



## Informationen zur Klausur

**DIE KLAUSUR LIEGEN LASSEN,  
BIS SIE LOSLEGEN DÜRFEN!**

**Prüfungsdauer:** 90 Minuten

**Wertung:** Insgesamt können 100 Punkte erreicht werden. Die pro Aufgabe bzw. Teilaufgabe erzielbaren Punkte sind angegeben. *Die erreichten Punkte werden Ihnen hälftig auf die Gesamtleistung angerechnet.*

**Hilfsmittel:** Es sind keine Hilfsmittel erlaubt. Es darf kein eigenes Papier verwendet werden.

**Hinweise:**

- Bitte verwenden Sie keinen Rotstift oder Bleistift.
- Ihre Lösungen tragen Sie bitte nur auf dem Lösungsblatt ein.
- Schreiben Sie Ihre Antworten bitte ausschließlich in die auf dem Lösungsblatt vorgesehenen Abschnitte.
- Sie müssen mit den vorhandenen leeren Blättern für Ihre Notizen und Lösungsskizzen auskommen. Sie bekommen kein weiteres Papier.
- Es werden nur leserliche Klausurlösungen bewertet!
- Die Angabe der Lösungen hat sich in Art und Form an den in der Veranstaltung verwendeten Konventionen zu orientieren.
- Sie geben am Ende der Klausur nur das Lösungsblatt ab. Die Aufgabenblätter sind für den Verbleib bei Ihnen bestimmt.



Liebe Studierende,

es gibt eine Menge an Dingen, die wir Ihnen zu Beginn der Prüfung sagen werden und teils aufgrund der Umstände auch sagen müssen. Hier haben Sie die Chance noch einmal alles nachzulesen.

- Es ist während der gesamten Klausur der Mundschutz zu tragen.
- Während der Klausur werden keine Fragen zum Verständnis oder zur Klärung einer Klausuraufgabe beantwortet.
- Smartphones, Smartwatches, Kopfhörer und andere Hightech-Geräte dürfen aus verständlichen Gründen nicht verwendet werden.
- Zu Toilettengängen: Wenn Sie den Raum verlassen müssen, melden Sie sich. Es kann immer nur eine Person die Toilette benutzen. Vergessen Sie nicht, sich gründlich die Hände zu waschen, bevor Sie die Halle wieder betreten.
- Wenn Sie vorzeitig die Klausur abgeben wollen, machen Sie sich durch Aufzeigen bemerkbar. Wir sammeln das Lösungsblatt ein. Verlassen Sie anschließend bitte leise den Saal.
- 20 Minuten vor Ende der Klausur bitte keine vorzeitigen Abgaben mehr. Das macht zu viel Unruhe und erzeugt Stress.
- Wenn Sie krank sind oder Grippe Symptome zeigen, verlassen Sie bitte sofort die Halle.
- Wenn Sie unter einer chronischen Erkrankung leiden, die mit erkältungsähnlichen Symptomen einhergeht und somit nicht ein Anzeichen einer möglichen Corona-Ansteckung ist, müssen Sie ein ärztliches Attest vorweisen.
- Legen Sie Ihren Studierendenausweis an den äußeren Rand ihres Tisches.
- Tragen Sie Ihren Namen und Ihre Matrikelnummer auf dem Lösungsblatt ein und unterschreiben Sie das Lösungsblatt.
- In die Lösungsfelder bitte erst dann eine Lösung eintragen, wenn Sie sich sicher sind.

Ich werde die Klausur vor dem Ende des Monats korrigieren und Ihre Noten im Prüfungssystem eintragen.

Viel Erfolg und herzliche Grüße,

Dominikus Herzberg

# Aufgabenteil zur PiS-Klausur, SoSe 2021

Prüfer: Prof. Dr. Dominikus Herzberg, Fachbereich MNI



Bitte nutzen Sie für Ihre Antworten ausschließlich die separat ausgehändigten Lösungsblätter! Sie geben am Ende der Klausur nur die Lösungsblätter ab! Nutzen Sie die Rückseiten der Aufgabenblätter als Notizblätter.

## 1 Objektvergleich (12 + 4 = 16 Punkte)

---

Gegeben sei eine Klasse `Rectangle`, die die Seitenlängen einer Rechteck-Instanz mittels zweier privater `float`-Variablen `height` und `width` verwaltet.

1. Implementieren Sie die Methode `equals` nach dem in der Veranstaltung erlernten Schema.
2. Implementieren Sie die Methode `int hashCode` für `Rectangle` in einer Weise, so dass kein besonderes Verständnis zur Berechnung eines HashCodes erforderlich ist.

## 2 Collections (4 + 6 + 3 + 2 = 15 Punkte)

---

1. Erstellen Sie eine Variable `listOfSetOfNumbers`, die eine immutable Liste mit immutable Mengen von Ganzzahlen verwaltet. Die Variable werde zugleich initialisiert, wobei die erste Menge aus den Zahlen 2, 5 und 6 besteht und die zweite Menge aus den Zahlen 1, 4, 5 und 8.
2. Erstellen Sie eine klassenfreie Methode (so wie man das in der JShell ohne Klassenkontext machen kann) namens `level`, die eine Menge von Ganzzahlen entgegen nimmt und eine *neue* Menge von Ganzzahlen zurückliefert. Die Methode halbiert gerade Zahlen und verdoppelt ungerade Zahlen. Für die Implementierung wird nur ein Semikolon benötigt.
3. Erstellen Sie einen Ausdruck, der aus `listOfSetOfNumbers` eine *neue* Liste erstellt, wobei die *neuen* Mengen in der Liste aus der Anwendung von `level` hervorgehen.
4. Was liefert der Ausdruck als Ergebnis, wenn er in der JShell ausgeführt wird?

## 3 Functional Interfaces (3 + 3 + 3 + 3 = 12 Punkte)

---

Erstellen Sie beispielhafte Lambda-Ausdrücke als Einzeiler vom Typ `Predicate`, `Function`, `Supplier` und `Consumer`; in allen Fällen wird eine Liste verarbeitet. Die Verwendung von Methoden-Referenzen ist nicht erlaubt. Der Ausdruck muss in der JShell ausführbar sein und darf keine Semikolons enthalten. Zur Anregung: Die Anzahl der Elemente einer Liste wird mit `size()` ermittelt.

## 4 Fakultät (10 Punkte)

---

Die Fakultät  $n!$  einer Zahl  $n \geq 1$  ist das Produkt der Zahlen von 1 bis  $n$ . Es gilt also  $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ . Implementieren Sie die Methode `factorial` und berechnen Sie die Fakultät mithilfe eines Stroms und ausschließlich mit den Methoden `rangeClosed` und `reduce`.

```
long factorial(long n) {  
    assert n >= 1;  
    return  
}
```

## 5 Median (9 + 12 = 21 Punkte)

---

Der Median einer Zahlenliste ist ihr Zentralwert. Damit ist der Wert gemeint, der in der *sortierten* Zahlenliste in der Mitte steht. Bei einer ungeraden Anzahl gibt es eine eindeutige Mitte. In der Liste 10, 13 und 15 (die Liste ist bereits sortiert) steht 13 in der Mitte und ist der Median. Bei einer ungeraden Anzahl liegt die Mitte zwischen zwei Zahlen, der Median ist das arithmetische Mittel dieser beiden Zahlen. Die (sortierte) Liste aus 10, 13, 15 und 17 hat  $(13 + 15)/2 = 14$  als Median.

Ergänzen Sie den nachstehenden Code. Sie dürfen weder ein `if` noch einen ternären Ausdruck verwenden. Im `return` wird zur Berechnung des Medians direkt auf die Elemente des Arrays zugegriffen. Insgesamt kommen im Rumpf von `medianA` nur vier Semikolons vor. Beachten Sie, dass `lower` und `upper` in der zweiten Implementierung `medianB` wiederverwendet werden.

```
double medianA(double... values) {  
    Arrays.sort(values);  
    int lower =  
    int upper =  
    return  
}
```

In der zweiten Variante der Implementierung verwenden Sie einen `stream` bei der Berechnung. Verwenden Sie u.a. die Methoden `average()` (liefert ein `Optional` zurück), `limit(long maxSize)` (begrenzt einen Strom auf maximal `maxSize` an Elementen) und `skip(long n)` (verkürzt den Strom, indem es die ersten `n` Elemente verwirft). Insgesamt kommen im Rumpf von `medianB` nur drei Semikolons vor. Einzig die `return`-Anweisung ist als Lösung von Interesse.

```
double medianB(double... values) {  
    int lower = ... // Code ist identisch mit lower aus medianA  
    int upper = ... // Code ist identisch mit upper aus medianA  
    return Arrays.stream  
}
```

## 6 Map (4 + 4 = 8 Punkte)

---

1. Erstellen Sie eine Variable namens `calc`, der eine Map zugewiesen wird, wobei den Zeichen `+` und `-` jeweils eine Funktion in Form eines Lambda-Ausdrucks zugeordnet wird; in dem einen Fall addiert die Funktion zwei Ganzzahlen, im anderen subtrahiert sie zwei Ganzzahlen.
2. Erstellen Sie einen Ausdruck, der die Variable `calc` nutzt, um auf die mit dem Additionszeichen assoziierte Funktion zuzugreifen, und die Funktion mit den Werten 2 und 3 aufruft.

## 7 Fragen (6 × 3 = 18 Punkte)

---

*Hinweis: Pro Frage kann mehr als eine Antwort korrekt sein. Die Fragen sind so formuliert, dass in der Regel nicht ableitbar ist, ob eine oder mehr Antworten zutreffen.*

**Frage A:** Die JVM ist als nebenläufige Architektur aufgesetzt. Welche Ressource verwaltet ein Thread separat für sich, d.h. die Ressource wird nicht mit anderen Threads geteilt?

1. Program Counter
2. Nativer Methodenstapel
3. Run-Time Constant Pool
4. JVM-Stapel
5. Heap

**Frage B:** Ein Thread kann sich in dem folgenden Zustand befinden:

1. COMPLETED
2. COMPETING
3. BLOCKED
4. TERMINATED
5. RECYCLED

**Frage C:** Welche alternativen Möglichkeiten der nebenläufigen und parallelen Programmierung gibt es, statt direkt mit Threads zu arbeiten?

1. Reactive Programmierung
2. Logische Programmierung
3. Strombasierte Programmierung
4. Ergebnisbasierte Programmierung
5. Nachrichtenbasierte Programmierung

**Frage D:** Welche Syntax für einen Lambda-Ausdruck ist gültig?

1. `(x) -> x + 1`
2. `() -> {}`

3. `(String s) -> s.length()`
4. `int x -> x++`
5. `t -> { t.start(); }`

**Frage E:** Eine Klasse E erweitere die Klasse X. Was syntaktisch möglich ist, ist semantisch zu überprüfen. Welche Aussage ist gültig?

1. Die erweiterte Klasse muss den *isa*-Test bestehen
2. Die erweiterte Klasse muss isomorph zur Oberklasse sein
3. Die Klassenhierarchie sollte nicht breiter sein als sie tief ist
4. Die Klassen müssen das Liskov'sche Substitutionsprinzip erfüllen
5. Klassennamen sind wichtig, da der Name die Syntax bestimmt

**Frage F:** Welche Aussage zu Datenstrukturen ist korrekt?

1. Eine Liste ist eine assoziative Verknüpfung von Listenelementen
2. Eine Map ist von der Zugriffszeit genauso effizient wie eine Liste
3. Eine Queue ist dasselbe wie eine Liste
4. Ein Set kann nicht leer sein
5. Keine der vorigen Antworten trifft zu

(Notizblatt)





(Notizblatt)





# Lösungsblatt zur PiS-Klausur

Bitte nutzen Sie für Ihre Antworten ausschließlich dieses ausgehändigte Lösungsblatt!  
Sie geben am Ende der Klausur nur das Lösungsblatt ab!

<b>Nachname, Vorname</b>	<b>Matrikelnummer</b>	<b>Note</b>
<b>Unterschrift</b> 25.09.2021		

## 1. Objektvergleich ( $12 + 4 = 16$ )

1.

2.

## 2. Collections ( $4 + 6 + 3 + 2 = 15$ )

1.

2.

3.

4.



### 3. Functional Interfaces (3 + 3 + 3 + 3 = 12)

*Predicate:*

*Function:*

*Supplier:*

*Consumer:*

### 4. Fakultät (10)

```
return
```

### 5. Median (9 + 12 = 21)

#### 5.1 medianA

```
int lower =
```

```
int upper =
```

```
return
```

#### 5.2 medianB

```
return Arrays.stream
```

### 6. Map (4 + 4 = 8)

1.

2.

### 7. Fragen (6 x 3 = 18)

1. 2. 3. 4. 5.	Punkte	1. 2. 3. 4. 5.	Punkte
A <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		D <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
B <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		E <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
C <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		F <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	

Sollten Sie eine Lösung revidieren wollen, streichen Sie die betreffende Reihe durch und notieren Sie hier neben der Tabelle Ihre korrigierte Lösung im Stil von Z13 (die hypothetische Frage Z, Antwort 1 und 3).