

Examination

Data Processing (DP)

Prof. 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 "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!

Examination Data	
Date	2013-06-19
Duration [min]	120
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	20	10	20	30	20	0	0	0	0	0	100

Verschiedene Themen der Java-Programmierung werden innerhalb fünfer Aufgaben geprüft. Sie sind so sortiert, wie sie im Kurs behandelt wurden. Subjektiv nach dem Schwierigkeitsgrad sortiert, ergibt sich folgende, von mir empfohlene Bearbeitungsreihenfolge: 3, 2, 4, 5, 1.

Task 1: Template and Wildcard [20]

Im Folgenden wird der Umgang mit der Generizität von Klassen geprüft.

- a) Definieren Sie eine generische Klasse namens "Paar<T,U>"! [2]
- b) Deklarieren Sie für spätere Zwecke die beiden Instanzvariablen als protected! [2]
- c) Lassen Sie Objekte dieser Klasse via Konstruktor Werte zweier Typen aufnehmen! [2]
- d) Stellen Sie zum Abfragen der Werte passende "get"-Methoden bereit! [2]
- e) Definieren Sie die generische Klasse "ZahlenPaar", die von der Klasse "Paar<T,U>" abgeleitet ist! [2]



f) Passen Sie die Signatur der Klasse "ZahlenPaar" so an, dass ihre Typargumente zur Klasse "Number" kompatibel sind! [2]

g) Geben Sie der Klasse "ZahlenPaar" eine Methode, welche die Summe der beiden Zahlen des Paares als "double"-Zahl zurückgibt! Hinweis: Die abstrakte Klasse "Number" enthält u. a. die abstrakte Methode "double doubleValue()", die z. B. von den Klassen "Integer", "Double" und "BigInteger" implementiert wird. [2]

h) Erstellen Sie eine Startklasse beliebigen Namens mit "main"-Methode! Implementieren Sie in dieser Klasse außerdem die Methode mit der Signatur:

```
public static void print(Paar<?,?> p)
```

welche die beiden Werte a und b eines Paares in der Form: (a, b) ausgibt! [2]

i) Erzeugen Sie in der "main"-Methode zwei Zahlenpaar-Objekte z1 und z2 wie folgt [2]:

ZahlenPaar<Integer, Double> initialisiert mit (12, 1.23)

ZahlenPaar<BigInteger, Double> initialisiert mit (new BigInteger("99999"), 1.23)

j) Rufen Sie die "print"-Methode einmal mit z1 als Parameter und ein andermal mit z2 auf! Geben Sie anschließend die Summen der Zahlenwerte durch Verwendung der "sum"-Methode des jeweiligen Zahlenpaar-Objektes aus! [2]

Task 2: Character Stream [10]

Ziel dieser Aufgabe ist die Bereinigung einer vorgegebenen HTML-Datei von enthaltenen, nicht standardkonformen Zeichen.

a) Definieren Sie zwei Dateireferenzen auf "input.html" und "output.html"! Hinweis: Berücksichtigen Sie eventuell vorhandene Pakete als relativen Pfad in der Dateireferenz! [1]

b) Erzeugen Sie zwei Objekte ungepufferter Zeichenstreams für die Ein- und Ausgabe! [1]

c) Behandeln Sie geworfene Ausnahmen durch Ausgabe eines Fehlertextes via Konsole! [1]

d) Erzeugen Sie zwei Objekte der Klassen "BufferedReader" und "PrintWriter"! [1]

e) Lesen Sie den Dateinhalt mittels Schleife zeilenweise ein! [2]

f) Wandeln Sie Umlaute und Sonderzeichen entsprechend der folgenden Tabelle um! Hinweis: Eine Möglichkeit ist die Verwendung der "String.replaceAll"-Methode. [2]

Zeichen	Ersetzung
ä	ä
Ä	Ä
ö	ö
Ö	Ö
ü	ü
Ü	Ü
ß	ß
"	"

g) Schreiben Sie die so manipulierte Zeile in die Ausgabedatei! [1]

h) Sorgen Sie dafür, dass "close"-Methoden der Stream-Objekte auch dann ausgeführt werden, wenn während der Verarbeitung eine Ausnahme auftrat! [1]

Hinweis: Bei Problemen mit der Zeichendarstellung in der Eclipse-Konsole kann es helfen, für Eclipse die "UTF-8"-Kodierung einzustellen, unter:

| Window | Preferences | General | Workspace | Text file encoding | Other: UTF-8

Task 3: Output Formatting [20]

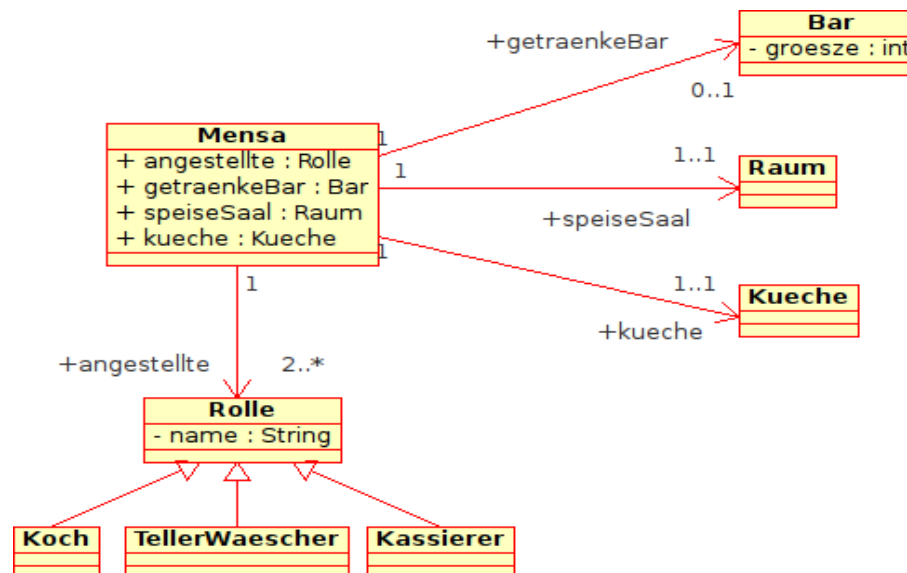
Das Ziel dieser Aufgabe ist die formatierte Ausgabe mittels der Methode "printf".

- Geben Sie mit Hilfe der "printf"-Methode den Vor- und den Nachnamen einer Person über die Konsole aus! Übergeben Sie die beiden Namen als Argumente an das Programm! Stellen Sie den Nachnamen mit einer geeigneten Option der "printf"-Methode in Großbuchstaben dar! [6]
- Fügen Sie an die ausgegebenen Namen durch eine weitere Konsolenausgabe zwei leere Zeilen an! Nutzen Sie dazu einen geeigneten Format-Spezifizierer! [2]
- Lassen Sie mit Hilfe der "printf"-Methode sowie einer "for"-Schleife die Zahlen von Null bis 15 einmal in dezimaler- und ein weiteres Mal in hexadezimaler Schreibweise ausgeben, letztere in Großbuchstaben! [4]
- Erstellen Sie eine Klassenmethode namens "printName", welche als Parameter die Strings "vorname", "nachname", "stadt" sowie den "double"-Wert "alter" erwartet! [2]
- Geben Sie darin mittels "printf"-Methode den Vornamen und den Nachnamen aus, wobei beide mindestens fünf Stellen haben und linksbündig ausgerichtet sein sollen! [2]
- Geben Sie außerdem die ersten drei Buchstaben der Stadt in Großbuchstaben und rechtsbündig ausgerichtet aus, wobei mindestens fünf Stellen verwendet werden sollen! [2]
- Geben Sie in einer neuen Zeile das via Parameter übergebene Alter mit drei Stellen vor und zwei Stellen nach dem Komma aus! Füllen Sie führende Stellen mit der Ziffer Null auf! [2]

Task 4: Cloning [30]

Das Verständnis des Klonens von Objekten (flache/tiefe Kopie) soll geprüft werden.

- Erstellen Sie ein Java-Projekt namens "cloning", welches das im folgenden UML-Klassendiagramm gegebene Domänen-Modell (Klassen, Methoden, Attribute, Vererbungen, Assoziationen) implementiert! [8]



- Erstellen Sie außerdem eine "Launcher"-Klasse mit "main"-Methode und instanziiieren Sie in letzterer das Domänen-Modell -- mit anderen Worten die "Mensa"-Klasse! [2]
- Weisen Sie dem Objekt folgende Werte zu [4]:
 - ein Objekt vom Typ "Kueche"
 - ein Objekt vom Typ "Raum"
 - kein Objekt vom Typ "Bar"
 - ein Objekt vom Typ "ArrayList<Rolle>"

d) Fügen Sie dem Container je ein Objekt folgender Typen hinzu [4]:

- Koch
- TellerWaescher

e) Erzeugen Sie nun ein zweites Objekt des Domänen-Modells vom Typ "Mensa" und zwar als flache Kopie (flat copy/ shallow copy) des bereits vorhandenen Modells! [2]

f) Erweitern Sie die "clone"-Methode der "Mensa"-Klasse so, dass eine tiefe Kopie (deep copy) des vorhandenen Objektes entsteht und richten Sie besonderes Augenmerk auf das Klonen des Containers! [4]

g) Ergänzen Sie noch fehlende Implementierungen der "clone"-Methode in anderen Klassen und implementieren Sie das "Cloneable"-Interface, wo nötig! [4]

Hinweis: Die drei Unter-Klassen der "Rolle"-Klasse besitzen der Einfachheit halber weder Attribute noch Methoden und benötigen daher keine Implementierung der "clone"-Methode.

h) Geben Sie alle Eigenschaften des originalen sowie des geklonten Modells auf dem Bildschirm aus! [2]

Task 5: Document Object Model (DOM) [20]

Ziel dieser Aufgabe ist die Verarbeitung einer XML-Datei mit Java-Mitteln.

a) Erstellen Sie ein Java-Projekt namens "dom", welches eine Hauptklasse mit "main"-Methode enthält! Binden Sie die gegebene "dom4j"-Bibliothek (*.jar Datei) ein! [2]

b) Verwenden Sie die Funktionalität der "dom4j"-Bibliothek, um die gegebene XML-Datei namens "addressbook.xml" zu parsen (Reader, Document, Root, Iterator, Element)! [4]

c) Schachteln Sie zwei Schleifen, um die Elemente der zwei Ebenen unterhalb des Wurzel-Knotens mittels Iteratoren zu verarbeiten! [4]

d) Geben Sie Name und Attribut eines jeden "address"-Knotens auf der Konsole aus! [4]

e) Geben Sie den Namen wie auch Inhalt eines jeden Elementes unterhalb des "address"-Knotens auf der Konsole aus! [4]

f) Führen Sie zwei Zählvariablen ein - je eine für die Anzahl der Adressen insgesamt und die Anzahl der Adressen, die zum "Märchenland" gehören! Geben Sie die beiden Werte auf der Konsole aus! [2]

Task 6: Regular Expressions [5] - Zusatzaufgabe

Gegeben sei folgende, E-Mail-Adressen enthaltende Zeichenkette, deren Adressen mit unterschiedlichen Trennsymbolen wie z. B.: \n, \t versehen sind:

"bill@gates.com\nkonrad@zuse.de\talbert@einstein.de\ntestAfehler.de\t@Testtext.com\nohne@punktnet\tmitZahl@12.de mit12@zahl.net nicht@unterstuetz.net"

a) Schreiben Sie ein Programm, das die Zeichenkette mit Hilfe des regulären Ausdruckses "\\s" in ihre Bestandteile zerlegt und in einem Array speichert! Hinweis: Der Ausdruck "\\s" repräsentiert die Kategorie "white space". [2]

b) Durchlaufen Sie das entstandene Array und suchen Sie nach korrekten deutschen E-Mail Adressen! Verwenden Sie dazu den folgenden regulären Ausdruck (Regex) [3]:

"([\\p{Alpha}]+)@([\\p{Alpha}]+)\\.de"

Hinweis: Der Ausdruck zum Prüfen von E-Mail Adressen ist nur für diese Aufgabe konzipiert, da er stark vereinfacht ist. Echte Prüfer von E-Mail-Adressen sind wesentlich komplizierter!

Viel Erfolg!