

# Prüfung

## User Interaction (UI)

Prof. Dr.-Ing. Christian Heller <christian.heller@ba-sachsen.de>

<b>Student</b>	
Vor- und Nachname	Ihre Daten werden von der Klausuraufsichtsperson schriftlich auf der Anwesenheitsliste festgehalten.
Matrikelnummer	
Studienrichtung und Jahr	CS19-1
Anmeldename	Das Login "klaus..." muss unbedingt schriftlich auf der Anwesenheitsliste festgehalten werden, da sonst keine Zuordnung des Logins zu Ihrem Namen und damit keine Korrektur der Klausur möglich ist!

<b>Prüfung</b>	
Datum	Dezember 2020
Dauer [min]	100
Hilfsmittel	<p>Dokumentation im lokalen Netzwerk (Intranet) sowie Recherche im Internet.          NICHT gestattet:</p> <ul style="list-style-type: none"> <li>* Kommunikation in jeglicher Form</li> <li>* Anmeldung via SSH auf dem Rechner "fileserv"</li> <li>* Anmeldung mit Klausur-Login nach Ende der Prüfung</li> </ul> <p>Dies kann leicht geprüft werden (last cs16*, Server-Log-Dateien).          Bitte unterlassen Sie also Täuschungsversuche in Ihrem eigenen Interesse.</p>
Bemerkungen	<p>Hinterlegen Sie alle Programme und Antworten in elektronischer Form! Es wird kein Papier angenommen.          Möchten Sie Lösungen erläutern, so nutzen Sie Quelltext-Kommentare oder legen eine Text-Datei an.          Speichern Sie sämtliche Daten im HOME-Verzeichnis des Nutzers, d. h. unter Windows auf Laufwerk H:\ (NICHT auf C:\ oder "Eigene Dateien")! Idealerweise legen Sie dort ein Unterverzeichnis namens "klausur" an.          Lesen Sie die Aufgaben komplett durch, bevor Sie sie lösen! Die Reihenfolge der Lösung ist Ihnen überlassen.          Probieren Sie immer, eine Aufgabe zu lösen, da auch auf richtige Teile nicht vollständiger Lösungen Punkte vergeben werden!          Falls vom Prinzip her richtig, so werden auch alternative Lösungen akzeptiert.          Sie dürfen beliebig viele Bildschirmausgaben von Werten in den Quelltext einbauen, um ein Programm besser nachvollziehen zu können.          Bitte duplizieren Sie Ihre Quelltextdateien (workspace) NICHT, da beim Korrigieren dann beide durchsucht werden müssen, was sinnlosen Aufwand verursacht. Diese Aufgabenstellung in Papierform können Sie nach dem Ende der Klausur behalten.</p>

<b>Bewertung</b>						
<b>Aufgabe</b>	1	2	3	4	5	Summe
<b>Punkte</b>	10	20	30	20	20	100

Im Verlauf der Prüfung soll ein Adventskalender als Java-Anwendung erstellt werden, der dem Bildschirmschnappschuss entspricht. Die Hintergrundgrafik sowie Bilddateien sind gegeben.



### Aufgabe 1: Swing View [10]

Zweck: Bauen des Hauptfensters.

- a) Erstellen Sie eine Klasse „View“, die ein Swing-Rahmenfenster repräsentiert! Fügen Sie eine Methode namens „initialise“ ein! [2]
- b) Geben Sie dem Fenster darin den Titel „Adventskalender“! Weisen Sie ihm eine Schließoperation zu, welche die Anwendung beendet! [2]
- c) Setzen Sie seine Größe auf 1024x768! Schalten Sie es sichtbar! [2]
- d) Platzieren Sie das Fenster durch lediglich einen Methodenaufruf mittig zum Bildschirm! [2]
- e) Erstellen Sie eine Startklasse namens „Launcher“, in deren „main“-Methode die „View“ instanziert und initialisiert wird! [2]

Ergebnis: Das Hauptfenster der Anwendung wird bei deren Start angezeigt.

### Aufgabe 2: AWT Graphics [20]

Zweck: Einbinden eines Hintergrundbildes.

- a) Erstellen Sie eine Klasse namens „BackgroundPanel“, die von „JPanel“ erbt! Überschreiben Sie die geerbte „paintComponent“-Methode! [2]
- b) Rufen Sie darin zunächst die Super-Methode auf! [2]
- c) Lesen Sie mittels „ImageIO.read“ die gegebene Datei „background.jpeg“ ein! Speichern Sie sie in einer lokalen Variablen! [2]
- d) Zeichnen Sie das Bild unter Verwendung einer Methode des Grafikkontextes! [2]
- e) Fangen Sie geworfene Ausnahmen ab! Schreiben Sie im Fehlerfalle eine Nachricht auf die Fehlerkonsole! [2]
- f) Geben Sie der Klasse eine weitere Methode namens „initialise“! [2]
- g) Erzeugen Sie darin für das „BackgroundPanel“ einen Rand von jeweils 50 Pixel an allen vier Seiten! [2]
- h) Erzeugen Sie einen gitternetzartigen „Layout-Manager“ mit vier Zeilen und sechs Spalten! Der Zellenabstand soll 10 Pixel betragen! [2]
- i) Weisen Sie dem „BackgroundPanel“ den oben erzeugten Rand sowie den „Layout-Manager“ zu! [2]
- j) Geben Sie der Klasse eine Ganzzahl als globales Klassenattribut namens „INDEX“! Initialisieren Sie sie mit dem arithmetischen Wert Null! [2]

Hinweis: Es wird später benötigt, um die Türchen hinzuzufügen.

Ergebnis: Das Hauptfenster wird mit Hintergrundbild angezeigt.



### Aufgabe 3: JavaBean [30]

Zweck: Erstellen einer „JavaBean“, welche ein Türchen repräsentiert.

- a) Erstellen Sie eine „JavaBean“ namens „Door“, die von „JPanel“ erbt! [2]
- b) Geben Sie ihr zwei Attribute: „cover“ vom Typ „ JButton“ und „content“ vom Typ „ JLabel“! [2]
- c) Machen Sie beide Attribute zu Eigenschaften der JavaBean! (Hinweis: Kapselung) [2]
- d) Ein drittes Instanzattribut namens „cardLayout“ ist vom Typ „CardLayout“! Es soll nicht als Eigenschaft nach außen geführt werden! [2]
- e) Geben Sie der Klasse auch zwei Konstanten namens COVER und CONTENT, und initialisieren Sie sie mit den Zeichenketten „cover“ und „content“! [2]
- f) Fügen Sie der Klasse eine „initialise“-Methode hinzu, die eine Ganzzahl entgegennimmt! [2]
- g) Erzeugen Sie darin zunächst die Objekte für die drei Instanzattribute! [2]
- h) Weisen Sie dem „cover“-Knopf eine grüne Hintergrundfarbe zu! [2]
- i) Weisen Sie ihm außerdem eine beliebige Schriftart mit Fettdruck und Größe 64 zu! [2]
- j) Weisen Sie ihm als Text die übergebene Ganzzahl, erhöht um eins, zu! [2]
- k) Weisen Sie dem „content“-Beschriftungsfeld („Label“) horizontal mittige Ausrichtung zu! [2]
- l) Weisen Sie ihm ein „ImageIcon“-Objekt zu, das neu zu erstellen ist! [2]
- m) Bestimmen Sie das zu ladende Bild aus dem Verzeichnis gegebener Dateien anhand des übergebenen ganzzahligen Parameters als Index! [2]

Hinweis: Transformieren der Ganzzahl in eine zweistellige Zeichenkette via Konstruktorauftrag „new DecimalFormat("00")“ und entsprechender Formatierung!

- n) Weisen Sie schließlich dem „Door“-Objekt selbst eine weiße Hintergrundfarbe zu! Weisen Sie ihm auch den oben erstellten „CardLayout“-Manager zu! [2]
- o) Fügen Sie den „cover“-Knopf sowie das „content“-Beschriftungsfeld zum „Door“-Objekt hinzu! (Hinweis: „CardLayout“ berücksichtigen!) [2]

Ergebnis: Die fertige Klasse enthält alle Daten eines Türchens. Keine Veränderung im Fenster.

### Aufgabe 4: Controller [20]

Zweck: Schrittweises Komplettieren des Fensters durch Türchen.

- a) Erstellen Sie eine Klasse namens „Controller“, welche „Action“-Ereignisse verarbeitet! [2]
- b) Übergeben Sie ihr mittels „initialise“-Methode das „View“-Objekt und speichern Sie es in einem Instanzattribut! [2]
- c) Erzeugen Sie ein „Timer“-Objekt, das aller halber Sekunden ein Ereignis auslöst! [2]



- d) Weisen Sie es einem neuen Instanzattribut zu! Starten Sie den „Timer“! [2]
- e) Prüfen Sie in der Ereignisverarbeitungsmethode zunächst, ob die globale Variable „BackgroundPanel.INDEX“ die Zahl von 24 Türchen noch nicht überschritten hat! [2]
- f) Falls die Zahl der Türchen erreicht ist, so stoppen Sie den „Timer“! [2]
- g) Falls die Zahl noch nicht erreicht ist, so erzeugen Sie ein neues „Door“-Objekt! Initialisieren Sie es unter Mitgabe des Indexes! [2]
- h) Fügen Sie es zum „BackgroundPanel“-Objekt der „View“ hinzu! Veranlassen Sie durch Aufruf der Methode „revalidate“ einen Neuaufbau (Neuzeichnen) der Kachel („Panel“)! [2]
- i) Inkrementieren Sie die globale „INDEX“-Variable! [2]
- j) Instanziieren und initialisieren Sie den „Controller“ in der „main“-Methode! [2]

Ergebnis: Das Hauptfenster wird nach Erstellung mit 24 Türchen gefüllt.

### Aufgabe 5: Event Handling [20]

Zweck: Öffnen eines Türchens.

- a) Erweitern Sie die „Door“-„JavaBean“ in der Weise, dass auch hier „Action“-Ereignisse abgefangen werden! [2]
- b) Übergeben Sie in der „initialise“-Methode das „Door“-Objekt selbst als Ereignislauscher an den „cover“-Knopf! [2]
- c) Implementieren Sie die Ereignisverarbeitungsmethode! [2]
- d) Bringen Sie, unabhängig davon, welches „ActionEvent“ ankommt, das „content“-Beschriftungsfeld in den Vordergrund (machen es sichtbar)! (Hinweis: „CardLayout“) [2]
- e) Bestimmen Sie die Aufschrift des „cover“-Knopfes! Speichern Sie die Zeichenkette in einer lokalen Variablen! [2]
- f) Prüfen Sie per Zeichenkettenvergleich, ob es sich um den Nikolaustag handelt! [2]
- g) Öffnen Sie im Erfolgsfall via „JOptionPane“ ein Hinweisfenster (MessageDialog PLAIN)! [2]
- h) Geben Sie dem Hinweisdialog den Fenstertitel „Sankt Nikolaus“ und die Nachricht „Es ist Nikolaustag!“! [2]
- i) Erzeugen Sie ein „ImageIcon“-Objekt für das gegebene Bild „nikolaus.gif“! [2]
- j) Übergeben Sie es als fünftes Argument an die „showMessageDialog“-Methode! [2]

Ergebnis: Nach dem Öffnen eines Türchens wird ein Bild angezeigt, zu Nikolaus mit Dialog.

### Viel Erfolg!

