

Dr. Arne Schmidt

Klausur
Programmieren 1 (Probe)
01.01.2024

Name:

Vorname:

Matr.-Nr.:

Studiengang:

☐ Bachelor ☐ Master ☐ Andere

Klausurcode:

*Dieser wird benötigt, um das
Ergebnis der Klausur abzurufen.*

Hinweise:

- Bitte das Deckblatt in Druckschrift vollständig ausfüllen.
- Die Klausur besteht aus 18 Blättern, bitte auf Vollständigkeit überprüfen.
- Erlaubte Hilfsmittel: keine
- Eigenes Papier ist nicht erlaubt.
- Die Rückseiten der Blätter dürfen beschrieben werden.
- Antworten, die *nicht* gewertet werden sollen, bitte deutlich durchstreichen. Kein Tippex verwenden.
- Mit *Bleistift* oder in *Rot* geschriebene Klausurteile können nicht gewertet werden.
- Werden mehrere Antworten gegeben, werten wir die mit der geringsten Punktzahl.
- Die Bearbeitungszeit für die Klausur beträgt 120 Minuten.

DISCLAIMER Diese Probeklausur soll einen Überblick über mögliche Aufgaben und Fragen geben. Sie wurde nicht als 120-Minuten-Klausur konzipiert. Aufgaben in der Klausur können, aber müssen nicht den Aufgaben dieser Klausur ähneln. Eine Musterlösung wird es nicht geben.

Aufgabe	1	2	3	4	5	6	7	8	Σ
Punkte	?	?	?	?	?	?	?	?	100
Erreicht									

Aufgabe 1: Datentypen und Operatoren

(? Punkte)

- a) Betrachte folgende Operatoren und gib deren Rückgabetypp und Rückgabewert an. Nimm dazu an, dass für jede Teilaufgabe `int x = 23` gilt.

(Hinweis: Der Platz unter der Aufgabe kann für Nebenrechnungen genutzt werden.)

(i) `x >> 2` Rückgabetypp:_____ Rückgabewert:_____

(ii) `++x * 2` Rückgabetypp:_____ Rückgabewert:_____

(iii) `x > 5 ? "Ja" : "Nein"` Rückgabetypp:_____ Rückgabewert:_____

(iv) `x / 4.0` Rückgabetypp:_____ Rückgabewert:_____

(v) `x & 3` Rückgabetypp:_____ Rückgabewert:_____

(vi) `x + x / 2` Rückgabetypp:_____ Rückgabewert:_____

- b) Begründe kurz, warum zwei double-Werte nicht direkt auf Gleichheit verglichen werden sollten. Wie sollte man stattdessen vergleichen?

- c) Nenne zwei Möglichkeiten, wie man zwei Stringobjekte auf inhaltliche Gleichheit testet.

Aufgabe 2: Imperative Programmierung

(? Punkte)

a) Übersetze folgenden Algorithmus in ein Struktogramm:

Gegeben seien zwei Integer-Zahlen a und b . Solange a und b beide größer 0 sind:
Subtrahiere von a den Wert b , falls $a > b$, ansonsten subtrahiere von b den Wert a .
Gib am Ende a aus, falls $b = 0$, andernfalls gib b aus.

Struktogramm für den beschriebenen Text:

A large empty grid consisting of 20 columns and 30 rows, intended for drawing a flowchart (Struktogramm).

Aufgabe 3: Klassen und Modifikatoren

(? Punkte)

- a) Liste mindestens zwei Zugriffsmodifikatoren auf und erkläre kurz, inwieweit sie den Zugriff einschränken.
- b) Erkläre jeweils kurz, was der Modifikator `final` für Klassen, Methoden und Attribute bewirkt.
- c) Erkläre jeweils kurz, was der Modifikator `static` für Methoden und Attribute bewirkt.
- d) Welche Struktur (Deklaration, Attribute, Methoden) besitzt ein Interface?

- e) Betrachte folgenden Code-Abschnitt. Das Konzept der Vererbung in Java wurde an zwei Stellen falsch umgesetzt. Nenne beide Stellen und beschreibe den jeweiligen Fehler.

```
1  abstrakt class Shape{
2      ...
3  }
4
5  class Ellipse extends Shape{
6      ...
7  }
8
9  class Rectangle extends Shape{
10     ...
11 }
12
13 final class Triangle extends Shape{
14     ...
15 }
16
17 class RoundedSquare extends Ellipse, Rectangle{
18     ...
19 }
20
21 class AcuteTriangle extends Triangle{
22     ...
23 }
```

Aufgabe 4: Fehlersuche

(10 Punkte)

Betrachte den unten stehenden Java-Code, welcher aufgrund von mindestens 7 Fehlern nicht kompiliert. Identifiziere mindestens 7 Fehler, um die volle Punktzahl zu erhalten. Gib dazu die Zeilennummer, eine Begründung, warum diese Zeile einen Fehler wirft, sowie eine Korrektur der Zeile an. Jede Zeile besitzt maximal einen Fehler.

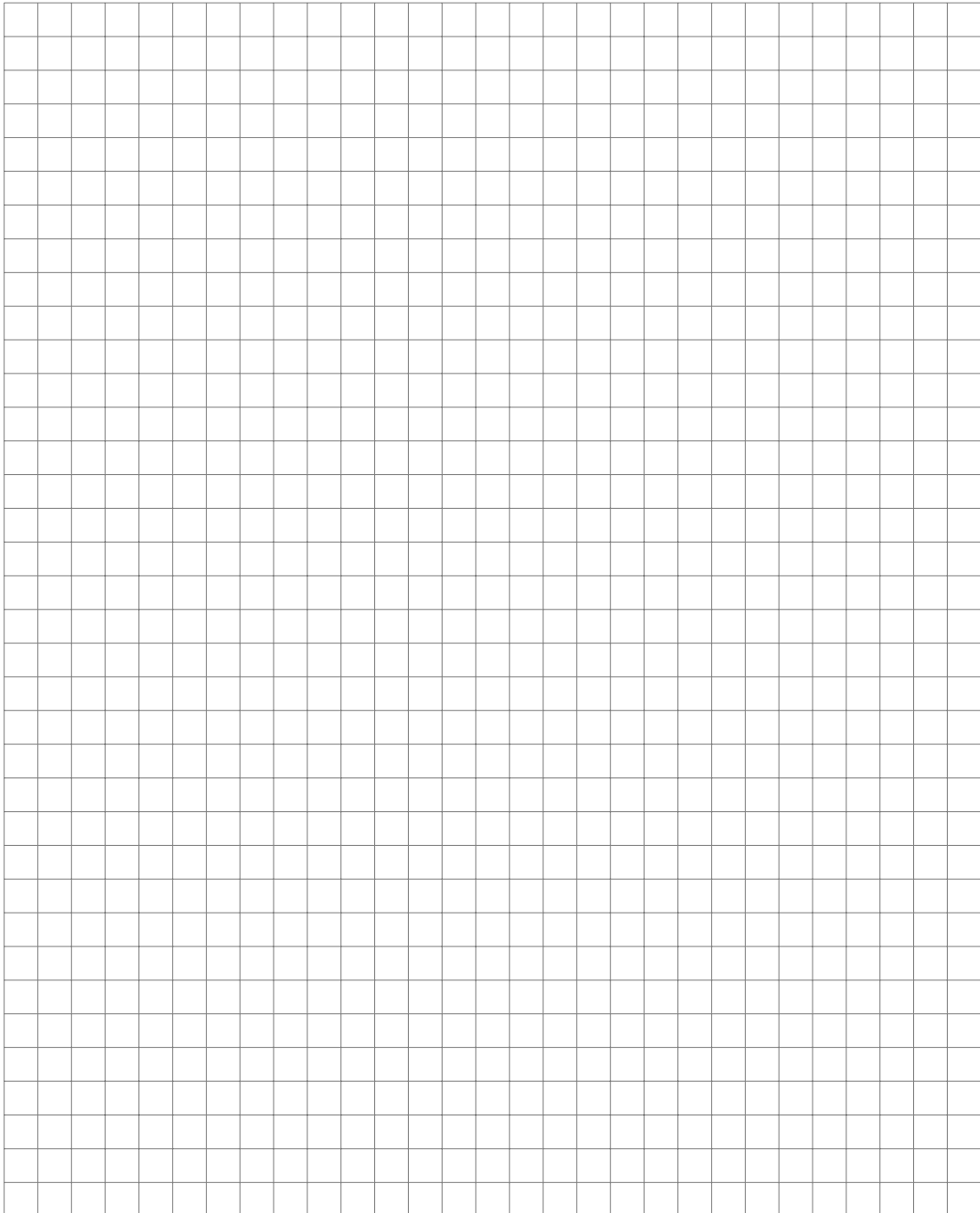
```
1 public class Number {
2     private int value;
3
4     public Number(value) {
5         this.value = value;
6     }
7
8     public boolean isPrime() {
9         if value <= 1 {
10             return false;
11         }
12         for (int i = 2; i < value; i++) {
13             if (value % i == 0) {
14                 return false
15             }
16         }
17         return true;
18     }
19
20     public static checkPrimeArray(Number[] numbers) {
21         for (Number num in numbers) {
22             if (!num.isPrime()) {
23                 return false;
24             }
25         }
26         return true;
27     }
28
29     public static void main(String[] args) {
30         Number num1 = new Number(5);
31         Number num2 = new Number(7);
32         Number num3 = new Number(11);
33
34         Number[] numbersArray = { num1, num2, num3 };
35
36         if (num1.checkPrimeArray(numbersArray)) {
37             System.out.println(All numbers are prime.);
38         }
39     }
40 }
```


Aufgabe 5: Rekursionen

(? Punkte)

Die Quersumme einer Zahl ist die Summe ihrer Ziffern. Wiederholt man diesen Vorgang auf dieser Summe solange, bis man eine einstellige Zahl erhält, so nennt man dies die iterative Quersumme. Bspw. ist die iterative Quersumme von 999 der Wert 9 ($999 \rightarrow 27 \rightarrow 9$).

- a) Implementiere eine rekursive Methode `iterativCrossSumRekursiv(int n)` in Form einer Endrekursion, um die iterative Quersumme einer Zahl n zu erhalten.



b) Entrekursiviere die in a) entworfene Methode auf geeignete Weise.



Aufgabe 6: Lückencode

(? Punkte)

Im nachfolgenden Code sind Lücken entstanden. Der Code soll für eine Auswahl von Rechtecken und Kreisen die Gesamtfläche bestimmen und ausgeben. Lese zunächst den lückenhaften Text einmal durch. Unter Umständen können Lücken erst durch Kontext gefüllt werden.

```
1 interface _____ {
2     double getArea();
3 }
4
5 class Rectangle _____ Shape {
6     private double length;
7     private double width;
8
9     public Rectangle(_____) {
10         this.length = length;
11         this.width = width;
12     }
13
14     @Override
15     public double getArea() {
16         return length * width;
17     }
18 }
19
20 class Circle _____ Shape {
21     private double radius;
22
23     public Circle(double radius) {
24         _____ = radius;
25     }
26
27     @Override
28     public double getArea() {
29         return Math.PI * radius * radius;
30     }
31 }
```

```

32
33 public class Main {
34     public static void main(String[] args) {
35         // Create an array of shapes with a mix of Rectangle and
           Circle
36         _____ shapes = {
37             new _____ (4, 5),
38             new _____ (3),
39             new _____ (5),
40             new _____ (2, 3)
41         };
42
43         double sumOfAreas = _____;
44
45         System.out.println("Sum of areas of all shapes: " +
           sumOfAreas);
46     }
47
48     private static double calculateSumOfAreas(Shape[] shapes) {
49         _____ sum = 0;
50         for (Shape shape : shapes) {
51             sum += _____;
52         }
53         return sum;
54     }
55 }

```

Aufgabe 7: Ausnahmebehandlung (? Punkte)

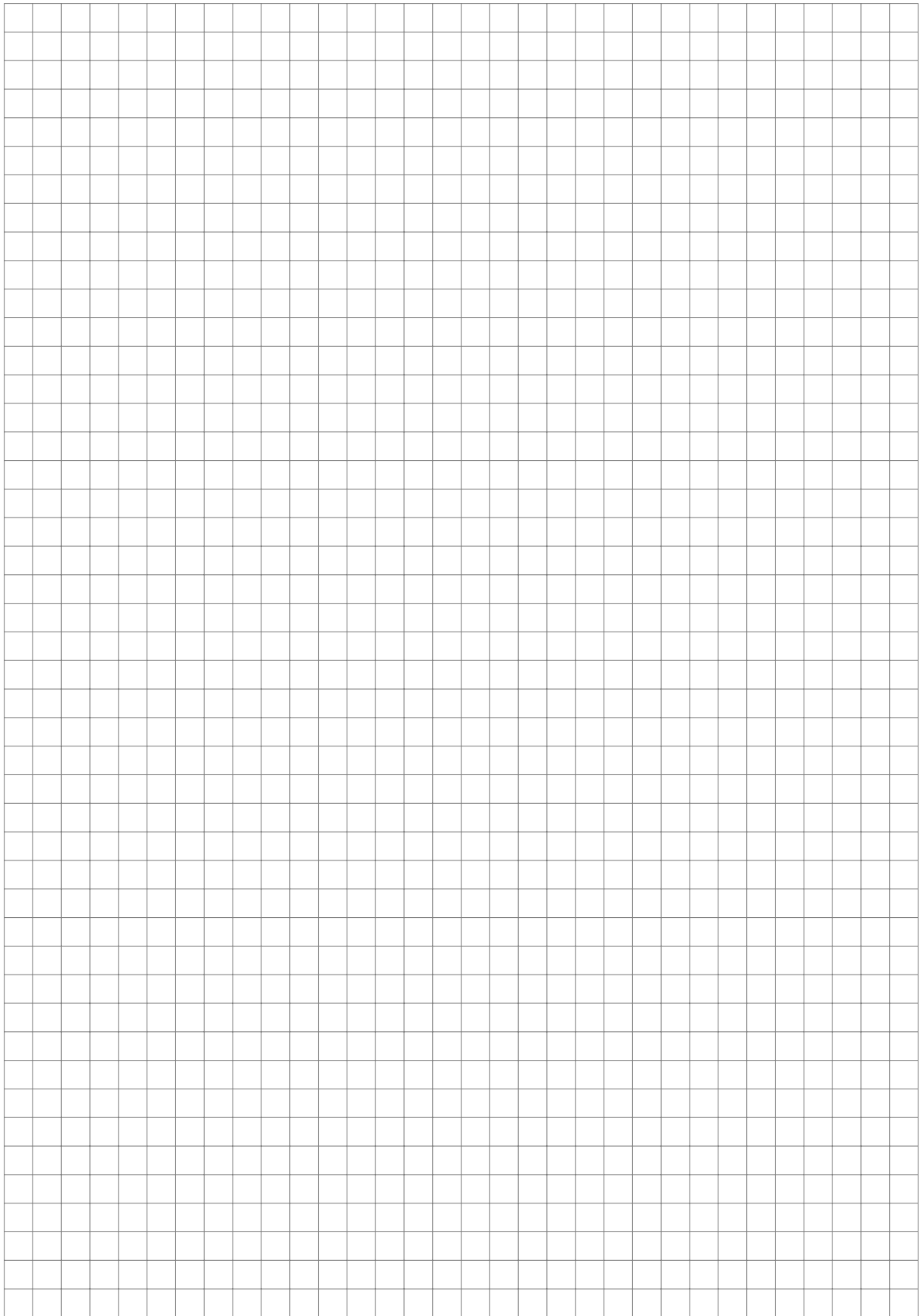
(? Punkte)

Wir möchten einen Temperaturumwandler schreiben, welcher Celsius in Fahrenheit umwandelt. Schreibe eine Methode `double convertTemperatur(double celsius)`, welche

- einen Wert in Celsius erhält (muss mind. -273.15 sein)
- einen Wert in Fahrenheit zurückgibt ($x \text{ Celsius entsprechen } (x * 9/5) + 32 \text{ Fahrenheit}$)
- Eine eigene Exception mit dem Name `InvalidTemperatureException` wirft, wenn der Eingabeparameter zu niedrig ist.

Schreibe zusätzlich eine main-Methode, welche einen über die Konsole eingegebenen Parameter in Fahrenheit umzuwandeln. Dabei soll eine Fehlermeldung ausgegeben werden, wenn die `InvalidTemperatureException` auftritt.

[illegible]



Aufgabe 8: Kurzfragen

(? Punkte)

- a) Erkläre kurz, in welchem Kontext `finally` benutzt wird und welche Auswirkungen dieses Schlüsselwort besitzt.
- b) Erkläre kurz den Begriff Polymorphismus.
- c) Wie zeichnet sich eine Utility Class aus?
- d) Was sind Wrapper-Klassen und wo werden sie eingesetzt?

- e) Erkläre kurz was ist ein Stack Overflow und wann dieser auftritt.

- f) Was bedeutet das Schlüsselwort super? Nenne ein Beispiel, wo dieses Schlüsselwort benutzt wird.

- g) Nenne jeweils ein existierendes Beispiel für eine checked und eine unchecked Exception.

- h) Erkläre was ein Seiteneffekt ist und warum diese auftreten können.

i) Was ist eine API?

j) Wie unterscheiden sich Java und Python bei der Zeichenfolge `//`?

k) Wie unterscheiden sich Java und Python bei ganzzahligen Werten?

l) Was tut die `assert`-Anweisung?

m) Mit welcher Anweisung kann auf der Konsole das Betriebssystem ausgegeben werden, auf dem das Java-Programm läuft?

n) Gib an, was die folgenden Anweisungen ausgeben. Beschreibe außerdem kurz, was %-3d, %1.2f und %n in der printf Methode bewirken.

```
1 int[] nums = {1, 20, 137, 5, 2, 13};  
2 for (int val : nums){  
3     System.out.printf("%-3d: %1.2f%n", val, 3*val/2.0);  
4 }
```

Viel Erfolg ☺