

Miniklausur 3 zur Vorlesung
Grundlagen der Programmierung
Aufgabenblatt

- Sie dürfen NICHT in Gruppen arbeiten. Mit der Teilnahme versichern Sie, dass Sie die Aufgaben alleine bearbeiten. Bearbeitungszeit ist jeweils von **09:00** bis **10:00**. Sie benötigen am Semesterende 50% der Gesamtpunktzahl aller 6 Miniklausuren.
- Die Angaben zur Evaluation bedeuten:
 - Bei “*Evaluation: Exakter Text*” müssen alle ausgegebenen Zeichen inklusive ihrer Formatierungen mit der Vorgabe übereinstimmen.
 - Bei “*Evaluation: vorgegebene main*” gibt es eine vorgegebene `main`-Methode, die die von Ihnen zu schreibenden Methoden aufruft. Diese Methode darf NICHT verändert werden.
 - Die von der Evaluierungsroutine ausgewerteten Teile des Ein/Ausgabeprotokolls werden in **blau** dargestellt, die Eingaben in **rot**.
- Es wird bei allen Aufgaben verlangt, Eingaben über den Scanner vorzunehmen. Wenn Sie das Einlesen anders implementieren, werden die Evaluierungen NICHT funktionieren und Sie erhalten KEINE Punkte, selbst wenn Ihre Lösung ansonsten in Ordnung sein sollte.

Aufgabe 1 (Evaluation: Exakter Text)

(10 Punkte)

Schreiben Sie ein Java-Programm, das als Eingabe eine `int`-Zahl n einliest und als Ausgabe (ähnlich dem Beispiel `K3B14_Flagge` und der entsprechenden Übungsaufgabe) Muster (aus $n \times n$ Zeichen) folgender Art generiert:

<p>Eingabe: 6</p> <p>XXXXXX XXXXXo XXXXXoo XXXooo XXoooo Xooooo</p>	<p>Eingabe: 7</p> <p>XXXXXXX XXXXXXo XXXXXoo XXXooo XXoooo XXooooo Xoooooo</p>
--	---

Achten Sie auch darauf, dass keine Zeilenenden fehlen und dass Sie keine unnötigen Leerzeilen erzeugen, ansonsten funktioniert die Evaluation nicht. Die im Muster verwendeten Buchstaben sind das große X und das kleine o.

Aufgabe 2 (Evaluation: vorgegebene main)

(10 Punkte)

Schreiben Sie eine statische Java-Methode `meineFunktion` mit vier Parametern, die die folgende Formel berechnet:

$$\text{meineFunktion}(a, b, c, d) = \begin{cases} a + b, & \text{falls } c < d \\ a - b, & \text{falls } c \geq d \end{cases}$$

<p>Eingabe: 1 2 3 4</p> <p>Ausgabe: 3</p>	<p>Eingabe: 4 3 2 1</p> <p>Ausgabe: 1</p>
---	---

Eine passende `main`-Methode ist vorgegeben und darf nicht verändert werden. Welche Signatur `meineFunktion` haben sollte (passend zur `main`-Methode), müssen Sie selbst herausfinden.

Aufgabe 3 (Evaluation: vorgegebene main)

(10 Punkte)

Mit der vorgegebenen `main`-Methode wird ein globales(!) Array `data` initialisiert. Schreiben Sie dazu eine statische Methode `meinMaximum(a,b)`, die den größten Wert von `data` für Indizes im Bereich von `a` bis `b` (jeweils inklusive) bestimmt und zurückgibt. Sie können dabei davon ausgehen, dass $a \leq b$ gilt und diese beiden Werte auch im für Indizes erlaubten Bereich liegen.

Die vorgegebene `main`-Methode enthält einige Aufrufe der Methode `meinMaximum(a,b)`.

Anzahl: 5 Werte: 1 5 2 4 3 Tests: 0 1 5 2 3 4 4 4 3 0 4 5	Anzahl: 6 Werte: 5 4 3 2 1 0 Tests: 1 2 4 2 3 3 3 5 2 5 5 0
---	---

Aufgabe 4 (Evaluation: vorgegebene main)

(10 Punkte)

Schreiben Sie eine statische Java-Methode `meineAusgabe` des Typs `void`, die ihren `int`-Parameter in Dezimalschreibweise ausgibt (mit `System.out.print(...)`), wobei aber die **Reihenfolge der Stellen umgedreht** ist (erst Einer, dann Zehner, dann Hunderter etc.). Sie können davon ausgehen, dass der Parameter > 0 ist. Eine `main`-Methode ist wieder vorgegeben:

Eingabe: 13579 Ausgabe: 97531	Eingabe: 1000 Ausgabe: 0001
----------------------------------	--------------------------------

Tipp: Sie können z.B. die Operatoren “%” und “/” in einer Schleife geeignet einsetzen, aber es gibt auch andere Wege, um Zahlen in Strings umzuwandeln.

Aufgabe 5 (Evaluation: vorgegebene main)

(10 Punkte)

Das vorgegebene Programm liest wie üblich an Anfang Daten in ein Array ein und gibt sie am Ende schließlich wieder aus. Vertauschen Sie nun die Werte im Array an der gekennzeichneten Stelle in der Mitte des Programmes so, dass die Reihenfolge der Werte bei der abschließenden Ausgabe umgedreht ist. Das vorgegebene Programm darf ansonsten nicht verändert werden. Beispiel:

Anzahl: 5 Werte: 1 2 3 4 5 umgedrehte Reihenfolge: 5 4 3 2 1
--

Achtung: Ziel ist nicht, dass Sie selbst die Werte in umgekehrter Reihenfolge ausgeben, sondern dass Sie die Werte im Array entsprechend vertauschen! Die Ausgabe geschieht bereits über das vorgegebene Programm!