds)` der Klasse *Stundenplan*, die das übergebene *Doppelstunde*-Objekt am angegebenen Tag zur angegebenen Stunde im Stundenplan setzt. Die Werte von *tag* und *nr* beziehen sich auf die Indizes des Arrays im Attribut *woche*. Die übergebenen Argumente *tag* und *nr* sollen allerdings vorher auf Gültigkeit geprüft werden, damit beim Zugriff auf das Array keine *IndexOutOfBoundsException* auftritt, der Parameter *ds* sollte auf ein Objekt verweisen. Ist einer der übergebenen Werte ungültig, so soll eine *IllegalArgumentException* geworfen werden. (7 Punkte)

b) Schreiben Sie für die Klasse *Doppelstunde* eine Methode *setModullInfo*, die es erlaubt, die folgenden Infos in einer Doppelstunde zu setzen: *bezeichner*, *kuerzel*, *dozent*. Prüfen Sie im Rumpf der Methode, ob einer der übergebenen Strings ein leerer String ist (bzw. ein String mit ausschließlich Leerzeichen). Falls ja, soll eine geprüfte Exception vom Typ *EingabeException* geworfen werden (siehe Blatt 4). Dem Konstruktor der Exception soll eine Message übergeben werden, die Auskunft darüber gibt, welcher Parameter falsch war (z.B. „Falsche Angabe für den Bezeichner“). (12 Punkte)

Aufgabe 3: Datei Ein-/Ausgabe

Schreiben Sie die Methode `laden(File datei)` der Klasse *Stundenplan*. Diese soll aus der übergebenen Textdatei alle Doppelstunden-Infos eines Stundenplans einlesen und den Stundenplan entsprechend aktualisieren. Jede Zeile der Textdatei enthält die Informationen für eine Doppelstunde im folgenden Format:

TagNr:DoppelstundeNr:Raum:Bezeichner:Kuerzel:Dozent

Alle Doppelstunden eines Stundenplans sind in der Textdatei in der folgenden Reihenfolge abgelegt: alle Doppelstunden des ersten Tages von 0 bis 6, alle Doppelstunden des zweiten Tages von 0 bis 6 usw. Die Nummerierung der Tage beginnt bei 0. (19 Punkte)

Aufgabe 4: GUI 1

Die Klasse *StundenplanView* zeigt ein Raster von Buttons, einen für jede Doppelstunde eines Tages. Die Buttons tragen als Aufschrift die Kurzbeschreibung der zugehörigen Doppelstunde und werden in dem zweidimensionalen Array *buttons* gesammelt.

Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag
08:00 - 09:30 <k.K> <k.R.>	08:00 - 09:30 <k.K> <k.R.>	08:00 - 09:30 <k.K> <k.R.>	08:00 - 09:30 <k.K> <k.R.>	08:00 - 09:30 <k.K> <k.R.>	08:00 - 09:30 <k.K> <k.R.>
10:00 - 11:30 <k.K> <k.R.>	10:00 - 11:30 <k.K> <k.R.>	10:00 - 11:30 <k.K> <k.R.>	10:00 - 11:30 <k.K> <k.R.>	10:00 - 11:30 <k.K> <k.R.>	10:00 - 11:30 <k.K> <k.R.>
12:15 - 13:45 <k.K> <k.R.>	12:15 - 13:45 <k.K> <k.R.>	12:15 - 13:45 <k.K> <k.R.>	12:15 - 13:45 <k.K> <k.R.>	12:15 - 13:45 <k.K> <k.R.>	12:15 - 13:45 <k.K> <k.R.>
14:15 - 15:45 <k.K> <k.R.>	14:15 - 15:45 <k.K> <k.R.>	14:15 - 15:45 <k.K> <k.R.>	14:15 - 15:45 <k.K> <k.R.>	14:15 - 15:45 <k.K> <k.R.>	14:15 - 15:45 <k.K> <k.R.>
16:00 - 17:30 <k.K> <k.R.>	16:00 - 17:30 <k.K> <k.R.>	16:00 - 17:30 <k.K> <k.R.>	16:00 - 17:30 <k.K> <k.R.>	16:00 - 17:30 <k.K> <k.R.>	16:00 - 17:30 <k.K> <k.R.>
17:45 - 19:15 <k.K> <k.R.>	17:45 - 19:15 <k.K> <k.R.>	17:45 - 19:15 <k.K> <k.R.>	17:45 - 19:15 <k.K> <k.R.>	17:45 - 19:15 <k.K> <k.R.>	17:45 - 19:15 <k.K> <k.R.>
19:30 - 21:00 <k.K> <k.R.>	19:30 - 21:00 <k.K> <k.R.>	19:30 - 21:00 <k.K> <k.R.>	19:30 - 21:00 <k.K> <k.R.>	19:30 - 21:00 <k.K> <k.R.>	19:30 - 21:00 <k.K> <k.R.>

```
import javax.swing.JFrame;

...

public class StundenplanView {
    private Stundenplan plan;
    private JFrame fenster;
    private JButton[][] buttons;

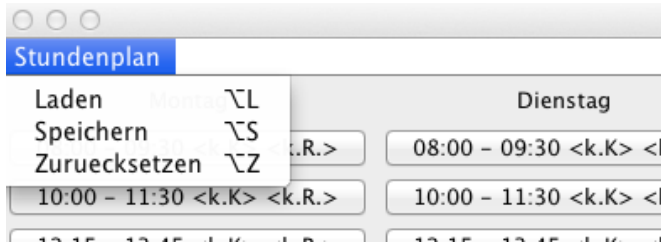
    public StundenplanView()
    {
        this.plan = new Stundenplan();
        buttons =
            new JButton [plan.ANZAHL_DOPPELSTUNDEN][plan.ANZAHL_TAGE];
        fenster = new JFrame("Stundenplan");
        fenster.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        initialisiere();
        menueHinzufuegen();
        fenster.setVisible(true);
    }

    private void menueHinzufuegen() {
        // Das machen Sie (Aufgabe 4a)
    }

    private void buttonsAktualisieren() {
        // Setzt die Beschriftung auf den Buttons gemäss der
        // Kurzbeschreibung der Doppelstunden in plan
    }

    public static void main(String[] args) {
        new StundenplanView();
    }
}
```

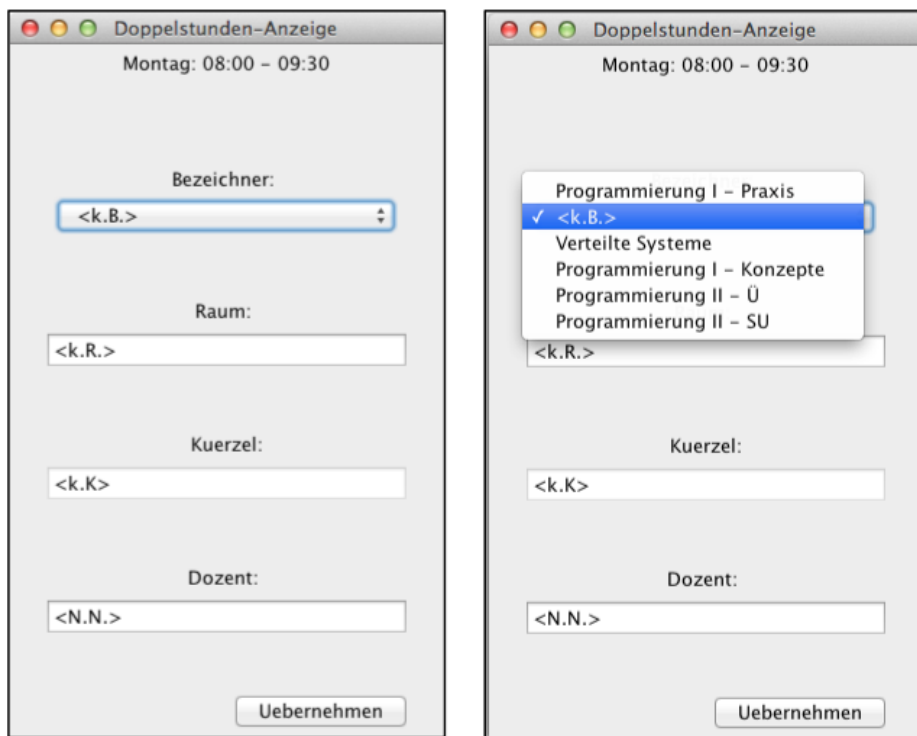
a) Schreiben Sie die Methode `menueHinzufuegen()`, die dem Fenster der *StundenplanView* das unten eingeblendete Menü hinzufügt. Es reicht, wenn sie dem Menü nur das Menü-Item „Laden“ hinzufügen, die anderen Items können Sie weglassen. Als Listener soll eine Instanz der Klasse *MenueListener* gesetzt werden. Die Instanz können sie mit dem Standard-Konstruktor erzeugen. (12 Punkte)



b) Ergänzen Sie die Klasse *StundenplanView* um die innere Klasse *MenueListener*, die auf die Menü-Aktionen des Fensters reagiert. Behandeln Sie hier nur die Aktion „Laden“, die anderen Aktionen können weggelassen werden. Es soll ein *JFileChooser* geöffnet werden, um die Datei auszuwählen, aus der geladen werden soll. Der Stundenplan soll mit den eingelesenen Daten belegt werden und auf den Buttons muss als Beschriftung die Kurzbeschreibung der zugehörigen Doppelstunde des Stundenplans neu gesetzt werden. Nutzen Sie hierfür die entsprechenden Methoden der Klassen *Stundenplan* und *StundenplanView*. (16 Punkte)

Aufgabe 5: GUI 2

Die Klasse *StundenView* öffnet ein Eingabefenster, das die Infos einer Doppelstunde anzeigt und bearbeiten lässt. Das Eingabefenster wird geöffnet, wenn der Nutzer einen Button in der *StundenplanView* (Aufgabe 4) drückt. Das Fenster zeigt anfangs die aktuellen Infos der ausgewählten Doppelstunde in den Eingabefeldern an.



JComboBox für die Auswahl der Modulbezeichner, das Kuerzel-Textfeld stellt sich automatisch ein.

Die Modulbezeichner werden über eine *JComboBox* zur Verfügung gestellt. Wird ein Modulbezeichner ausgewählt, so stellt sich das Textfeld für das Modulkürzel automatisch auf das entsprechende Kürzel ein. Das Textfeld für das Modulkürzel soll daher nicht editierbar sein.

Die Bezeichner, mit denen die *JComboBox* initialisiert wird, befinden sich in dem Array *bezeichner*. Die *HashMap* *bezeichnerMap* speichert zu jedem Bezeichner (key) das zugehörige Kürzel (value). Die genaue Funktionsweise der *JComboBox* müssen Sie für diese Aufgabe nicht kennen.

Drückt man den Button „Uebernehmen“, so sollen die im Eingabefenster eingegebenen Werte in der entsprechenden Doppelstunde (Attribut *stunde*) gesetzt werden.

Studieren Sie zunächst gründlich die Klasse *StundenView* (siehe Blatt 3), damit Sie wissen, was hier bereits zur Verfügung gestellt wird.

a) Fügen Sie der Methode `initialisiere()` exemplarisch den Code für das Hinzufügen eines Labels sowie eines Textfelds für das Anzeigen des Modulkürzels hinzu. Das Textfeld soll nicht editierbar sein. Die anderen Eingabefelder können Sie weglassen. Fügen Sie außerdem den Button hinzu. Der Button soll mit einer Instanz der Klasse *ButtonListener* verknüpft werden (siehe Blatt 3). (13 Punkte)

```
private void initialisiere() {  
    ... // für die restlichen Infos siehe Klasse StundenView
```

```
}
```

b) Schreiben Sie die Methode `uebernehmen()`. Diese wird vom *ButtonListener* aufgerufen wenn man auf den Button mit der Aufschrift „Uebernehmen“ gedrückt hat. Die Doppelstunde im Attribut *stunde* soll daraufhin mit den Infos des Fensters gefüllt werden, falls alle Angaben o.k. waren. Fangen Sie die dabei möglicherweise auftretende Exception und zeigen Sie deren Nachricht in einer passenden *JOptionPane* an.

(13 Punkte)

Tipp: Bei einer *JComboBox* kann man auf das aktuell selektierte Objekt mit Hilfe der folgenden Methode zugreifen: `Object getSelectedItem()`

```

public class Doppelstunde {
    private String von;
    private String bis;
    private int tag;
    private String raum;
    private String bezeichner;
    private String kuerzel;
    private String dozent;

    /** @param nr bezeichnet die Position der Doppelstunde am Tag (1., 2.
     * usw.) wobei hier mit der Nummerierung bei 0 begonnen wird.
     */
    public Doppelstunde(int tag, int nr) {
        setVonBis(nr);
        this.tag = tag;
        this.raum = "<k.R.>";
        this.bezeichner = "<k.B.>";
        this.kuerzel = "<k.K.>";
        this.dozent = "<N.N.>";
    }

    public void setModulInfo(String bez, String kuerz, String doz) ...
        // Das machen Sie (Aufgabe 2)

    private void setVonBis(int nr){
        switch(nr) {
            case 0: von = "08:00"; bis = "09:30"; break;
            case 1: von = "10:00"; bis = "11:30"; break;
            case 2: von = "12:15"; bis = "13:45"; break;
            case 3: von = "14:15"; bis = "15:45"; break;
            case 4: von = "16:00"; bis = "17:30"; break;
            case 5: von = "17:45"; bis = "19:15"; break;
            case 6: von = "19:30"; bis = "21:00"; break;
        }
    }

    public String getTag() {
        return Stundenplan.bezeichnerFuerTag(tag);
    }

    // Die getter- und setter-Methoden für die Attribute von, bis, raum,
    // bezeichner, kuerzel, dozent stehen wie üblich zur Verfügung.

    // Rest siehe nächste Seite

```

```

public String getKurzbeschreibung() {
    String ausgabe = "";
    ausgabe = von + " - " + bis + " "+ kuerzel + " " + raum;
    return ausgabe;
}
}

```

Blatt 2

```

import java.io.File;
...

public class Stundenplan {
    public static int ANZAHL_DOPPELSTUNDEN = 7;
    public static int ANZAHL_TAGE = 6;
    private Doppelstunde[][] woche;

    public Stundenplan() {
        // das machen Sie (Aufgabe 1)
    }

    public void setDoppelstunde(int tag, int nr,
                                Doppelstunde ds) {
        // das machen Sie (Aufgabe 2)
    }

    public Doppelstunde getDoppelstunde(int tag, int nr) {
        // liefert die entsprechende Doppelstunde
    }

    public static String bezeichnerFuerTag(int t) {
        String text = "nix";
        switch(t) {
            case 0: text = "Montag"; break;
            case 1: text = "Dienstag"; break;
            case 2: text = "Mittwoch"; break;
            case 3: text = "Donnerstag"; break;
            case 4: text = "Freitag"; break;
            case 5: text = "Samstag"; break;
        }
        return text;
    }

    public void speichern(File datei) {
        // das machen Sie nicht ;- )
    }

    public void laden(File datei) {
        // das machen Sie (Aufgabe 3)
    }
}

```

```

import java.awt.Dimension;
...

public class StundenView {
    Doppelstunde stunde;
    JButton button;
    JFrame fenster;
    JTextField raumTF;
    JComboBox auswahlBezeichner;
    JTextField kuerzelTF;
    JTextField dozentTF;
    HashMap<String, String> bezeichnerMap;
    Object[] bezeichner;

    public StundenView(Doppelstunde stunde) {
        bezeichnerMap = new HashMap<String, String>();
        mapFuellen();
        this.stunde = stunde;
        fenster = new JFrame("Doppelstunden-Anzeige");
        initialisiere();
        fenster.setSize(new Dimension(300, 500));
        fenster.setVisible(true);
    }

    private void mapFuellen() {
        // Hier wird die HashMap und das Array für die JComboBox gefüllt.
        // Die Details sind für Sie unwichtig.
    }

    private void initialisiere() {
        BorderLayout layout =
            new BorderLayout(fenster.getContentPane(), BorderLayout.Y_AXIS);
        fenster.setLayout(layout);
        JPanel panel = new JPanel();
        String text = stunde.getTag() + ": " + stunde.getVon() + " - " +
            stunde.getBis();
        JLabel label = new JLabel(text);
        panel.add(label);
        fenster.add(panel);

        // hier kommt Ihr Code hinein (Aufgabe 5).
    }

    private void uebernehmen() {
        // // Das machen Sie (Aufgabe 5)

    } // weiter nächste Seite

```

```
private class ButtonListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        String action = e.getActionCommand();  
        if(action.equals("Uebernehmen")) uebernehmen();  
    }  
}
```

```
public class EingabeException extends Exception {  
    EingabeException(String message) {  
        super(message);  
    }  
}
```