

Probe-Programmmentwurf Programmieren II

Disclaimer: So würde der „echte“ Programmmentwurf präsentiert



- **Während Prüfung: WLAN & Co. aus!**
- **Letzte Chance**
 - **JavaDoc für OFFLINE-NUTZUNG installiert?**
 - **Vorlesungsfolien auf Rechner geladen?**

<https://www.iai.kit.edu/javav1/>

Probe-Programmmentwurf Programmieren

Heute und jetzt

Institut für Automation und angewandte Informatik

```
final List<String> allResults = new ArrayList<String>();  
final Map<String, Integer> typeWordResultCount = new HashMap<String, Integer>();  
final Map<String, Integer> typePoints = new HashMap<String, Integer>();  
evaluation.put(type, typePoints);  
  
for (final Sheet sheet : this.sheets) {  
    final String sheetResult = sheet.getPlayerInput(type);  
    if (sheetResult.startsWith(start) && this.isValidWord(sheetResult, type)) {  
        validWordCountForType++;  
        allResults.add(sheetResult);  
    }  
}
```

Ablauf

■ Jede/r bekommt Aufgabe

- Auf Papier
- Auf USB-Stick passend zur Matrikelnummer
heute: per Download

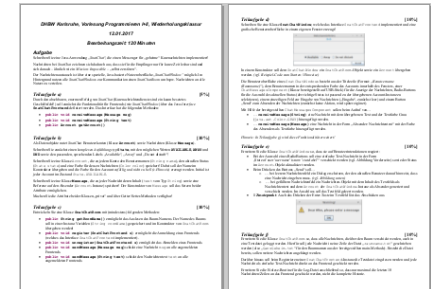
■ Gemeinsame Besprechung der Aufgabe


- Gelegenheit für Verständnisfragen

■ Im Anschluss 120 Minuten Bearbeitungszeit

■ Danach: Abgabe des Quellcodes

- Details: gleich



TINF: 

Matrikelnummer	USB-Stick
1234567	S1
1234568	S2
1234569	S3
1234570	S4
1234571	S5

(später größer)

```
/**
 * Start the exam handler application
 *
 * @param args
 *      CLI arguments
 */
public static void main(final String[] args) throws Exception {
    System.setProperty("java.protocol.handler.pkgs", "org.xhtmlrenderer.protocols");
    UIUtil.enableAlternateTableFlowColoring();

    final ExamHandlerFrame ehf = new ExamHandlerFrame();
    SwingUtilities.invokeLater(() -> {
        ehf.setLocationRelativeTo(null);
        ehf.setVisible(true);
    });

    if (args != null && args.length > 0) {
        final Path toOpen = Paths.get(args[0]);
        ehf.openProject(toOpen);
    }

    if (ExamHandlerSettings.INSTANCE.updateCheckOnStart) {
        ehf.checkForUpdate(false);
    }
}
```

.java

Hinweise & Regeln

- Studierendenausweis bitte sichtbar hinlegen
- Am besten neues Java-Projekt für die Prüfung anlegen

- Erlaubte Hilfsmittel
 - „Stand-Alone“ Laptop mit IDE
 - Vorlesungsfolien + Übungsaufgaben (inkl. Ihrer & Musterlösungen)
 - JavaDoc auf Laptop (Offline!)
 - Bücher zur Java-Programmierung

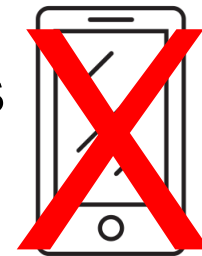
- Nicht erlaubte Hilfsmittel
 - Internet / WLAN und sonstige Verbindungen mit anderen Geräten
 - Abschreiben / Teamarbeit
 - ➔ wer erwischt wird: **Ausschluss von der Prüfung + durchgefallen**

Hinweise & Regeln II

- Auf dem Tisch haben somit alle Dinge außer
 - Studierendenausweis
 - Laptop + Equipment (Netzteil, Maus, ...)
 - Prüfungs-USB-Stick
 - Aufgabenblatt + evtl. Notizblätter / Stift
 - Java-Bücher

nichts zu suchen!

- Das betrifft insbesondere auch Smartphones und sonstige Mobiltelefone! → **Flugmodus!**



- Vorbehalt *(der Vollständigkeit halber – ersparen Sie sich das selbst und uns!)*
 - Wir behalten wir uns vor, dass der komplette Kurs den Programmentwurf **auf Papier** wiederholt

- **Quellcode (*.java-Dateien)** auf USB-Stick kopieren*
 - In das vorhandene Verzeichnis mit der Matrikelnummer
 - Am besten komplettes **Source**-Verzeichnis des Prüfungsprojekts
 - Voraufgespielte Daten auf dem Stick nicht löschen!
- USB-Stick enthält zwecks automatisierter Zuordnung signierte Datei (`$Stick.drivemeta`) mit
 - Kursname, Matrikelnummer und Stick-Nummer
- Abgabe soll anonym sein
 - Falls IDE automatisch Namen o.Ä. in Code schreibt, bitte entfernen

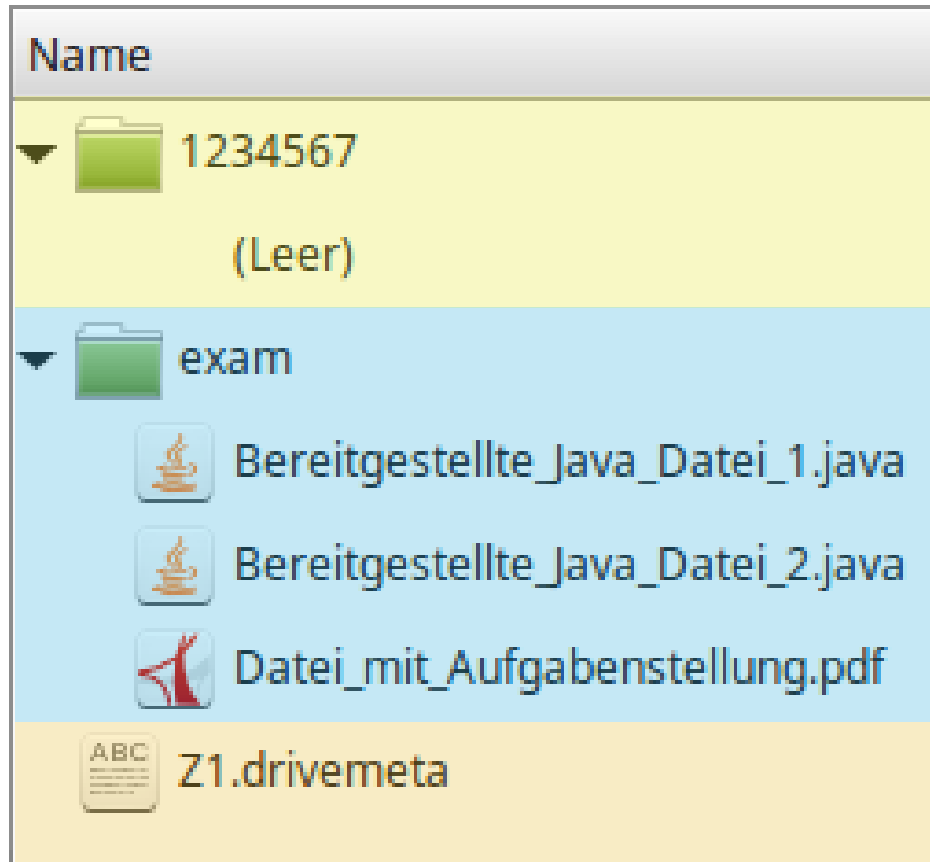


* wem dies aufgrund von Beschränkungen des Firmenlaptops nicht möglich ist, beim Abholen des USB-Sticks den Dozenten darauf hinweisen!

Abgabe II

heute: gar nicht – nur zur Info

■ Vorschau auf USB-Stick-Inhalt



- Matrikelnummer
 - Leeres Verzeichnis für **Ihren Quellcode!**
- Aufgabenstellung
 - Bereitgestellter Quellcode
 - Aufgabenstellung als PDF
- Signierte Datei (Zuordnung)
 - nicht löschen!

- Falls Bibliotheken über Java SE hinaus verwendet wurden müssen die entsprechenden JAR-Dateien mit auf den USB-Stick!
- Kopieren der Dateien muss nicht in den 120 Minuten Bearbeitungszeit erfolgen
- Nach 120 Minuten jedoch **nur noch Kopieren** der Dateien erlaubt!
 - Danach: Laptop zu!
 - Wer bei etwas anderem erwischt wird: Ausschlussgrund!

Abgabe IV

heute: gar nicht – nur zur Info

■ Variante A: USB-Stick

- Mit **bespieltem USB-Stick** und **Studierendenausweis** nach vorne kommen
- Nicht drängeln – alle kommen dran! 😊



■ Variante B: ohne USB-Stick

- WLAN **noch nicht** aktivieren!
- Laptop vorne bei Dozent aufs Pult stellen
- Warten bis alle USB-Stick-Abgaben durch sind
- Danach **unter Aufsicht**:
 - Studierendenausweis mitbringen
 - ZIP des Quellcodes erstellen
 - WLAN an + Schicken per Mail
 - Von Dozent an sich selbst per Web-Mailer → Anonymisierung



Toilettenregelung

- Gleichzeitig darf nur 1 Person auf die Toilette gehen
- Gegen Ende wird angekündigt:
Wer noch auf die Toilette will, soll jetzt gehen.
- Erst wenn niemand mehr auf Toilette gehen will, dürfen Prüfungsteilnehmer den Raum verlassen



Heute: anschließende Besprechung

- Gemeinsames Entwickeln einer Lösung
- Korrekturbogen zur Bewertung
→ welche Note wäre es geworden!?

Korrekturbogen Probe-Programmtextwurf

Kriterium	Mögliche Punkte	Erreichte Punkte
a)		
b)		
c)		
d)		
e)		
Summe	100	

Bewertung (Punkte→Note)									
100	1,0	86	1,5	72	2,4	58	3,3	44	4,2
99	1,0	85	1,6	71	2,5	57	3,4	43	4,2
98	1,0	84	1,6	70	2,5	56	3,4	42	4,3
97	1,0	83	1,7	69	2,6	55	3,5	41	4,4
96	1,0	82	1,8	68	2,7	54	3,5	40	4,4
95	1,0	81	1,8	67	2,7	53	3,6	39	4,5
94	1,0	80	1,9	66	2,8	52	3,7	38	4,6
93	1,1	79	2,0	65	2,8	51	3,7	37	4,6
92	1,1	78	2,0	64	2,9	50	3,8	36	4,7
91	1,2	77	2,1	63	3,0	49	3,9	35	4,7
90	1,3	76	2,2	62	3,0	48	3,9	34	4,8
89	1,3	75	2,2	61	3,1	47	4,0	33	4,9
88	1,4	74	2,3	60	3,2	46	4,0	32	4,9
87	1,5	73	2,3	59	3,2	45	4,1	<41	5,0



NOCH FRAGEN ZUM ABLAUF?



Dann ist jetzt der Zeitpunkt gekommen:

**SÄMTLICHE
NETZVERBINDUNGEN AUS!**

Probe-Programmmentwurf Programmieren I+II



1 – 2 – 3 – vorbei...

So sieht die Tabelle aus, welche Matrikelnummer zu USB-Stick-Nummer zuordnet

A collection of black USB drives with white labels, some labeled C14, C18, C19, C23, C13, C16, and C20, scattered on a light-colored, textured surface.

USB-Stick-Fail-Notfallplan

■ Probe-Programmmentwurf Programmieren

- Download Aufgabenblatt:

https://www.iai.kit.edu/javavl/pe/aufgabe/ppe_jb.pdf

- Benutzer: **java**
- Passwort: **sommer**

heute: Normalfall – bitte alle Runterladen!



jBay 1-2-3-vorbei...



Aufgabe

Schreiben Sie eine Java-Anwendung `jBay`, die ein Auktionssystem implementiert!

Hinweis zur Bewertung:

- 25% der Punkte werden nach Funktionstests Ihrer Lösung vergeben.
- 75% der Punkte werden entsprechend des in den Teilaufgaben angegebenen Schlüssels auf Basis des Quellcodes vergeben.

jBay 1-2-3-vorbei...



Teilaufgabe a)

[30 %]

Entwickeln Sie die Klassen `Bieter`, `Ware`, `Gebot`, `Auktion` und `Auktionshaus`!

Ein `Bieter` hat einen Vornamen und einen Nachnamen. Die Methode `getFullName()` soll Vor- und Nachnamen getrennt durch ein Leerzeichen als `String` zurückgeben.

Eine `Ware` hat einen Titel und eine Beschreibung. Implementieren Sie einen Konstruktor `Ware(String titel, String beschreibung)`!

Ein `Gebot` besteht aus einem (Höchst-)Betrag (`double`), den ein `Bieter` in einer Auktion maximal zu zahlen bereit ist, und der Referenz auf diesen `Bieter`.

Eine `Auktion` hat Referenzen auf eine `Ware` und ein `Gebot` (das derzeitige Höchstgebot, anfangs `null`). Außerdem hat sie Attribute für den aktuellen Preis (`double`, anfangs `0.0`) und das Ende der `Auktion` (Typ `Calendar`, s. *Hinweis zu Calendar am Ende*).

(Hinweis: Der aktuelle Preis ist im Allgemeinen niedriger als der im Höchstgebot enthaltene Höchstbetrag. Er erhöht sich mit jedem Gebot nach den unten stehenden Regeln.)

jBay 1-2-3-vorbei...



Auktion beinhaltet eine statische Konstante `double increment=1.0`. Dabei handelt es sich um den Mindestpreis, den ein erstes Gebot erzielen muss. Auch jedes neue Gebot muss den aktuellen Preis um mindestens dieses `increment` übertreffen.

Auktion hat eine Methode `boolean gebotAbgeben(Gebot g)`, die ein neues Gebot entgegennimmt. Wird dieses Gebot `g` zum Höchstgebot, wird `true` zurückgegeben, ansonsten `false`. Für das neue Gebot `g` gelten folgende Regeln:

- Ist der in `g` gebotene Betrag niedriger als (aktueller Preis+`increment`), wird das Gebot abgelehnt.
- Ist `g` das erste Gebot dieser Auktion, wird der aktuelle Preis auf `increment` festgesetzt.*
- Ist das Gebot `g` vom selben Bieter wie das momentane Höchstgebot, wird nur eine Erhöhung des Gebots akzeptiert. Am aktuellen Preis ändert sich jedoch nichts.
- Ist der in `g` gebotene Betrag zwar höher oder gleich (aktueller Preis+`increment`), jedoch geringer oder gleich dem momentanen Höchstgebot, wird der aktuelle Preis auf das Minimum aus dem in `g` gebotenen Betrag+`increment` und momentanem Höchstgebot festgesetzt.
- Ist der in `g` gebotene Betrag höher oder gleich (aktueller Preis+`increment`) und auch höher als das momentane Höchstgebot, wird der aktuelle Preis auf das Minimum aus dem in `g` gebotenen Betrag und (momentanem („alten“) Höchstgebot+`increment`) festgesetzt.*

*) `g` wird in diesen Fällen neues Höchstgebot.

Implementieren Sie einen Konstruktor `Auktion(Ware ware, int dauer)`, wobei `dauer` die Dauer der Auktion in Minuten bedeutet (s. Hinweis zu `Calendar`).

Teilaufgabe b)

[3%]

Schreiben Sie die Klasse `jBay` mit der folgenden `main`-Methode (*Konstruktor-Aufrufe von `Ware` bzw. `Auktion` wie in Teilaufgabe a); die Zahlen 2, 4 und 5 bedeuten, dass die jeweilige Auktion in 2 (bzw. 4 bzw. 5) Minuten enden soll.:*)

```
public static void main(String[] args) {  
    Auktionshaus jbay = new Auktionshaus();  
    jbay.addAuktion(new Auktion(  
        new Ware("Turnschuhe", "Tolle Turnschuhe, kaum getragen"), 2));  
    jbay.addAuktion(new Auktion(  
        new Ware("iPad", "Nagelneues iPad 3"), 4));  
    jbay.addAuktion(new Auktion(  
        new Ware("Currywurst", "Scharf, ohne Pommes"), 5));  
    // An dieser Stelle wird in Teilaufgabe c) erweitert  
}
```

jBay 1-2-3-vorbei...

Teilaufgabe c)

[30 %]

Entwickeln Sie eine Klasse `BieterTerminal`, die jeweils eine Bieter-Sitzung implementiert!

Sie soll einen Konstruktor `BieterTerminal(Bieter bieter, Auktionshaus ah)` haben, der ein neues Terminal für einen bestimmten `Bieter` und ein bestimmtes `Auktionshaus` erzeugt.

Das `BieterTerminal`-Fenster soll im Titel den Namen des Bieters, oben die aktuelle Uhrzeit (zunächst zum Zeitpunkt des Öffnens des Fensters) sowie eine Liste der im `Auktionshaus` vorhandenen Auktionen (jeweils mit Titel der Ware, aktuellem Preis, momentan höchstbietendem Bieter („---“ falls noch kein Gebot vorliegt), Ende der Auktion und einem Button „Gebot“) anzeigen (s. *Screenshot*).



Beim Klick auf „Gebot“ soll für diese Auktion ein neues Gebot eingelesen werden (z.B. mit `JOptionPane.showInputDialog(...)`). Dabei soll (der aktuelle Preis+`increment`) als Betrag vorgeschlagen werden. Falls die Auktion schon beendet ist (Aktuelle Zeit>Ende der Auktion), soll stattdessen ein entsprechender Hinweis kommen (s.u.). Nach Übergabe des eingelesenen Betrages an die Methode `gebotAbgeben` der Auktion, soll der Nutzer in einer Box mitgeteilt bekommen, ob das Gebot zu gering war (s.u.) oder ob er nun Höchstbietender ist. Ungültige Eingaben sollen abgefangen und wie 0.0 behandelt werden.

jBay 1-2-3-vorbei...



Eingabe

? Bitte neues Gebot eingeben.
Mindestens 5.0 Euro

5.0

OK Abbrechen

Meldung

i Die Auktion ist leider schon vorbei...

OK

Meldung

i Sie sind Höchstbietender!

OK

Meldung

i Gebot zu gering!

OK

jBay 1-2-3-vorbei...



Alle an das Auktionshaus angeschlossenen `BieterTerminals` sollen dann ihre Anzeige entsprechend anpassen, z.B. den neuen Höchstbietenden und den aktualisierten Preis anzeigen.

Erweitern Sie Auktionshaus um Methoden `void register(BieterTerminal bt)` und `void unregister(Bieterterminal bt)`, die ein `BieterTerminal` bei diesem Auktionshaus an- bzw. abmelden.

Implementieren Sie eine Methode `void updateTerminals()`, die den `BieterTerminals` mitteilt, dass sie neu dargestellt werden müssen.

Erweitern Sie die `main`-Methode von `jBay` um die folgenden Zeilen:

```
BieterTerminal b1 = new BieterTerminal(new Bieter("Micky", "Maus"), jbay);  
BieterTerminal b2 = new BieterTerminal(new Bieter("Donald", "Duck"), jbay);
```

jBay 1-2-3-vorbei...



Teilaufgabe d)

[7%]

Loggen Sie alle Gebote (egal ob erfolgreich oder nicht) in einer Textdatei „auktionen.txt“ (im Projekt-Verzeichnis) mit. Neue Logs sollen dabei an die Datei angehängt werden, ohne dass die bisherigen Einträge entfernt werden. Ein Log-Eintrag soll etwa diese Form haben:

```
[Fre Oct 12 10:26:55 CEST 2012] Gebot von Donald Duck für iPad: 1.0 Euro.
```


jBay 1-2-3-vorbei...



Teilaufgabe e)

[5%]

Zeigen Sie in jedem `BieterTerminal`, jeweils gesteuert durch einen Java-Thread, sekundlich die aktuelle Uhrzeit an.

jBay 1-2-3-vorbei...



Hinweis zu „Calendar“:

Mit Hilfe der Klasse `java.util.Calendar` lässt sich die aktuelle Uhrzeit abrufen oder eine beliebige Uhrzeit (jeweils inkl. Datum) darstellen.

Eine Calendar-Instanz mit der aktuellen Uhrzeit erzeugen:

```
java.util.Calendar zeit = java.util.Calendar.getInstance();
```

Darstellung dieser Zeit als String:

```
zeit.getTime().toString(); // z.B. "Fri Oct 12 10:26:42 CEST 2012"
```

Inhalt von zeit auf „in zwei Minuten ab jetzt“ festlegen:

```
int minuten = 2;  
zeit.setTimeInMillis(System.currentTimeMillis() + 60000 * minuten);
```

`(System.currentTimeMillis())` liefert die Anzahl der gemäß Systemuhr seit Mitternacht des 1. Januar 1970 vergangenen Millisekunden als `long`, derzeit etwa 1.349 Billionen ms.)

Vergleich von zeit mit der aktuellen Uhrzeit:

```
if (zeit.after(java.util.Calendar.getInstance())) { ... }
```

Analog dazu gibt es eine Methode `before(...)`

jBay 1-2-3-vorbei...



Weitere Hinweise zum Programm:

- In der grafischen Oberfläche und der Log-Datei muss die Anzeige von Geldbeträgen nicht auf zwei Nachkommastellen normiert werden!
- Zur Vereinfachung: Das Betätigen des Schließen-Buttons in einem der `BieterTerminals` soll jeweils die ganze Anwendung (alle Fenster) schließen.
- Verwenden Sie sprechende (selbsterklärende) Namen für die Variablen etc.
- Die Verwendung von GUI-Buildern ist nicht erlaubt.



1 – 2 – 3 – vorbei...

Die 120 Minuten laufen

VIEL ERFOLG!