

Examination

Web Technologies (WT)

Computer Programming (CP)

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

Personal Data	
First and Last Name	Ihre Daten werden von der Klausuraufsichtsperson schriftlich auf der Anwesenheitsliste festgehalten.
Matriculation Number	
Subject and Year	CS 2012
Login	Das Login "klausur..." 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!

Examination Data	
Date	2012-12-12
Duration [min]	180
Maximum Points [Point]	100
Permitted Study Aids	Dokumentation im lokalen Netzwerkverzeichnis (Intranet); NICHT gestattet sind Kommunikationsmöglichkeiten (Internet) oder Anmeldung via SSH auf dem Rechner "fileserv", wo Ihr Homeverzeichnis liegt, oder eine Anmeldung mit Ihrem Klausur-Login nach Ende der Prüfung. Dies kann leicht geprüft werden (last cs12*, Server-Log-Dateien). Bitte unterlassen Sie also Täuschungsversuche in Ihrem eigenen Interesse.
Remarks	<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.</p> <p>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.</p> <p>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.</p> <p>Sie dürfen beliebig viele Bildschirmausgaben von Werten in den Quelltext einbauen, um ein Programm besser nachvollziehen zu können.</p> <p>Diese Aufgabenstellung in Papierform können Sie nach dem Ende der Klausur behalten.</p>

Evaluation											
Task	1	2	3	4	5	6	7	8	9	10	Summe
Points	5	10	10	10	5	10	20	10	10	10	100

Im Rahmen des ersten Teiles der Klausur soll ein dynamischer Webauftritt zum Thema *Fließendes Geld* erstellt werden. Sein Aussehen soll dem folgenden ähneln:



Task 1: Internet Terminology [5]

Erläutern Sie die folgenden Begriffe und nennen Sie die Langform für Abkürzungen!

- a) Internet [1]
- b) W3C [1]
- c) Nennen Sie beispielhaft vier Dienste des Internets! [1]
- d) MIME [1]
- e) Cookie [1]

Task 2: Hypertext Markup Language (HTML) [10]

Ziel dieser Aufgabe ist die Erstellung und Strukturierung einer Web-Seite.

- a) Erstellen Sie eine Datei namens *index.html* mit Grundgerüst gemäß HTML-Format! [1]
- b) Verwenden Sie als Zeichenkodierung Ihrer Web-Seite *utf-8*! [1]
- c) Geben Sie ihr die Überschrift zweiter Kategorie *Das Wunder von Wörgl*! Maskieren Sie darin den Umlaut ö durch seine *Zeichen-Entität-Referenz*! [1]
- d) Binden Sie die gegebene Bilddatei *arbeitsbestaetigungsschein.png* ein! Verwenden Sie dazu ein *Empty Tag*! Setzen Sie die Bildbreite auf *480 Picture Elements*! Richten Sie das Bild *rechtsbündig* aus! [1]
- e) Fügen Sie den in der Datei *text.txt* gegebenen Text per *Copy&Paste* in die Web-Seite ein! Trennen Sie Abschnitte durch Verwendung des *<p>* Tags! Kennzeichnen Sie Zwischenüberschriften durch ein passendes Tag als *sechstkategorisch*! [1]
- f) Markieren Sie alle Zwischenüberschriften durch einen *Anker* mit eindeutigem Namen! [1]
- g) Fügen Sie ein kleines Inhaltsverzeichnis der fünf Zwischenüberschriften als *Aufzählungsliste* unterhalb der Hauptüberschrift der Seite und oberhalb des Textes ein! [1]
- h) Definieren Sie für jeden Eintrag des Inhaltsverzeichnisses einen *Verweis* auf den Anker der entsprechenden Zwischenüberschrift! [1]
- i) Geben Sie der Web-Seite einen *Titel* als *Meta-Information* für z. B. *Browser-Historie* oder *Suchmaschinen*! [1]
- j) Binden Sie zur Formatierung eine *CSS-Datei* namens *formate.css* in die Web-Seite ein! [1]
- k) Zusatzaufgabe: Geben Sie als Dokumenttyp *HTML 5* an! [1]

Task 3: Cascading Style Sheets (CSS) [10]

Ziel dieser Aufgabe ist die Formatierung der Web-Seite.

- a) Begrenzen Sie die Seitenbreite auf *800 px* und lassen Sie links einen Rand von *100 px*! [2]
- b) Entfernen Sie jegliche Formatierung für innerhalb der Aufzählungsliste zu findende *Listeneinträge*! [2]
- c) Formatieren Sie Verweise so, dass sie *nicht unterstrichen* sind und in der Farbe mit dem Hexadezimalwert *#996600* dargestellt werden! Selbige soll im Falle des Darüber-Hinwegfahrens mit der Maus zur Hintergrundfarbe werden, ergänzt durch eine weiße Schriftfarbe. [2]
- d) Definieren Sie eine Klasse namens *frame* mit einem *durchgezogenen* Rand der Breite *1 px* und der Farbe *#996600* sowie einem Innenabstand von *10 px*! [2]
- e) Ordnen Sie diese Klasse im HTML-Dokument der Hauptüberschrift zu! [2]

Task 4: Java Script (JS) and Document Object Model (DOM) [10]

Ziel dieser Aufgabe ist die Dynamisierung der Web-Seite.

- a) Fügen Sie der HTML-Datei ober- oder unterhalb des Bildes ein neues *<p>* Element mit dem Attribut *id="hint"* hinzu! [2]
- b) Binden Sie eine *JavaScript*-Datei namens *logic.js* in die Web-Seite ein! [2]
- c) Erstellen Sie in der Datei *logic.js* zwei Funktionen namens *showHint()* und *hideHint()*! [2]
- d) Ergänzen Sie die beiden Funktionen so, dass sie sich auf das Element mit der Identifikation *hint* beziehen! Dabei soll *showHint()* ihm die folgende Zeichenkette zuordnen: 'Notgeld ersetzt fehlende gesetzliche Zahlungsmittel und wird von Staaten, Gemeinden oder privaten Unternehmen herausgegeben.' In *hideHint()* hingegen wird ein leerer String zugewiesen. [2]
- e) Sorgen Sie dafür, dass bei einem Klick mit der Maus auf das Bild die *JavaScript*-Funktion *showHint()* aufgerufen wird! Beim Verlassen der Bildfläche durch die Maus soll die Funktion *hideHint()* aufgerufen werden. [2]

Task 5: Microformat (µF) [5]

Ziel dieser Aufgabe ist die Anreicherung der Website durch Metadaten.

Geben Sie am Ende der Web-Seite *index.html* die Lage der österreichischen Stadt Wörgl durch einen kurzen Textabschnitt mit folgenden Daten an: Breite: 47° 29' N; Länge: 12° 4' O! Verwenden Sie das Geo Microformat, um diese für Menschen sichtbaren Informationen auch für Agenten auffindbar zu machen!



Der zweite Teil der Klausur untersucht anhand kleiner, voneinander unabhängiger Java-Programme die Problematik des Zinseszinses. Dazu zwei Begriffsdefinitionen:

Zinssatz: dezimaler Wert wie z. B. 0,05

Zinsfuß: Zahl vor Prozentzeichen wie z. B. 5 in 5 % (entspricht Zinssatz 0,05)

Task 6: Operator and Wrapper Class [10]

Im Goldenen Mittelalter zwischen ca. 1050 und 1450 diente Geld als reines *Tauschmittel*. Die in Mitteleuropa genutzten *Brakteaten* wurden jährlich verrufen, wodurch jedermann bestrebt war, sie möglichst schnell wieder für Realwerte auszugeben. Dieses *Fließende Geld* führte zu einer wirtschaftlichen Blütezeit. Erst in der Zeit der Fugger im 15. Jh. erlangte der so genannte *Dickpfennig* an Bedeutung. Er wurde nicht verrufen und diente zunehmend als *Schatzmittel*. Das dem Wirtschaftskreislauf dadurch entzogene Geld rief seither regelmäßig Krisen hervor. *Zinsen* wurden eingeführt, um das gehortete Geld in den Wirtschaftskreislauf zurück zu locken. Ziel dieser Aufgabe ist die Verwendung einer einfachen Formel der Zinsmathematik.

a) Verwenden Sie lokale *Gleitkommavariablen*, um zwei via Kommandozeile übergebene *Argumente* zu speichern! Das erste Argument soll für ein *Guthaben* und das zweite für den gewünschten *Zinsfuß* stehen. [2]

b) Wandeln Sie die in Empfang genommenen Argumente unter Verwendung einer passenden Methode einer *Wrapper*-Klasse vom Typ *String* in den Typ *double* um, damit sie den lokalen Variablen zugewiesen werden können! [2]

c) Dividieren Sie den *Zinsfuß* durch den Wert *100* und speichern Sie den resultierenden *Zinssatz* in einer weiteren lokalen Variablen! [2]

d) Bestimmen Sie die *Zinsen* durch Multiplikation des Kapitals mit dem Zinssatz und speichern Sie das Ergebnis in einer weiteren lokalen Variablen! [2]

e) Addieren Sie die berechneten Zinsen unter Verwendung eines *zusammengesetzten Zuweisungsoperators* (compound assignment) zum eingesetzten Kapital hinzu und geben das Ergebnis auf der *Konsole* aus! [2]

Task 7: Structured Programming and Array [20]

Im eine *Natürliche Marktwirtschaft* behindernden Kapitalismus heutiger Prägung vermehrt sich Kapital leistungslos, auf Basis des *Zinseszinses*, und führt somit zur Umverteilung des Vermögens von Fleißig nach Reich. Ziel dieser Aufgabe ist die Untersuchung eines durch Zinseszinsen aus sich selbst heraus vermehrenden Guthabens.

a) Verwenden Sie eine *for*-Schleife, um das Wachstum eines Guthabens über die Jahre zu berechnen! [5] Gegeben seien dazu folgende Werte:

Startkapital: 0,01 EUR im Jahre 0

Zinsfuß: 5 %

Laufzeit: bis ins heutige Jahr 2012

Zinseszinsformel: $\text{kapital} += \text{kapital} * \text{zinsfuß} / 100.0;$

b) Speichern Sie die berechneten Werte in einem Feld (*Array*) passender Größe! [5]

c) Iterieren Sie über das Feld (*Array*) und geben jeweils die Jahreszahl aus, wenn folgende Werte erreicht sind [5]:

1 EUR
100 EUR
1000 EUR
1 kg Gold
1 t Gold
1 Erdkugel Gold
1 Mio. Erdkugeln Gold
1 Mrd. Erdkugeln Gold

Gegeben seien dazu folgende Daten:

Goldpreis: 42361.13 EUR für 1 kg (Barren)
Erdmasse: 5.974e24 kg

d) Geben Sie *nur* die in Teilaufgabe c) geforderten Jahreszahlen auf der Konsole aus, andere nicht! [5]

Hinweis: Eine Möglichkeit besteht darin, für jede Stufe ein *Flag* zu definieren und bei Erreichen der Stufe zu setzen. Die Bedingungen der Vergleiche werden dann mit dem Flag verknüpft.

e) Zusatzaufgabe: Geben Sie am Ende den *gerundeten* Wert des Guthabens im Jahre 2012 aus! Verwenden Sie zur Rundung eine statische Methode der *Math*-Klasse! [2]

Task 8: Procedural Programming [10]

Alle großen Weltreligionen verbieten den *Wucher* (Geldverleih gegen Zins). Es gibt jedoch einen gravierenden mathematischen Unterschied im Kurvenverlauf: mit Zinseszins *exponentiell* und mit einfachem Zins *linear*. Ziel dieser Aufgabe ist der Aufruf zweier Methoden.

a) Erstellen Sie eine Startklasse, die neben der *main*-Methode zwei weitere Methoden namens *berechneGuthabenZins* und *berechneGuthabenZinseszins* enthält! Beide sollen je drei Parameter entgegennehmen: ein Startkapital *s* des Types *double*, den Zinsfuß *z* des Types *double* und die Laufzeit *l* des Types *int*. [2]

b) Initialisieren Sie in *beiden* Berechnungsmethoden eine lokale Variable namens *kapital* mit dem Wert des Startkapitals und geben Sie sie am Ende der Methode als *Rückgabewert* an! [2]

c) Verwenden Sie dazwischen eine *for*-Schleife zur Berechnung, nach folgenden Formeln [2]:

Zinsberechnung (Zins jährlich abgeholt): $\text{kapital} += \text{startguthaben} * \text{zins} / 100.0;$

Zinseszinsberechnung (Zins belassen und selbst mitverzinst): $\text{kapital} += \text{kapital} * \text{zins} / 100.0;$

d) Rufen Sie beide Methoden in der *main*-Methode auf und geben Sie folgende Werte mit [2]:

`startkapital = 0.01; zinsfusz = 5.0; laufzeit = 2012;`

e) Speichern Sie die Rückgabewerte in lokalen Variablen und geben Sie sie auf der Konsole aus! Welches Guthaben erreicht man mit *einfachem Zins* (linearem Verlauf) gegenüber dem *Zinseszins* (exponentiellem Verlauf)? Notieren Sie die (dramatisch abweichenden) Euro-Beträge als Kommentar im Quelltext! [2]

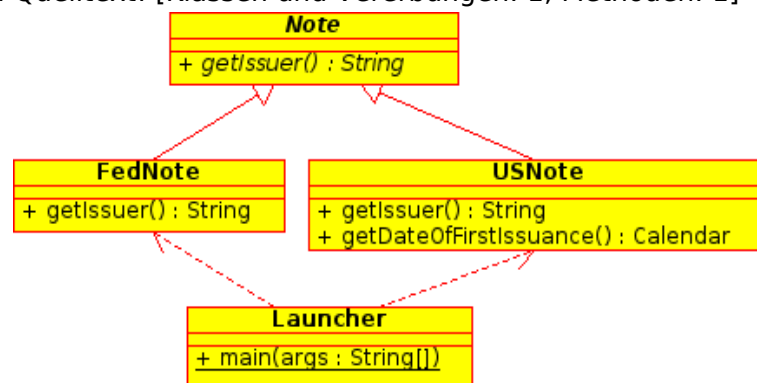
Task 9: Object Oriented Programming (OOP) and Polymorphism [10]

Eigentlich müsste ein Staat überhaupt nicht verschuldet sein und ständig Zinsen zahlen, denn er könnte ja eigenes Geld herausgeben. Geld entsteht jedoch in den *Zentral- und Geschäftsbanken*, bei deren Eigentümern alle Fäden zusammenlaufen. Heute basiert das globale Finanzsystem auf dem *US Dollar* als Weltreservewährung. Durch den *Federal Reserve Act* von 1913 hat eine Gruppe von Privatpersonen – die Eigentümer der größten privaten Banken – diese Währung unter ihre Kontrolle gebracht. Der letzte US-Präsident, der die *Fed* entmachten wollte, hieß John Fitzgerald Kennedy. Kaum bekannt ist, dass er am 4. Juni 1963 eine Verordnung *Executive Order No. 11110* unterschrieben hat, die es der US-Regierung ermöglichte, eigenes, silbergedecktes Geld – die *US Note* (siehe Abbildung) – herauszugeben, anstatt Zinsen für Fed Notes zu bezahlen. 1963 kursierten vorübergehend Dollar-Noten von zwei verschiedenen Emittenten (Quelle: <http://www.wissensmanufaktur.net/>):



Kennedys Nachfolger, Lyndon B. Johnson, hat in seiner ersten Amtshandlung *Verordnung No. 11110* ausgesetzt. Das Staatsgeld wurde umgehend aus dem Verkehr gezogen. Ziel dieser Aufgabe ist die Implementierung zweier Banknoten-Klassen.

a) Implementieren Sie die im folgenden UML-Klassendiagramm dargestellte Architektur einer Anwendung in Java-Quelltext! [Klassen und Vererbungen: 1; Methoden: 1]



b) Weisen Sie folgende Rückgabewerte zu [2]:

`FedNote.getIssuer()` --> "Federal Reserve System (Fed). Zentralbanksystem der Vereinigten Staaten. Seine Eigentümer sind durch ihre Einlagen private Banken."

`USNote.getIssuer()` --> "Finanzministerium der Vereinigten Staaten. Eigentümer waren die United States of America (USA)."

`USNote.getDateOfFirstIssuance()` --> `new GregorianCalendar(1963, 06, 04)`

c) Erstellen Sie in der *main*-Methode ein Feld (*Array*) des Types *Note* und weisen Sie ihm zwei Objekte des Types *FedNote* sowie eines des Types *USNote* zu [2]!

d) Verwenden Sie eine *while*-Schleife, um durch das Feld (*Array*) zu iterieren und bringen Sie die *Polymorphie* zur Anwendung, indem Sie an allen Objekten die *getIssuer()*-Methode aufrufen! [2]

e) Nutzen Sie innerhalb der Schleife den *instanceof*-Operator, um herauszufinden, ob es sich bei einem Objekt um eine *USNote* handelt und rufen Sie, wenn dem so ist, seine Methode *getDateOfFirstIssuance()* auf, um das erste Ausgabejahr auf der Konsole auszugeben! [2]

Hinweise: Ein so genannter *Downcast* ist nötig. Man erhält die Jahreszahl, indem am zurückgelieferten *Calendar*-Objekt die *get()*-Methode aufgerufen wird, wie folgt:

`c.get(Calendar.YEAR)`

Task 10: Command Line Interface (CLI) and Exception Handling [10]

Es gibt eine Vielfalt an Lösungen zur Behebung des die aktuelle Finanzkrise verursachenden Fehlers im System. Dazu gehört beispielsweise eine *Komplementärwährung* für Griechenland, die ergänzend und stabilisierend auf den Euro wirken würde. Geld ist eine *Infrastruktur* und sollte mit einer *Nutzungsgebühr statt Zinsen* versehen werden. Ziel dieser Aufgabe ist das Abfangen von Fehlern in einer Java-Anwendung.

a) Der Datenstrom *System.out* dient der Standardausgabe. Nutzen Sie den Datenstrom der Standardeingabe *System.in*, um über die Konsole ein Zeichen einzulesen! [2]

b) Speichern Sie das Zeichen in einer *lokalen Variablen*! [2]

c) Geben Sie es zum Einen als *Zeichen* und zum Anderen als *Zeichencode* aus! [2]

d) Fangen Sie geworfene Ausnahmen in der *main*-Methode ab! [2]

e) Wie könnte man das durch den Compiler geforderte Abfangen der Ausnahme *umgehen*? (Notieren Sie die entsprechende Lösung oder Beschreibung als Kommentar im Quelltext.) [2]

Viel Erfolg!

Quellen

- <http://www.woistunsergeld.de/>
- <http://www.humane-wirtschaft.de/>
- <http://www.cgw.de/>
- <http://www.monetative.de/>
- <http://www.wissensmanufaktur.net/>
- <http://www.wissensmanufaktur.net/steuerboykott/>
- <http://www.compact-magazin.com/>
- <http://www.regiogeld.de/>
- <http://www.macht-geld-sinn.de/>
- <http://www.lust-auf-neues-geld.de/>
- <http://www.geldmitsystem.org/>
- <http://www.zeit.de/2010/52/Woergl?page=all>