

Prog 1 Klausur Struktur & System analyse.

1S Aufgaben insgesamt: → Seine Aufgabe: 1 punkt.

Große Struktur:

- 1 - 5 (Behauptungen) Ankreuz aufgabe richtig/falsch
1. Java ist Pass-by-Value.
 2. Der 'Static-modifizierer' kontrolliert die Sichtbarkeit eines Attributs.
 3. Die Operation && und || haben die gleiche Randordnung.
 4. Die folgende Methode führt zu einem Syntaxerror.
 5. Die for-Schleife und while-Schleife drucken die gleichen Ausgaben

Syntax in Java: → Theoretische mechaniken.

1. Java ist pass by value → Es gibt pass by reference und pass by value.

Hintergrund Prozess
↳ (Speicher)

← Eine Objekt variable zeigt immer auf das echte Objekt im heap

2. Static-modifier → Es gibt Access modifiers und non Access modifiers.

Daten-Kapselung

↳ Syntax

↳ (Access/non-access)

Access	non-Access
private	static
protected	final
public	abstract
	synchronized
	transient
	volatile
	strictfp

3. && und || → Randordnung: priorität Operatoren

Prio 1 - 7:

Hintergrund Prozess
↳ (Operationen)

→ &&, || → prio 7 bis 1

4. Syntax error

```
public String myMethod(String s){ ← illegal start of method.  
    s += "+\\\"";  
    return s;  
}
```

Es geht darum einen Syntax Fehler zu finden.

5. for-/while-Schleife vergleich (code Inhalt) (Syntax)

```
for (int i = 0; i < 10; i = i + 2) { | int i = 0;  
    System.out.println(i); | while (i < 10) { ← zeigt jedes  
} | if (i % 2 == 0) { zweite print.  
    System.out.println(i);  
} | i++;  
}
```

↑
ist gleich!

1 - 5 : Aufgaben typen:

Aufgaben tags:

Auktion:	Hintergrund Prozess: II	(Speicher) (Operationen)
	Syntax Bezogenes: III	(Access/non-Access) (code Inhalt) (Syntax error)

- 6 - 10 (Was & Wie Fragen) multiple choice

6. Was ist Wert von c wenn code ausgeführt.

7. Was ist Wert von c wenn code ausgeführt.

8. Java doc (Informationen entnehmen)

9. Was wird in Konsole ausgegeben.

10. Wieviel mal wird in Konsole ausgegeben.

Frage: → code
(Bibliothek)

6. (1 Punkt)

Was ist der Wert von Variable c, wenn wir die folgenden Zeilen Code ausführen?

```
double a = 12.345;  
double b = 67.890;  
double c = Math.min(Math.round(a), Math.floor(b));  
○ 12.0  
○ 13.0  
○ 67.0  
○ 68.0
```

math library

Frage: → code
(Wertzuweisung)

↳ klassische
Frage: a = 200

↳ Welcher Wert
hat e.g. C nach
mehrfachen
Wertzuweisungen?

7. (1 Punkt)
Was ist der Wert von Variable c, wenn wir die folgenden Zeilen Code ausführen?

int a = 200;
int b = 0;

switch (a){
case 100:
 b = 100;
case 200: case 300:
 b = 200; ← b = 200
 b = 300; ← b = 300 ← letzte Zuweisung
 break;
default:
 b = 999;
 break;
}
int c = b; }

C = 300

○ 0
○ 100
○ 200
○ 300
○ 999

Frage: → Javadoc

(Javadoc)

↳ Rückgabewert
↓

in Eclipse kann man
von jeder Java
Methode, die
Definition anschauen.

8. (1 Punkt)
Hier haben wir einen Screenshot aus einem Teil einer Javadoc:

Class LocalDate
java.lang.Object
java.time.LocalDate

All Implemented Interfaces:
Serializable, Comparable<ChronoLocalDate>, ChronoLocalDate, Temporal, TemporalAccessor, TemporalAdjuster

Public final class LocalDate
extends Object
implements Temporal, TemporalAdjuster, ChronoLocalDate, Serializable

LocalDate is an immutable object that represents a date.....

Method Summary

Modifier and Type	Method and Description
LocalDateTime	atTime(int hour, int minute, int second) Combines this date with a time to create a LocalDateTime.

Laut der Javadoc, was ist der Typ der Rückgabe für die atTime()-Methode?

- int
○ long
○ LocalDate
○ LocalDateTime
○ keine Rückgabe

↳ Press command, hover the cursor on the function name,
click it (MAC)
↓ VS-Code

Frage: → code

(Random)

String[] fussballWeltmeister = {
 "Belgien",
 "England",
 "Frankreich",
 "Kroatien"
};
Random random = new Random(1234); rand: 1-4.
int position = random.nextInt(4); konstruktor
System.out.println(fussballWeltmeister[position]); random position

- Belgien
○ Deutschland
○ Wer weiß? Der gedruckte String wird zufällig und jede Ausführung wird es anders
○ Wer weiß? Der gedruckte String wird zufällig, aber jede Ausführung wird der gleiche
String gedruckt

Frage: → code

(Syntax)

10. (1 Punkt)
Wir haben eine kleine Methode:

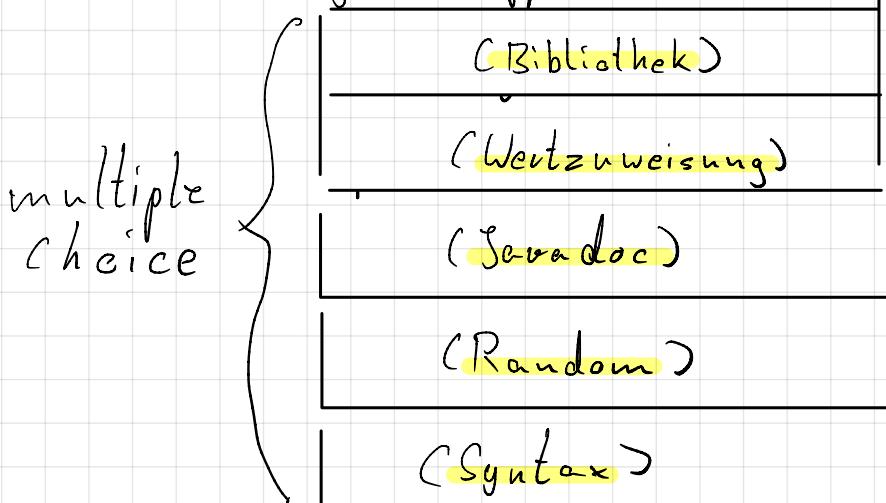
```
public int myMethod(int a) {  
    int returnInt = a;  
    System.out.println("wieder");  
  
    if (a > 1) {  
        returnInt += myMethod(a - 3);  
    }  
    else {  
        returnInt = 0;  
    }  
  
    return returnInt;  
}
```

Wie viele Mal wird wieder in der Console gedruckt, wenn wir die folgende Zeile Code ausführen?

```
int b = myMethod(11);
```

3 Mal
 4 Mal
 5 Mal
 6 Mal

6 - 10: Aufgaben typen:



- 11 - 15 (Code Abschnitte) → Konsolen - Ausgabe

11. Set, HashSet

12. String editing methods

13 . . . } methoden überladen
14 . . . }
15 . . . }

11. (1 Punkt)

```
Set<Integer> set = new HashSet<Integer>();  
set.add(1); set.add(2);  
set.add(1); set.add(2); } adding to set  
System.out.println(set.size()); } print size
```

12. (1 Punkt)

```
String s = "And I-ii-aaai, will always (atmen!), love you-uuu-uu-wah";
String[] tokens = s.split("-");
System.out.println(tokens[4]);
```

→ Split string at every '-'.

↳ String editing methods

Für Aufgaben 13, 14 und 15 benutzen wir die folgende Klasse:

```
public class Student { → class
    public String name = "";
    public int alter = 0;
    public String studiengang = "Medieninformatik"; } attribut

    Konstruktor
    ↘
    public Student(String name){ → parameter: name
        this.name = name; → überladen.
    } ↑ setter-element
    public Student(String name, int alter){ → parameter: name, alter
        this(name, alter, "Musik und Tanz"); → überladen.
    }
    public Student(String name, int alter, String studiengang){ → parameter: name, alter, studiengang, überladen.
        this.name = name;
        this.alter = alter;
        this.studiengang = studiengang;
    }
}
```

13. (1 Punkt)

Was wird in der Console gedruckt, wenn wir die folgenden Zeilen Code ausführen?

```
Student anna = new Student("Anna Alt");
System.out.println(anna.studiengang);
```

Konstruktor

14. (1 Punkt)

Was wird in der Console gedruckt, wenn wir die folgenden Zeilen Code ausführen?

```
Student bob = new Student("Bob Berg", 20);
System.out.println(bob.studiengang);
```

Erster überladener Konstruktor

15. (1 Punkt)

Was wird in der Console gedruckt, wenn wir die folgenden Zeilen Code ausführen?

```
Student carla = new Student("Carla Canter", 21, "Technische Informatik");
System.out.println(carla.studiengang);
```

Zweiter überladener Konstruktor

- 16 → coding Aufgabe
Jedamit zuerst anfangen

16. (15 Punkte) Diese Aufgabe handelt von Methoden.

Wir sind eine Firma und wir versenden Waren zu unseren Kunden. Ein Kunde wohnt entweder in Berlin oder in Brandenburg. Die Versandart ist entweder normal oder express.

Wir möchten berechnen, wie viel Versandgebühr es kostet:

- Die Basis-Gebühr ist immer 10€.
- Wenn der Kunde in Brandenburg wohnt, gibt es einen Aufpreis von 5€.
- Wenn der Kunde Express-Versand möchte, gäbe es einen Aufpreis von 20€ (egal ob der Wohnort in Berlin oder in Brandenburg ist).

Wir möchten die Berechnung als 3 überladene Methoden (overloaded methods) implementieren:

Methode
überladen

Name der Methode	Eingabe(n)	Rückgabe	Inhalt der Methode
1. Methode berechnen	Keine	Ein int – Versandgebühr	Diese Methode ruft die 2. Methode auf, wobei der Default-Wohnort Berlin angegeben wird.
2. Methode berechnen	1 Eingabe: – ein String – Wohnort	Ein int – Versandgebühr	Diese Methode ruft die Hauptmethode auf, wobei der gegebene Wohnort und die Default-Versandart normal angegeben werden.
3. Methode Hauptmethode berechnen	2 Eingaben: – ein String – Wohnort – ein String – Versandart	Ein int – Versandgebühr	Die Methode berechnet die Versandgebühr und gibt sie zurück.

(15 Punkte)

Schreiben Sie die beschriebenen Methoden.

(Vorschlag: Sie können gerne kürzere Variablen-Namen benutzen, z.B. schreiben Sie wo als Wohnort.)

1. ruft 2. auf

2. ruft 3. auf

↳ recursion.

Zusammenfassung inkl. A16 (Coding)

1 - 5 : Aufgaben typen:

Aufgaben:	{	Hintergrund Prozess : II	(Speicher) Operationen
		Syntax Bezugenes : III	(Access/non-access) Code Inhalt (Syntax error)

6 - 10: Aufgaben typen:

multiple choice	{	(Bibliothek)
		(Wertzuweisung)
		(Serverdoc)
		(Random)
		(Syntax)

- 11 - 15 (Code Abschnitte) → Konsolen - Ausgabe

11 . Set , HashSet

12 . String editing methods

13 .

14 .

15 .

} methoden überladen

Probeklausur 1.

Große Struktur:

Aufgaben 1 bis 7 stellen Behauptungen dar. Kreuzen Sie jeweils an, ob die Behauptung richtig oder falsch ist.

1. (1 Punkt) String ist ein elementarer Datentyp.
 Richtig Falsch

2. (1 Punkt) Die folgende Methode hat keine Rückgabe:

```
public void myMethod(String s){
    // blah blah blah
}
```

Richtig Falsch

3. (1 Punkt) Kapselung bedeutet, eine Klasse gruppiert seine Daten und alle anderen Klassen dürfen die Attribute jederzeit zugreifen und verändern.

Richtig Falsch

4. (1 Punkt) Die folgenden Zeilen Code führen zu einer unendlichen Schleife:

```
for (int i = 0; i < Integer.MAX_VALUE; i++) {
    // blah blah blah
}
```

Richtig Falsch

5. (1 Punkt) Die for-Schleife und die while-Schleife drucken die gleichen Ausgaben in der Console:

```
for (int i = 0; i < 10; i = i + 2) {
    System.out.println(i);
}
int i = 0;
while (i < 10) {
    if (i % 2 == 0) {
        System.out.println(i);
    }
    i++;
}
```

Richtig Falsch

Für Aufgaben 6 und 7 benutzen wir die folgende Klasse:

```
public class Piratenschiff {
    protected final String rufname = "Black Pearl";
    private static String kapitaen = "Jack Sparrow";

    public Piratenschiff(String rufname, String kapitaen) {
        this.rufname = rufname;           // AAA
        this.kapitaen = "Johnny Depp";   // BBB
    }
}
```

6. (1 Punkt) Es gibt einen syntax error in Zeile AAA.

Richtig Falsch

7. (1 Punkt) Es gibt einen syntax error in Zeile BBB.

Richtig Falsch

— (OOP)

Access modifier
non-Access modifier
Daten-Kapselung

} Syntax error

(methoden)
C MAX_VALUE)
(Access/-non Access)
(Syntax-error)
(Datentypen) — Inhaltlich : |||
(Kapselung) & OOP

Vergleich: Probeklausur 2.

1-5 : Aufgaben typen:

Aufgaben tags:
Hintergrund Prozess : Syntax Bezugenes :

Aufgaben tags:

(Speicher) (Operationen)

(Access/non-Access) (Code Inhalt)
(Syntax error)

Aufgaben 8 bis 14 stellen Code-Abschnitte dar, die etwas in der Console drucken.

Schreiben Sie in den Antwort-Kästen, was in der Console gedruckt wird.

8. (1 Punkt)

```
int a = 10;  
if (a <= 10) {  
    System.out.println("eins");  
}  
else if (a >= 10) {  
    System.out.println("zwei");  
}  
else {  
    System.out.println("drei");  
}
```

'<=' hat eine höhere
(Priorität) als '>='

eins

9. (1 Punkt)

```
int a = 20;  
long b = 6;  
double c = Math.ceil(a / b);  
System.out.println(c);
```

(math library)

3.0

10. (1 Punkt)

```
// Der Zahl-Wert fuer das Euro-Zeichen ist 8364 im Dezimalsystem  
// Und 8364 im DezimalSystem ist 0010000010101100 im Binaersystem  
char x = '€';    (ASCII)  
char y = '\u20AC';    (ASCII)  
if (x == y) {  
    System.out.println("gleich");  
}  
else {  
    System.out.println("nicht gleich");  
}
```

gleich

11. (1 Punkt)

```
Set<Integer> set = new HashSet<Integer>();  
set.add(1); set.add(2); set.add(3);  
boolean b = true;  
String s = "123456";  
boolean kombi = !b && set.size() + 5 != 10 || Integer.parseInt(s.substring(3, 4)) >= 4;  
System.out.println(kombi);
```

true

(Set)
↳ HashSet

Methods:
· size()
· parseInt()
· substring()
equals .length from array
but for sets.

Für Aufgaben 12, 13 und 14 benutzen wir die folgenden Klassen:

```
-----  
public class Person{  
    protected String name = "";  
  
    public Person(String name) {  
        this.name = name;  
    }  
  
    public void etwasSagen() {  
        System.out.println("Person " + name);  
    }  
}  
  
-----  
public class Student extends Person{  
    public Student(String name) {  
        super(name);  
    }  
  
    public void etwasSagen() {  
        System.out.println("Student " + name);  
    }  
}  
  
-----  
public class Tutor extends Student{  
    public Tutor(String name) {  
        super(name);  
    }  
  
    public void etwasSagen() {  
        System.out.println("Tutor " + name);  
    }  
}
```

Access mod.
Setter - Element
Inheritance
Super

Jetzt führen wir ein paar Zeilen Code in einem MainProgramm aus. Was wird in der Console gedruckt?

12. (1 Punkt)

```
Tutor anna = new Tutor("Anna");  
if (anna instanceof Person) {  
    System.out.println("ja");  
}  
else {  
    System.out.println("nein");  
}
```

instanceof
↳ if instance or interface
↳ non-static value
outside of any
method.

ja

13. (1 Punkt)

```
Person bob = new Student("Bob");  
bob.etcasagen();
```

Student Bob

Aufgabe 14 ist auf
der nächsten Seite

14. (1 Punkt)

```
Tutor cassie = new Tutor("Cassie");  
Person cassie2 = (Person) cassie;  
Student cassie3 = (Student) cassie2;  
cassie3.etcasagen(); } ?
```

Tutor Cassie

Zusammenfassung:

Priorität

math library

Ascii

Set → HashSet

(Access-mod.,
this, Inheritance,
Super, instanceof)

Inheritance

Vergleich: Probeklausur 2.

- 11 - 15 (Code Abschnitte) → Konsolen - Ausgabe

11. Set, HashSet

12. String editing methods

13. } methoden überladen
14. }
15. }

overload

some OOP topic

Aufgaben 15 bis 20 bieten jeweils mehrere Antworten. Kreuzen Sie für jede Aufgabe nur eine Antwort an.

15. (1 Punkt) Wie viele Mal läuft die for-Schleife?

```
for (int i = 0; i < 5; i++) {  
    i = i + 1;  
}
```

L immer bei 0 anfangen zu zählen.

- 2 Mal 3 Mal 4 Mal 5 Mal

16. (1 Punkt) Wie viele Mal wird Hershey's Schoko :(in der Console gedruckt?

```
for (int i = 0; i < 5; i++) {  
    if (i < 3) {  
        continue;  
    }  
    System.out.println("Hershey's Schoko :(");  
}
```

- 2 Mal 3 Mal 4 Mal 5 Mal

17. (1 Punkt) Hier haben wir einen Screenshot aus einem Teil einer Javadoc:

Class Long

```
java.lang.Object  
java.lang.Number  
java.lang.Long
```

All Implemented Interfaces:

```
Serializable, Comparable<Long>
```

```
public final class Long  
extends Number  
implements Comparable<Long>
```

The Long class

Method Summary

Modifier and Type	Method and Description
static int	signum(long i) Returns the signum function of the specified long value.

Laut der Javadoc, was ist der Datentyp der Rückgabe für die signum() Methode?

- int Integer long Long keine Rückgabe

— wieder einfach
in Eclipse definitionen
schauen.

18. (1 Punkt) Was passiert, wenn wir die folgenden Zeilen Code ausführen?

```
String s = "We'll! Stay! Foreeeever this way! And my heart will (atmen!), go oooon...";  
String[] tokens = s.split("!");  
String erstesLeerzeichenWeg = tokens[2].contains("this").substring(1);  
System.out.println(erstesLeerzeichenWeg);
```

- this wird in der Console gedruckt
- Foreeeever this way wird in der Console gedruckt
- Foreeeever this way! wird in der Console gedruckt
- syntax error
- runtime error

[String methods]

— String editing

19. (1 Punkt) Was passiert, wenn wir die folgenden Zeilen Code ausführen?

```
Map<String, Integer> markenZuAnzahlen = new HashMap<String, Integer>();  
markenZuAnzahlen.put("Audi", 22);  
markenZuAnzahlen.put("BMW", 5);  
markenZuAnzahlen.put("Porsche", 1);  
markenZuAnzahlen.put("VW", 43);  
  
for (Entry<String, Integer> entry : markenZuAnzahlen.entrySet()) {  
    System.out.println(entry.getValue());  
}
```

- Die Automarken werden in der Console gedruckt
- Die Anzahlen von Autos werden in der Console gedruckt
- syntax error
- runtime error

HashTables are
datastructures

— Hashmap

Hashmaps use
datastructure internally

— Entry

↳ EntrySet

20. (1 Punkt) Wir haben eine kleine Klasse:

```
public class Hoersaal {  
    public int sitzplaetze = 100;  
  
    public void beschreiben() {  
        sitzplaetze = 50;  
        int sitzplaetze = 70;  
        System.out.println(sitzplaetze);  
    }  
}
```

Kann in methode neu definiert werden
klassisch
zuweisungen

Was passiert, wenn wir die folgenden Zeilen Code in einem MainProgramm ausführen?

```
Hoersaal raumB254 = new Hoersaal();  
raumB254.beschreiben();  
  
 50 wird in der Console gedruckt  
 70 wird in der Console gedruckt  
 100 wird in der Console gedruckt  
 syntax error  
 runtime error
```

Zusammenfassung:

multiple
choice

Schleifen

Javadoc

Stringmethoden

Hashmap

Entry - Set

Zuweisung

Syntax

multiple
choice

6 - 10: Aufgaben typen:

(Bibliothek)

(Wertzuweisung)

(Javadoc)

(Random)

(Syntax)

21. (6 Punkte) Wir haben ein **Array von Wörtern**, d.h. Datentyp des Arrays ist String. Wir interessieren uns für die Wörter, **die mit dem Zeichen "n" enden**, wobei es egal ist, ob großes "N" oder kleines "n". Wir möchten eine Methode schreiben, die die Positionen für solche Wörter in der Console drucken.

Z.B. haben wir das folgende Array:

0	1	2	3	4	5	6	7	8
Sonne	Mond	Stern	PARTYS	CHILLEN	TANZEN	Apfel	Birne	Sanddorn

Entsprechend ist die erwünschte Ausgabe in der Console:

2
4
5
8

Schreiben Sie unten die beschriebene Methode. Die Methode soll ein String-Array als Eingabe nehmen.