

Examination

Computer Programming (CP)

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 2015
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	2016-03-23
Duration [min]	108 von insgesamt 180
Maximum Points [Point]	60
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	10	10	10	20	10	0	0	0	0	0	60

Ihre Programmierkenntnisse werden anhand kleiner, voneinander unabhängiger Java-Programme geprüft.

Task 1: Variable Usage [10]

Ziel dieser Aufgabe ist die Verwendung verschiedener Variablen.

- Erstellen Sie eine Klasse namens "Person" mit einer Klassenvariablen "anzahl" und einer Instanzvariablen "alter", beide vom Typ "int"! [2]
- Erstellen Sie einen Konstruktor, der als Argument eine Parametervariable des Types "int" entgegennimmt! [2]
- Inkrementieren Sie darin die Klassenvariable! Initialisieren Sie außerdem die Instanzvariable mittels des übergebenen Parameterwertes! [2]

d) Erstellen Sie eine "main"-Methode, welche zwei Argumente entgegennimmt und in je einer lokalen Variablen des Types "int" auf dem "Stack" speichert! [2]

Hinweis: Eine Konvertierung ist nötig.

e) Erzeugen Sie zwei Objekte der Klasse "Person" im "Heap"-Speicher, wobei das erste beziehungsweise zweite Kommandozeilenargument als Alter an den Konstruktor mitgegeben werden soll! Geben Sie schließlich das Alter der beiden Personen sowie die Anzahl an Personen auf der Konsole aus! [2]

Task 2: String [10+2]

Ziel dieser Aufgabe ist die Kodierung eines Textes.

a) Erstellen Sie ein Programm mit einer lokalen Variablen des Types "String", welcher standardmäßig folgender Text als Wert zugeordnet wird [1]:

```
"Drei Chinesen mit dem Kontrabass\n" +  
"saßen auf der Straße und erzählten sich was.\n" +  
"Kam die Polizei: 'Ei was ist denn das!'\n" +  
"Drei Chinesen mit dem Kontrabass.\n"
```

b) Falls ein Kommandozeilenargument mitgegeben wurde, so weisen Sie dessen Wert der lokalen Variablen zu, so dass der Standardtext aus Aufgabe a) ersetzt wird! [1]

c) Konvertieren Sie den Text in Kleinbuchstaben! [1]

d) Erstellen Sie eine statische Methode namens "replace", welche eine Zeichenkette als Parameter entgegennimmt und am Ende nach vorgenommenen Änderungen zurückgibt! Ersetzen Sie darin deutsche Umlaute sowie das "Scharfe S" folgendermaßen:

ä durch ae; ö durch oe; ü durch ue; ß durch sz

Rufen Sie die Methode in passender Weise aus der "main"-Methode heraus auf! [2]

e) Schreiben Sie mit gleicher Signatur eine weitere statische Methode namens "encode"! Definieren Sie darin folgende Zeichenkette mit den Buchstaben des Alphabetes:

```
String a = "abcdefghijklmnopqrstuvwxyz";
```

Erstellen Sie unter Verwendung des "new"-Operators außerdem eine neue lokale Zeichenkette zur Speicherung des Ergebnistextes! [1]

f) Iterieren Sie mittels "for"-Schleife durch die als Parameter gegebene Zeichenkette! [1]

g) Bauen Sie den Ergebnistext so zusammen, dass jeder Buchstabe durch seine Positionsnummer im oben erstellten Alphabet-Feld ersetzt wird! Fügen Sie der besseren Lesbarkeit halber hinter jede Positionsnummer ein Leerzeichen ein! [1]

h) Ignorieren Sie alle Zeichen, die keine Buchstaben sind! [2]

i) Zusatzaufgabe: Erkennen Sie Zeilenumbrüche und fügen Sie sie in den Ergebnistext ein! [2]

Task 3: Random Values [10]

Ziel dieser Aufgabe ist die Erzeugung von Zufallszahlen.

a) Erstellen Sie ein Programm, in dem ein Objekt des Types "java.util.Random" erzeugt und für es als Startbedingung (Random Seed) die Ziffer 1 angegeben wird! [2]

b) Nutzen Sie eine Schleife, um mittels der "Random"-Klasse "nextInt"-Methode zehn Zufallszahlen auf der Konsole auszugeben! [2]

c) Setzen Sie die Startbedingung des Objektes über eine geeignete Methode erneut, wiederum auf den Wert 1! Geben Sie die erzeugten Werte per Schleife auf der Konsole aus! [2]

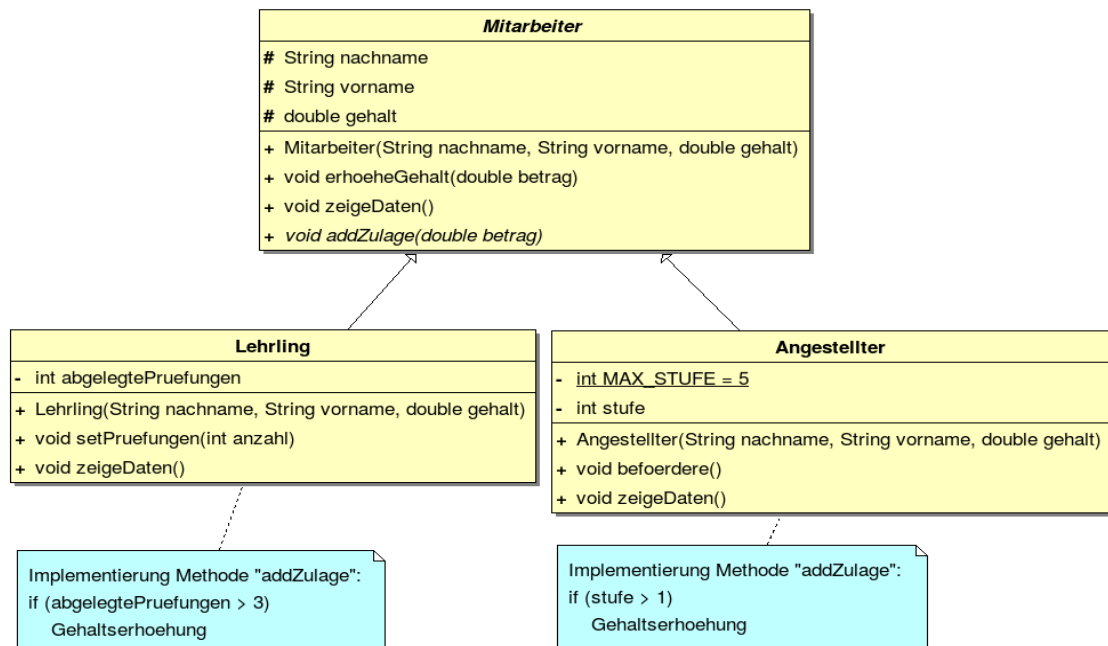
Hinweis: Wenn alles stimmt, so sollten die gleichen Werte wie in der ersten Schleife dastehen.

d) Erzeugen Sie eine weitere Instanz der Klasse "Random", dieses Mal jedoch ohne Startbedingung! Erzeugen Sie wiederum zehn Zufallswerte! Speichern Sie diese in einem passenden Feld (Array)! [2]

e) Geben Sie die im Feld gespeicherten Werte mittels einer weiteren (bereits der vierten!) Schleife auf der Konsole aus! [2]

Task 4: Object Oriented Programming (OOP) [20]

Ziel dieser Aufgabe ist die Implementierung eines kleinen Klassengerüsts. Gegeben sei dazu folgendes Klassendiagramm in UML-Notation:



Hinweis: Die Klasse "Mitarbeiter" und die Methode "addZulage" sind kursiv gesetzt.

a) Definieren Sie die abstrakte Klasse "Mitarbeiter" und leiten Sie davon die Klassen "Lehrling" und "Angestellter" ab! [2]

b) Implementieren Sie die Attribute gemäß UML-Diagramm! [6]

Hinweis: Beachten Sie Sichtbarkeiten und Unterstreichungen!

c) Implementieren Sie die Methoden gemäß UML-Diagramm! [10]

Hinweis: Beachten Sie Kursivsetzung!

d) Testen Sie ALLE Methoden in einer neu zu erstellenden "Launcher"-Klasse! Erzeugen Sie dazu mindestens je ein Objekt der Klassen "Lehrling" und "Angestellter"! [2]

Task 5: Exception Handling [10]

Ziel dieser Aufgabe ist das Abfangen von Fehlern innerhalb eines Programmes.

a) Erstellen Sie die Klasse "Monat" mit dem Attribut:

```
private int zahl;
```

und dem initialisierenden Konstruktor [2]:

```
public Monat(int monatsZahl)
```

b) Schreiben Sie außerdem folgende Methode zur Bestimmung des Monatsnamens [2]:

```
public String getMonatsname()
```

c) Lösen Sie in dieser Methode eine Ausnahme vom Typ "MonatsAusnahme" aus, wenn eine Monatszahl außerhalb des gültigen Bereiches von 1 bis 12 als Parameter übergeben wird! Geben Sie dabei die ungültige Monatszahl als Text an das Fehlerobjekt mit! [2]

d) Erstellen Sie die oben bereits verwendete Fehlerklasse "MonatsAusnahme"! [2]

e) Testen Sie den Konstruktor und die Methode! Fangen Sie Ausnahmen ab! [2]

Viel Erfolg!