

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 2014
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	2014-12-11
Duration [min]	108
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

Im Rahmen der Klausur werden Ihre Programmierkenntnisse anhand kleiner, voneinander unabhängiger Java-Programme geprüft.

Task 1: Operator and Structure [10]

Ziel dieser Aufgabe ist die Ausgabe einer ganzen Zahl in Binärdarstellung (32 Bit).

Hinweis: Das Bit mit der Nummer i (Nummerierung beginnt bei 0) in der Binärdarstellung von "zahl" hat den Wert 1 genau dann, wenn der Ausdruck " $\text{zahl} \& (1 \ll i)$ " von 0 verschieden ist.

- Erstellen Sie ein Java-Programm mit der Hauptklasse "Launcher" und "main"-Methode! [1]
- Deklarieren Sie drei lokale Variablen namens: "zahl", "bit" und "ergebnis", zur Speicherung je einer ganzen Zahl! Initialisieren Sie sie mit den Werten: 4711, 0 und 0! [3]
- Verwenden Sie eine klassische "for"-Schleife, um die einzelnen 32 Bit rückwärts, d. h. "runter" zu zählen! [1]

d) Wenden Sie innerhalb der Schleife den Schiebeoperator << auf die Ziffer 1 (eins) an, so dass das Bit, welches den Wert 1 repräsentiert, um i Stellen nach links wandert! Speichern Sie das Ergebnis in der Variablen "bit"! [1]

e) Verknüpfen Sie die Variablen "zahl" und "bit" mittels des &-Bitoperators! Speichern Sie das Ergebnis in der Variablen "ergebnis"! [1]

f) Testen Sie für eine Programmverzweigung die Variable "ergebnis" auf den Wert 0! Geben Sie im Erfolgsfalle die Zahl 0 als Zeichen auf der Konsole aus; ansonsten die Zahl 1! [2]

g) Verwenden Sie eine weitere Bedingung, in welcher Sie die Laufvariable per Modulo-Operator durch die Ziffer 8 teilen! Geben Sie für den Fall, dass es keinen Rest gibt, ein Leerzeichen aus! Hinweis: Dadurch sollen die vier Byte der ausgegebenen Bitfolge optisch abgesetzt werden. [1]

Task 2: String [10]

In dieser Aufgabe soll der Teil einer Zeichenkette bestimmt werden, der mit einem Startzeichen beginnt und mit einem Schlusszeichen endet.

a) Definieren Sie eine statische Methode namens "extrahiere" mit drei Parametern: dem zu untersuchenden String, dem Startzeichen und dem Endzeichen, letztere vom Typ "char"! [2]

b) Bestimmen Sie mittels geeigneter String-Methoden jeweils die Positionen des Start- und Endzeichens und speichern Sie beide in je einer lokalen Variablen! [2]

c) Testen Sie, ob der Anfangs- hinter dem Endindex liegt! Geben Sie in diesem Falle "null" zurück! [1]

d) Testen Sie, ob der Anfangsindex ungültig ist! Geben Sie in diesem Falle ebenfalls "null" zurück! [1]

e) Bestimmen Sie schließlich mittels geeigneter Methode der String-Klasse die Teilzeichenkette und geben Sie sie als Ergebniswert der Methode zurück! [2]

f) Definieren Sie in der "main"-Methode die folgende Zeichenkette: "Er sagte laut \"Guten Tag!\"" Hinweis: Die doppelten Anführungszeichen in der Zeichenkette müssen speziell gekennzeichnet werden. Extrahieren Sie durch Aufruf der gleichnamigen Methode die Teilzeichenkette "Guten Tag!" und lassen Sie sie auf der Konsole ausgeben! [2]

Task 3: Procedure [10]

Ziel dieser Aufgabe ist die Bestimmung des Maximums beliebig vieler ganzer Zahlen.

a) Definieren Sie eine statische Methode namens "max" mit einem Rückgabewert des Types "int"! Verwenden Sie folgende Notation für eine Liste beliebig vieler Parameter [2]:

(int... werte)

b) Speichern Sie den Wert des ersten Parameters in einer lokalen Variablen namens "max" des Types "int"! [2]

Hinweis: Auf die einzelnen Werte der Parameterliste wird wie bei einem Array zugegriffen.

c) Iterieren Sie mittels "for-each"-Schleife durch die Parameterliste! [2]

d) Speichern Sie den jeweils aktuellen Wert in der lokalen Variablen "max" und geben Sie sie am Ende der Methode zurück! [2]

e) Rufen Sie die "max"-Methode in der "main"-Methode mindestens zweimal auf [2]:

- einmal mit direkt angegebenen, verschiedenen, auch negativen Werten;
- ein anderes Mal mit als Array übergebenen Werten!

Task 4: Class [20]

Ziel dieser Aufgabe ist die Anwendung des Zinseszinses auf einen Kontostand. Gegeben seien 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)

Sparbuch
- kontonummer : int
- kapital : double
- zinsfuß : double
+ zahleEin(betrag : double)
+ hebeAb(betrag : double)
+ berechneZins(laufzeit : int) : double
+ berechneZinseszins(laufzeit : int) : double

- Implementieren Sie die in obigem UML-Klassendiagramm abgebildete Klasse "Sparbuch"! [6]
- Erstellen Sie Zugriffsmethoden zum Auslesen wie auch Zuweisen eines jeden Attributes! [2]
- Verpassen Sie der Klasse einen Konstruktor, der Parameter zur Initialisierung aller Attribute entgegennimmt! [2]
- Vervollständigen Sie die Methoden "zahleEin" bzw. "hebeAb" in der Weise, dass sich das Guthaben ("kapital") um den übergebenen Betrag erhöht bzw. vermindert! [2]
- Implementieren Sie die Methoden "berechneZins" und "berechneZinseszins"! Iterieren Sie mittels klassischer "for"-Schleife über die Laufzeit! [4]
- Instanziieren Sie den Typ "Sparbuch" in der "main"-Methode zweimal, mit jeweils unterschiedlichen Kontonummern, einem Startguthaben von 0 sowie einem Zinsfuß von 5 %! Zahlen Sie auf beide Sparbücher je 1 Geldeinheit ein! [2]
- Wenden Sie für eine Laufzeit von 2014 Jahren die Methode "berechneZins" auf das erste und die Methode "berechneZinseszins" auf das zweite Sparbuch an! Weisen Sie die Ergebnisse als Guthaben (Kapital) jeweils neu zu! Geben Sie beide Kontostände auf der Konsole aus! [2]

Task 5: Wrapper [10+2]

Im Mittelpunkt dieser Aufgabe steht der Umgang mit Hüllklassen.

- Initialisieren Sie eine lokale Variable des Types "String" mit dem Wert "1971"! Wandeln Sie die Zeichenkette per statischer Methode einer passenden Hüllklasse um in einen Wert vom Typ "Integer"! Extrahieren Sie den eigentlichen primitiven Zahlenwert und speichern Sie ihn in einer dritten lokalen Variablen vom Typ "int"! Geben Sie das Ergebnis auf der Konsole aus! [2]
- Initialisieren Sie eine lokale Variable des Types "double" mit dem Wert 123.45! Kapseln Sie diesen Wert mittels eines "Double"-Objektes! Konvertieren Sie das Objekt explizit in eine Zeichenkette! Geben Sie diese schließlich auf der Konsole aus! [2]
- Geben Sie die ganze Zahl 150 in dualer, oktaler und hexadezimaler Schreibweise aus! [2]
- Prüfen Sie mittels geeigneter Methode einer Hüllklasse, ob es sich bei dem Zeichen '8' um eine Zahl handelt! Geben Sie das Ergebnis auf der Konsole aus! [2]
- Bestimmen Sie Minimum und Maximum des Wertebereiches des Datentypes "Byte"! Geben Sie beide auf der Konsole aus! [2]
- Zusatzaufgabe: Provozieren Sie folgende Ausgabe: "NaN"! Provozieren Sie die Ausgabe der definierten Werte "Infinity" und "-Infinity"! [2]

Viel Erfolg!

Schlussbemerkung zu Aufgabe 4:

Sind Ihre Formeln richtig, so wurden im Jahre 2014 bei Verwendung des einfachen Zinses circa 100 Geldeinheiten leistungslos über Zinsen verdient. Bei Verwendung des Zinseszinses jedoch umfasst das Vermögen mehrere Tausend Milliarden Erdkugeln aus purem Gold. Damit ist der exponentielle Verlauf bei Anwendung des Zinseszinses mathematisch nachgewiesen.

In Wirtschaft und Gesellschaft stehen jedem Guthaben spiegelbildlich Schulden in gleicher Höhe gegenüber. Der höchst gefährliche Zinseszinseffekt unseres heutigen Geldsystems verursacht u. A. Wachstum, Umweltzerstörung, die Kluft zwischen Fleißig und Reich und einen historisch nachweisbaren Zusammenbruch des Geldsystems aller 50 bis 70 Jahre, leider oft verbunden mit Krieg.

Eine Lösung liegt im so genannten "Fließenden Geld", auf welches regelmäßig eine Gebühr bei Hortung anfällt. Weiterführende Quellen:

- <http://www.woistunsergeld.de/>
- <http://www.lust-auf-neues-geld.de/>
- <http://www.humane-wirtschaft.de/>
- <http://www.macht-geld-sinn.de/>
- <http://www.cgw.de/>
- <http://www.regiogeld.de/>
- <http://www.lindentaler.org/>
- <http://www.zeit.de/2010/52/Woergl?page=all>