

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 2017-1
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	
Duration [min]	108 von insgesamt 180
Maximum Points [Point]	60
Permitted Study Aids	Dokumentation im lokalen Netzwerk (Intranet) sowie Recherche im Internet. NICHT gestattet: * Kommunikation in jeglicher Form * Anmeldung via SSH auf dem Rechner "fileserv" * Anmeldung mit Klausur-Login nach Ende der Prüfung Dies kann leicht geprüft werden (last cs16*, Server-Log-Dateien). Bitte unterlassen Sie also Täuschungsversuche in Ihrem eigenen Interesse.
Remarks	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.

Evaluation											
Task	1	2	3	4	5	6	7	8	9	10	Summe
Points	5	10	15	20	10	0	0	0	0	0	60

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

Task 1: Operation [5]

Zweck: Berechnung der Anzahl an Sekunden im Monat Dezember als Zahlenwert.

a) Erstellen Sie vier ganzzahlige, lokale Variablen mit den Werten für: Tage im Monat Dezember, Stunden eines Tages, Minuten einer Stunde, Sekunden einer Minute! [2]

b) Berechnen Sie ihr Produkt und speichern Sie es in einer weiteren, lokalen Variablen! [2]

c) Geben Sie den Ergebniswert auf der Konsole aus! [1]

Ergebnis: Die Sekundenanzahl wurde für den Monat Dezember ausgegeben.



Task 2: Structured Programming [10]

Zweck: Ausgabe eines Sternchen-Dreiecks der folgenden Form auf dem Bildschirm:

```
*
**
***
****
*****
*****
*****
*****
*****
*****
```

- a) Erstellen Sie eine Startklasse mit "main"-Methode und zwei lokalen Variablen "zeile" und "spalte", jeweils vom Typ "int"! [2]
- b) Verwenden Sie eine Schleife Ihrer Wahl, um über die Zeilen zu zählen! [2]
- c) Betten Sie in diese eine weitere (innere) Schleife ein, welche die Spalten zählt! [2]
- d) Passen Sie das Laufkriterium der inneren Schleife so an, dass ein Dreiecksmuster entsteht! [2]
- e) Platzieren Sie den nötigen Zeilenumbruch in der äußeren Schleife! [2]

Ergebnis: Das Dreiecksmuster wurde korrekt ausgegeben.

Task 3: Procedural Programming [15]

Zweck: Rundung zweier Gleitkommazahlen.

- a) Geben Sie der Startklasse eine statische Methode namens "round", welche einen "double"-Wert zurückgibt! [2]
- b) Fügen Sie ihr als Parameter hinzu: die zu rundende Zahl des Types "double" und eine Ganzzahl, welche die Nachkommastellen repräsentiert! [2]
- c) Bilden Sie unter Verwendung einer geeigneten Methode der "Math"-Klasse die Potenz zur Basis 10! Verwenden Sie den zweiten übergebenen Parameter als Exponenten! Speichern Sie das Ergebnis in einer lokalen Variablen! [2]
- d) Verschieben Sie nun das Komma durch Multiplikation mit der eben errechneten Potenz nach rechts und runden Sie den Zahlenwert mittels geeigneter Methode der "Math"-Klasse! [2]
- e) Schieben Sie das Komma anschließend wieder um die gleiche Anzahl Stellen nach links! [2]
- f) Überladen Sie die selbst erstellte "round"-Methode für den Typ "float"!

Hinweis: Die Logik bleibt unverändert. [2]

- g) Rufen Sie die "round"-Methode aus der "main"-Methode heraus für den Zahlenwert 1234.56789 auf, wobei Nachkommastellen weggeschnitten werden sollen! Geben Sie das Ergebnis auf der Konsole aus! [2]
- h) Rufen Sie die "round"-Methode für die Zahl "123.456789f" und zwei Nachkommastellen erneut auf und geben Sie das Ergebnis aus! [1]

Ergebnis: Die beiden Zahlenwerte wurden korrekt gerundet zu: 1235.0 und 123.46.

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

Zweck: Implementierung einer Behälterklasse mit Schnittstelle.

a) Erstellen Sie eine Schnittstelle namens "IntegerList" mit folgenden Methodensignaturen [2]:

- `int getLength();` // Liefert die Länge der Liste.
- `void insertLast(int value);` // Fügt value am Ende der Liste ein.
- `int getFirst();` // Liefert das erste Element der Liste.
- `void deleteFirst();` // Löscht das erste Element der Liste.
- `boolean search(int value);` // Prüft, ob value in der Liste vorhanden ist.

b) Erstellen Sie desweiteren eine Klasse namens "ArrayIntegerList"! Geben Sie ihr ein Attribut namens "array", welches als Feld (Array) Ganzzahlen speichert! Fügen Sie außerdem ein Attribut namens "length" vom Typ "int" hinzu! Ordnen Sie beiden private Sichtbarkeit zu! [2]

c) Definieren Sie einen Konstruktor, in welchem das "array"-Attribut erzeugt wird! Lassen Sie ihn die Größe des zu erzeugenden Feldes (Arrays) als Parameter entgegennehmen! [2]

d) Instanzieren Sie die Behälterklasse "ArrayIntegerList" mit der Größe 100 in der "main"-Methode einer Startklasse namens "Launcher"! Fügen Sie ihr vier Elemente hinzu! [2]

e) Testen Sie alle ihre Methoden! [2]

f) Lassen Sie die Klasse "ArrayIntegerList" von der Schnittstelle "IntegerList" erben! Implementieren Sie die geforderten Methoden in adäquater Weise! [10]

Ergebnis: Die Funktionsfähigkeit der Behälterklasse wurde nachgewiesen.

Task 5: Exception Handling [10]

Zweck: Eine Ganzzahl soll eingelesen werden.

a) Erzeugen Sie eine "Scanner"-Instanz zum Einlesen von Nutzereingaben via Konsole! [2]

b) Realisieren Sie mittels "while" eine Endlosschleife, die permanent auf Nutzereingaben wartet! Bauen Sie eine Abbruchbedingung in den Schleifenkörper ein, welche ein zuvor auf "false" initialisiertes Flag prüft! [2]

c) Fordern Sie den Anwender über eine kurze Konsolennachricht zur Eingabe einer Zahl auf! Empfangen Sie mittels geeigneter Methode des "Scanner"-Objektes eine Ganzzahl und geben Sie sie zur Kontrolle auf der Konsole aus! [2]

d) Fangen Sie eine möglicherweise auftretende "InputMismatchException" ab! Informieren Sie den Anwender mittels einer kurzen Nachricht auf der Konsole! [2]

e) Vermeiden Sie die im Fehlerfall auftretenden Endlosausgaben durch Einlesen und ignorieren der fehlerhaften Zeichen im "catch"-Block! [2]

Ergebnis: Falscheingaben werden gemeldet und Ganzzahlen korrekt erkannt.

Viel Erfolg!