

1.

Welche der folgenden Aussagen bezüglich der Vererbung in Java trifft zu?

- ☐ Instanzvariablen in einer Oberklasse verdecken Variablen mit dem gleichen Namen in den davon abgeleiteten Unterklassen. ?
- ☒ Java erlaubt Mehrfachvererbung durch abstrakte Klassen.
- ☐ Klassenvariablen einer Oberklasse werden an alle abgeleiteten Unterklassen vererbt.
- ☒ Java erlaubt Mehrfachvererbung durch Interfaces

Die bestmögliche Lösung lautet::

Welche der folgenden Aussagen bezüglich der Vererbung in Java trifft zu?

- ☐ Instanzvariablen in einer Oberklasse verdecken Variablen mit dem gleichen Namen in den davon abgeleiteten Unterklassen.
- ☐ Java erlaubt Mehrfachvererbung durch abstrakte Klassen.
- ☒ Klassenvariablen einer Oberklasse werden an alle abgeleiteten Unterklassen vererbt.
- ☐ Java erlaubt Mehrfachvererbung durch Interfaces

?

2.

Der folgende Graph soll mit Hilfe von Breitensuche traversiert werden (Startknoten A). Welche Reihenfolge resultiert daraus? Ordnen Sie die Knoten entsprechend an.

```
graph TD; A((A)) --> E((E)); A --> F((F)); E --> D((D)); F --> C((C)); D --> G((G)); G --> B((B)); C --> B
```

A B C D E F G

A E F G D B C ✔

?

Die bestmögliche Lösung lautet::

A B C D E F G

?

3.

Welche Komplexität hat das Suchen in einem Binärbaum mit n Knoten?

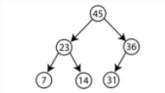
- ☒ $\mathcal{O}(\log(n))$
- ☐ $\mathcal{O}(n)$
- ☐ $\mathcal{O}(n \log(n))$
- ☐ $\Theta(\log(n))$
- ☐ $\Omega(n \log(n))$

Die bestmögliche Lösung lautet::

- ☐ $\mathcal{O}(\log(n))$
- ☒ $\mathcal{O}(n)$
- ☐ $\mathcal{O}(n \log(n))$
- ☐ $\Theta(\log(n))$
- ☐ $\Omega(n \log(n))$

4.

Aus folgender Halde werde das Element 45 entnommen. Wie lautet die Array-Implementierung der neuen vollständig sortierten Halde?



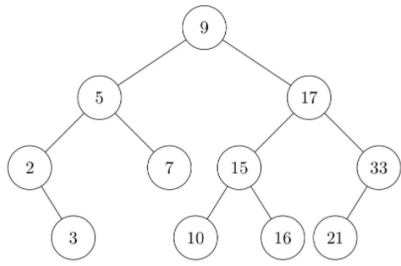
- ☐ 31, 23, 36, 7, 14
- ☐ 31, 23, 7, 14, 31
- ☐ 36, 23, 7, 14, 31
- ☒ 36, 23, 31, 7, 14

Die bestmögliche Lösung lautet::

- ☐ 31, 23, 36, 7, 14
- ☐ 31, 23, 7, 14, 31
- ☐ 36, 23, 7, 14, 31
- ☒ 36, 23, 31, 7, 14

5.

Bringen Sie die Zahlen in die Reihenfolge, die beim Inorder-Traversieren des Binärbaums entsteht!



2 3 5 7 9 10 15 16 17 21 33 ✓

Die bestmögliche Lösung lautet::

2 3 5 7 9 10 15 16 17 21 33

6.

Wieviele merge-Operationen sind notwendig, um eine Liste mit 17 Elementen mittels Merge sort (Sortieren durch Mischen) zu sortieren?

- ☐ 11
- ☐ 14
- ☒ 16
- ☐ 18
- ☐ 24

Die bestmögliche Lösung lautet::

- ☐ 11
- ☐ 14
- ☒ 16
- ☐ 18
- ☐ 24

7.

Welcher Größe entspricht die Summe aller Knotengrade in einem ungerichteten Graphen?





- ☐ der Kantenzahl
- ☐ der doppelten Kantenzahl
- ☐ der doppelten Knotenzahl
- ☒ der Summe aus Knoten- und Kantenzahl

Die bestmögliche Lösung lautet::

- ☐ der Kantenzahl
- ☒ der doppelten Kantenzahl
- ☐ der doppelten Knotenzahl
- ☐ der Summe aus Knoten- und Kantenzahl

8.

Wozu dient das Schlüsselwort **this** in Java?

- ☐ Zum Verdecken einer lokalen Variable gleichen Namens. 
- ☒ Zum Erreichen einer Instanzvariable mit dem selben Namen wie der einer lokalen Variable. 
- ☒ Zum Erreichen einer lokalen Variable. 
- ☐ Zum Verdecken einer Instanzvariable durch eine lokale Variable gleichen Namens. 

Die bestmögliche Lösung lautet::

- ☐ Zum Verdecken einer lokalen Variable gleichen Namens.
- ☒ Zum Erreichen einer Instanzvariable mit dem selben Namen wie der einer lokalen Variable.
- ☐ Zum Erreichen einer lokalen Variable.
- ☐ Zum Verdecken einer Instanzvariable durch eine lokale Variable gleichen Namens.


9.

Folgende Liste soll mit dem Quick-Sort-Algorithmus sortiert werden:

6 8 5 2 1 9 17 42 3 7

Das Pivot-Element soll hier immer das letzte Element der (Teil-)Liste sein.

Bringen Sie die Elemente in die Reihenfolge, in der sie nach dem **ersten** Sortierdurchgang (also kurz vor dem Suchen der 2. Pivotelemente) sind.





6 5 2 1 3 9 17 42 8 7 

Die bestmögliche Lösung lautet::

6 5 2 1 3 7 9 17 42 8

10.

Welche der folgenden Aussagen über die switch-Anweisung in Java ist korrekt?

- ☐ Der an die switch-Anweisung zu übergebende Ausdruck muss vom Typ boolean sein. 
- ☒ Es müssen mindestens zwei verschiedene case-Ziele definiert werden. 
- ☒ Die case-Ziele müssen vom Typ int sein. 
- ☐ Wenn der übergebene Ausdruck mit einem case-Ziel übereinstimmt, werden alle folgenden Anweisungen ausgeführt. 

Die bestmögliche Lösung lautet::

- ☐ Der an die switch-Anweisung zu übergebende Ausdruck muss vom Typ boolean sein.
- ☐ Es müssen mindestens zwei verschiedene case-Ziele definiert werden.
- ☐ Die case-Ziele müssen vom Typ int sein.
- ☒ Wenn der übergebene Ausdruck mit einem case-Ziel übereinstimmt, werden alle folgenden Anweisungen ausgeführt.

11.

Wie viele paarweise Vergleiche von Einträgen müssen Sie anstellen, um eine Liste von 7 Elementen mittels Bubble Sort (Sortieren durch Austauschen) zu sortieren?

- ☐ 14
- ☐ 18
- ☒ 21
- ☐ 24
- ☐ 27

Die bestmögliche Lösung lautet::

- ☐ 14
- ☐ 18
- ☒ 21
- ☐ 24
- ☐ 27

12.

Sie besitzen einen Limonadenstand und müssen täglich vielen Leuten Wechselgeld geben, welches Sie nur in Münzen aushändigen können. Da das Kopfrechnen lästig wird wollen Sie ein Programm schreiben, welches Ihnen automatisch sagt, wie viele von jeder Muenzen Sie zurück geben müssen. Sie wollen den Leuten immer möglichst wenig Münzen zurück geben.

Vervollständigen Sie die Methode **giveChange**, welche für jeden sichtbar sein soll und ein **int Array** zurück gibt. Letzteres gibt an, wie viel von welcher Münze gebraucht wird (2 Euro bis 1 Cent). Sie bekommen **amount** übergeben. Die Methode soll ohne eine Instanzvariable aufrufbar sein.

Vervollständigen Sie nun auch noch eine Methode **printChangeComposition** welche die Zusammensetzung der Münzen ausgibt. Die Methode soll öffentlich sichtbar sein, keinen Rückgabewert haben und ein **int Array** composition verarbeiten. Auch hier soll die Methode ohne eine Instanzvariable aufrufbar sein.

Die Ausgabe soll wie folgt aussehen: Change for customer: <amount> x 2 Euro, <amount> x 1 Euro, <amount> x 50 cent, <amount> x 20 cent, <amount> x 10 cent, <amount> x 5 cent, <amount> x 2 cent, <amount> x 1 cent.

Natürlich soll <amount> ersetzt werden.

ACHTUNG: Füllen Sie alle Lücken OHNE die Verwendung jeglicher Leerzeichen aus! Ansonsten wird Ihre Antwort durch die automatische Korrektur als falsch gewertet.

```
public class Limonade {

    private static int [] giveChange( double amount) {

        int [] coinValues = {200, 100, 50, 20, 10, 5, 2, 1};

        int [] changeComposition = new int[coinValues.length];

        int amountAsInt = (int)( amount *100);

        for(int i=0 ; i<coinValues.length ; i++ ) {

            int coin = coinValues[i];

            changeComposition[i] = amountAsInt/coin ;

            amountAsInt = amountAsInt%coin oder amountAsInt-changeComposition[i]*coin ;

        }

        return changeComposition ;

    }

}
```

```

private static void printChangeComposition(int [] composition) {

    int [] coinValues = {2, 1, 50, 20, 10, 5, 2, 1};

    System.out.print("Change for customer: ");

    for(int i=0 ; i<composition.length ; i++) {

        if(i < 2) {

            System.out.print(composition[i] + " x " + coinValues[i] + " Euro, ");

        } else if (i < composition.length - 1) {

            System.out.print(composition[i] + " x " + coinValues[i] + " Cent, ");

        } else {

            System.out.print(composition[i] + " x " + coinValues[i] + " Cent.");

        }

    }

}

public static void main(String[] args) {

    printChangeComposition(giveChange(7.99));





}

}

```

13.

Welche der folgenden Aussagen über Fehlerbehandlung trifft zu?

- ☒ Eine Ausnahme kann mit dem Befehl **catch** ausgeworfen werden. 
- ☐ Eine Ausnahme vom Typ **Error** muss nicht zwingend behandelt werden. 
- ☐ Alle Ausnahmen erben von der Klasse **Throwable** die Methode **time()**. 
- ☒ Eine Ausnahme vom Typ **RuntimeException** muss nicht zwingend behandelt werden. 

Die bestmögliche Lösung lautet::

- ☐ Eine Ausnahme kann mit dem Befehl **catch** ausgeworfen werden.
- ☒ Eine Ausnahme vom Typ **Error** muss nicht zwingend behandelt werden.
- ☐ Alle Ausnahmen erben von der Klasse **Throwable** die Methode **time()**.
- ☒ Eine Ausnahme vom Typ **RuntimeException** muss nicht zwingend behandelt werden.

14.

Gegeben sei ein Stack **s1**, der die Zahlen 0,1,2,3 enthält (0 ist die unterste Zahl).

Nun werden eine Queue **q** und ein Stack **s2** angelegt. Welche Zahlen enthält **s2** nach Ausführung der folgenden Anweisungen? (Geben Sie das Ergebnis angefangen vom untersten Wert in Stack **s2** bis hin zum obersten Wert an)

```
q.enqueue(s1.pop())
q.enqueue(s1.pop())
s2.push(q.dequeue())
q.enqueue(s1.pop())
s2.push(q.dequeue())
q.enqueue(s1.pop())
s2.push(q.dequeue())
s2.push(q.dequeue())
```

Achtung: Tippen Sie Ihr Ergebnis mit Kommas, aber OHNE jegliche Leerzeichen ein!

0123 ❌

Die bestmögliche Lösung lautet::

3,2,1,0

15.

Was besagt das FIFO Prinzip?

- ☒ Das Element, das zuletzt eingefügt wurde, wird auch zuletzt entnommen.
- ☐ Das Element, das zuerst eingefügt wurde, wird zuletzt entnommen.
- ☐ Je später ein Element eingefügt wurde, desto eher wird es entnommen.

Die bestmögliche Lösung lautet::

- ☒ Das Element, das zuletzt eingefügt wurde, wird auch zuletzt entnommen.
- ☐ Das Element, das zuerst eingefügt wurde, wird zuletzt entnommen.
- ☐ Je später ein Element eingefügt wurde, desto eher wird es entnommen.

16.

Welche Datenstruktur ist für die Sortierung von numerischen Elementen geeignet?

- ☒ Binärer Suchbaum ❌
- ☒ AVL-Baum ❌
- ☒ Halde ✔️
- ☐ Stapel ✔️

Die bestmögliche Lösung lautet::

- ☐ Binärer Suchbaum
- ☐ AVL-Baum
- ☒ Halde
- ☐ Stapel

17.

Welches der folgenden ist ein Typ von Rekursion?

- ☒ Kaskadenartige Rekursion ✓
- ☒ Lineare Rekursion ✓
- ☒ Geschachtelte Rekursion ✓
- ☒ Verschränkte Rekursion ✓

Die bestmögliche Lösung lautet::

- ☒ Kaskadenartige Rekursion
- ☒ Lineare Rekursion
- ☒ Geschachtelte Rekursion
- ☒ Verschränkte Rekursion

18.

Die Darstellung der Zahl z benötigt im Zahlensystem zur Basis B drei Stellen. Wie viele Multiplikationen und wie viele Additionen benötigt das Horner-Schema, um den Wert der Zahl z aus der Ziffernfolge zu berechnen?

- ☐ eine Multiplikation, zwei Additionen ✓
- ☐ zwei Multiplikationen, zwei Additionen
- ☒ zwei Multiplikationen, drei Additionen ✗
- ☐ drei Multiplikationen, drei Additionen ✗

Die bestmögliche Lösung lautet::

- ☐ eine Multiplikation, zwei Additionen
- ☒ zwei Multiplikationen, zwei Additionen
- ☐ zwei Multiplikationen, drei Additionen
- ☐ drei Multiplikationen, drei Additionen

19.

Welche Funktion wird mit der folgenden Methode `methode` für Werte $n \geq 1$ berechnet?

```
public long methode(int n, int m) {  
    if (n == 0) {  
        return 1;  
    } else {  
        return (n - 1) * methode(n - 1, m - 1);  
    }  
}
```

- ☐ 0
- ☒ $(n - 1)!$ ✗
- ☐ $(m - 1)!$
- ☐ $(n - 1)!(m - 1)!$

Die bestmögliche Lösung lautet::

- ☒ 0
- ☐ $(n - 1)!$
- ☐ $(m - 1)!$
- ☐ $(n - 1)!(m - 1)!$

20.

Frage (1 Punkt)

Sie haben die folgende Antwort gegeben:

Gegeben sei folgende Postfix-Notation:

4 3 + 2 * 6 + 2 /

Berechnen Sie das Ergebnis und geben Sie es in das Feld ein.

10 ✓

Die bestmögliche Lösung lautet::

Der Wert muss zwischen 10 und 10 liegen

21.

Seien a, b und c drei logische Elemente, die beide lediglich die Werte 0 und 1 annehmen können. Wie viele logische Verknüpfungen können mit diesen drei Elementen realisiert werden?

- ☐ 4 ✓
☒ 8
☒ 64
☒ 256

Die bestmögliche Lösung lautet::

- ☐ 4
☐ 8
☐ 64
☒ 256

22.

Gegeben ist der folgende Abstrakte Datentyp **Graph**. Er definiert eine Datenstruktur, in der die gerichteten Kanten eines Graphen zwischen Knoten aus der Menge V gespeichert werden können. (Die Knoten werden in diesem Fall nicht noch einmal gesondert gespeichert.)

ADT Graph
Sorten Graph, V , Boolean, Integer
Operationen
 create: \rightarrow Graph
 add: $\text{Graph} \times V \times V \rightarrow$ Graph
 f_1 : $\text{Graph} \rightarrow$ Integer
 f_2 : $\text{Graph} \times V \times V \rightarrow$ Boolean
 f_3 : $\text{Graph} \times V \times V \rightarrow$ Boolean
Axiome (siehe unten)

Die Operation create legt einen neuen Graphen ohne Kanten an, die Operation **add**(g, v_1, v_2) fügt dem Graphen g eine neue gerichtete Kante von v_1 nach v_2 hinzu.

Wählen Sie für die Operation f_1 (Axiome siehe unten) die korrekte Beschreibung der Wirkung der Operation!

$f_1 : \text{Graph} \rightarrow \text{Integer}$

$$f_1(\text{create}) = 0$$

$$f_1(\text{add}(g, v_1, v_2)) = 1 + f_1(g)$$

- Ⓒ $f_1(g)$ gibt die Anzahl der Kanten in g zurück.
☒ $f_1(g)$ gibt die Anzahl an Knoten in g zurück.
☐ $f_1(g)$ gibt die Position des Knotens v_1 in g zurück.
☐ $f_1(g)$ gibt die Position des Knotens v_2 in g zurück.

23.

Was ist der Wert der Variablen a und erg nach der Abarbeitung des folgenden Codes?

```
int a = -6;  
int erg = --a - a++;
```

- ☐ a = -7, erg = -14
- ☐ a = -6, erg = -1
- ☒ a = -6, erg = 0
- ☐ a = -6, erg = 1

24.

Rechnen Sie die angegebenen Dezimalzahlen in die geforderte Darstellung um.

-8 → ☒ Zweierkomplement, 5 bit

-42 → ☒ Einerkomplement, 8 bit

Die bestmögliche Lösung lautet::

-8 → ☒ Zweierkomplement, 5 bit

-42 → ☒ Einerkomplement, 8 bit

25.

Formulieren Sie die Postfix-Notation

als Infix-Notation.

Achtung: Tippen Sie das Ergebnis OHNE jegliche Leerzeichen ein!

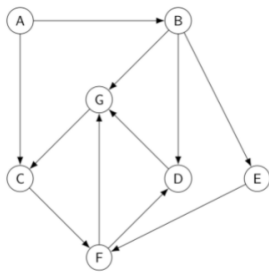
Infix-Notation: ☒

Die bestmögliche Lösung lautet::

Infix-Notation:

26.

Der folgende Graph soll mit Hilfe von Tiefensuche (Startknoten A) traversiert werden. Welche Reihenfolge resultiert daraus? Ordnen Sie die Knoten entsprechend an.



A B D E G C F ✖

Die bestmögliche Lösung lautet::

A B D G C F E

27.

Nehmen Sie an, dass Sie 16 Personen in einer Hashtabelle erfassen wollen, wobei jede Person unter ihrem Geburtstag abgelegt werden soll (365 mögliche Tage). Wieviele Einträge erwarten Sie in der Hashtabelle in etwa für jeden einzelnen Tag?

- ☐ 0.01
- ☒ 0.04
- ☐ 2.2
- ☐ 22

Die bestmögliche Lösung lautet::

- ☐ 0.01
- ☒ 0.04
- ☐ 2.2
- ☐ 22

28.

In welche Komplexitätsklasse gehört folgendes Programm?

```
n = 145;  
m = -56;  
for(i = 0; i <= n, i++) {  
    j = 0;  
    while(j < n * n) {  
        m -= i;  
        j++;  
    }  
}
```

- ☐ $O(m)$
- ☒ $O(n)$
- ☐ $O(n^2)$
- ☐ $O(n^3)$

Die bestmögliche Lösung lautet::

- ☐ $O(m)$
- ☐ $O(n)$
- ☐ $O(n^2)$
- ☒ $O(n^3)$

29.

Was ist die Ausgabe des folgenden Programms?

```
public class Ratio {  
    public static void main (String[] args) {  
        try {  
            System.out.println("ratio is:" + computeRatio(5.2, 0.0));  
        } catch (NumberFormatException e) {  
            System.err.println("Error:" + e.getMessage());  
        }  
        public static double computeRatio(double x, double y) {  
            return x/y;  
        }  
    }  
}
```

- ☐ ratio is:5.2
- ☐ ratio is:infinity
- ☐ Error: / by zero
- ☒ Exception in thread "main" java.lang.ArithmeticException: / by zero
at Ratio.computeRatio(Ratio.java:8)
at Ratio.computeRatio(Ratio.java:4)

30.

Die Java-Klasse **Gebäude** besitzt die Methode **eintreten**. Davon werden die Klassen **Bürogebäude** und **Wohnhaus** abgeleitet. Von der Klasse **Wohnhaus** wird eine weitere Klasse **Einfamilienhaus** abgeleitet, in welcher die Methode **eintreten** überschrieben wird. Ein Programmierer erzeugt dann ein neues Objekt der Klasse **Wohnhaus** mittels des Befehls

```
Wohnhaus wh = new Einfamilienhaus();
```

Welche Methode wird dann mittels des Befehls

```
wh.eintreten();
```

aufgerufen?

- ☐ Die Methode **eintreten** der Klasse **Gebäude**.
- ☒ Die Methode **eintreten** der Klasse **Wohnhaus**.
- ☐ Die Methode **eintreten** der Klasse **Einfamilienhaus**.

Die bestmögliche Lösung lautet::

- ☐ Die Methode **eintreten** der Klasse **Gebäude**.
- ☐ Die Methode **eintreten** der Klasse **Wohnhaus**.
- ☒ Die Methode **eintreten** der Klasse **Einfamilienhaus**.

31.

Zu welcher Komplexitätsklasse gehört die Funktion $2n + 3n^2 + 54n \log(n)$?

- ☐ $\mathcal{O}(2)$
- ☒ $\mathcal{O}(n \log(n))$
- ☐ $\Theta(n \log(n))$
- ☐ $\Theta(n^2)$

Die bestmögliche Lösung lautet::

- ☐ $\mathcal{O}(2)$
- ☐ $\mathcal{O}(n \log(n))$
- ☐ $\Theta(n \log(n))$
- ☒ $\Theta(n^2)$

32.

Welches Sortierverfahren ist auf verketteten Listen am schnellsten?

- ☐ Sortieren durch Auswählen (Selection Sort)
- ☐ Sortieren durch Einfügen (Insertion Sort)
- ☐ Sortieren durch Austauschen (Bubble Sort)
- ☐ Sortieren durch Mischen (Merge Sort)
- ☒ Sortieren durch Zerlegen (Quick Sort)
- ☐ BucketSort
- ☐ RadixSort

Die bestmögliche Lösung lautet::

- ☐ Sortieren durch Auswählen (Selection Sort)
- ☐ Sortieren durch Einfügen (Insertion Sort)
- ☐ Sortieren durch Austauschen (Bubble Sort)
- ☒ Sortieren durch Mischen (Merge Sort)
- ☐ Sortieren durch Zerlegen (Quick Sort)
- ☐ BucketSort

33.

Gegeben sind folgende rekursive Funktionen $g(n)$ und $u(n), n \geq 0$:

$$g(n) = \begin{cases} \text{true} & n = 0 \\ g(n/2) & n \text{ modulo } 2 = 0 \text{ und } n > 0 \\ u(n/2) & n \text{ modulo } 2 = 1 \end{cases} \quad u(n) = \begin{cases} \text{false} & n = 0 \\ u(n/2) & n \text{ modulo } 2 = 0 \text{ und } n > 0 \\ g(n/2) & n \text{ modulo } 2 = 1 \end{cases}$$

Beachten Sie, dass für die Implementierung der Funktion in Java für n nur primitive ganzzahlige Datentypen verwendet werden sollen.

ACHTUNG: Füllen Sie alle Lücken OHNE die Verwendung jeglicher Leerzeichen aus! Ansonsten wird Ihre Antwort durch die automatische Korrektur als falsch gewertet.

1. Um welchen Rekursionstyp handelt es sich?

verschränkt oder wechselseitig rekursiv.

2. Ergänzen Sie die folgende rekursive Implementierung dieser Funktionen in Java, die obige Funktionsdefinitionen direkt umsetzen soll.

```
public class Rekursion {

    public static void main (String[] args) {

        System.out.println(g(102));

    }

    public static boolean g( int n) {

        if ( n==0 ) {

            return true ;

        }

        if ( n%2==0 ) {

            return g(n/2) ;

        } else {

            return u(n/2);

        }

    }

}
```

```

public static boolean u( int n) {

    if ( n==0 ) {

        return false ;

    }

    if ( n%2==0 ) {

        return u(n/2) ;

    } else {

        return g(n/2);

    }

}

}

```

34.

Wieviele Knoten und wieviele Kanten hat die Expansion des folgenden Graphen im Knoten 1?



- ☒ 5 Knoten, 5 Kanten
- ☐ 5 Knoten, 6 Kanten
- ☐ 6 Knoten, 5 Kanten
- ☐ 8 Knoten, 7 Kanten

Die bestmögliche Lösung lautet::

- ☐ 5 Knoten, 5 Kanten
- ☐ 5 Knoten, 6 Kanten
- ☐ 6 Knoten, 5 Kanten
- ☒ 8 Knoten, 7 Kanten





35.

Welche der folgenden Variablendefinitionen produziert in Java eine Fehlermeldung?

- ☐ char zeichen = '1';
- ☒ double PI = 3.14;
- ☒ int number-2 = 10;
- ☒ final int N = 1000;

36.

Welche Aussagen zu Sortierverfahren für Eingaben mit n Elementen stimmen?




- ☒ Sortierverfahren heißen stabil, wenn ihre worst-case-Laufzeit in $O(n \cdot \log(n))$ ist. 
- ☐ Quick Sort kann nicht in-situ implementiert werden. 
- ☐ Die best-case- und die worst-case-Laufzeit von HeapSort sind in $O(n \cdot \log(n))$. 
- ☒ Für die gleiche Eingabe kann Bubble Sort schneller als Merge Sort sein. 

Die bestmögliche Lösung lautet::

- ☐ Sortierverfahren heißen stabil, wenn ihre worst-case-Laufzeit in $O(n \cdot \log(n))$ ist.
- ☐ Quick Sort kann nicht in-situ implementiert werden.
- ☒ Die best-case- und die worst-case-Laufzeit von HeapSort sind in $O(n \cdot \log(n))$.
- ☒ Für die gleiche Eingabe kann Bubble Sort schneller als Merge Sort sein.

37.

Wandeln Sie die Zahl 12.75 in eine binäre Gleitkommazahl nach dem IEEE-754-Standard (32 Bit) um. Die Nullen zum Auffüllen der Mantisse müssen nicht mit angegeben werden.

Vorzeichen: 
Exponent: 
Mantisse: 

Die bestmögliche Lösung lautet::





Vorzeichen:
Exponent:
Mantisse:

38.

Frage (1 Punkt)

Sie haben die folgende Antwort gegeben:

Was sind typische Methoden für Stapel?

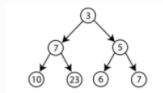
- ☐ put 
- ☒ pop 
- ☐ bottom 
- ☐ insert 

Die bestmögliche Lösung lautet::

- ☐ put
- ☒ pop
- ☐ bottom
- ☐ insert

39.

In welche Kategorie gehört der folgende Baum?



- ☐ Binärer Suchbaum
- ☒ AVL-Baum
- ☐ Halde

Die bestmögliche Lösung lautet::

- ☐ Binärer Suchbaum
- ☐ AVL-Baum
- ☒ Halde

40.

Welches Problem kann bei einer einfach verketteten Liste entstehen?

- ☐ Die Liste kann schwer von vorne nach hinten durchsucht werden.
- ☐ Beim Einfügen eines Elementes hinter einem bestimmten Element ist die Referenz des nachfolgenden Elementes schwierig zu bestimmen.
- ☒ Beim Einfügen eines Elementes vor einem bestimmten Element muss noch die Referenz des vorherigen Elementes gemerkt werden.

Die bestmögliche Lösung lautet::

- ☐ Die Liste kann schwer von vorne nach hinten durchsucht werden.
- ☐ Beim Einfügen eines Elementes hinter einem bestimmten Element ist die Referenz des nachfolgenden Elementes schwierig zu bestimmen.
- ☒ Beim Einfügen eines Elementes vor einem bestimmten Element muss noch die Referenz des vorherigen Elementes gemerkt werden.

41.

Markieren Sie die Codeausschnitte, in denen die Methode `sum` korrekt überladen wurde.

```
public static double sum (int i, int d) {  
    return (double) (i + d);  
}
```

```
public static double sum (int d, int i) {  
    return (double) (d + i);  
}
```

Ausschnitt (a)

```
public static int sum (int i, int d) {  
    return (i + d);  
}
```

```
public static double sum (int d, int i) {  
    return (double) (d + i);  
}
```

Ausschnitt (b)

```
public static double sum (int i, double d) {  
    return (double) (i + d);  
}
```

```
public static double sum (double d, int i) {  
    return (double) (d + i);  
}
```

Ausschnitt (c)

- ☐ Ausschnitt (a) ✓
- ☐ Ausschnitt (b) ✓
- ☒ Ausschnitt (c) ✓

42.

Bringen Sie die Zahlen in die Reihenfolge, die beim Preorder-Traversieren des Binärbaums entsteht!

9 5 2 3 7 17 15 10 16 33 21 ✓

43.

Welche der folgenden Datentypen werden in Java durch eine Referenz auf das entsprechende Objekt abgespeichert?

- ☒ eindimensionale Felder ✓
- ☒ zweidimensionale Felder ✓
- ☐ Strings ✗
- ☐ Integer Variablen ✗

Die bestmögliche Lösung lautet::

- ☒ eindimensionale Felder
- ☒ zweidimensionale Felder
- ☒ Strings
- ☒ Integer Variablen

44.

Welchen Wert haben die Variablen x und y nach der Abarbeitung des folgenden Java- Codes?

```
public static void setze_null(int a, int[] b) {
    a = 0;
    for (int i = 0; i < b.length; i++) {
        b[i] = 0;
    }
}

public static void main(String[] args) {
    int x = 6;
    int y[] = {1,2,3,4,5};
    setze_null(x, y);
}
```

☒ x=0, y={0,0,0,0,0} ✗
☐ x=0, y={1,2,3,4,5}
☐ x=6, y={0,0,0,0,0}
☐ x=6, y={1,2,3,4,5}

Die bestmögliche Lösung lautet::

☐ x=0, y={0,0,0,0,0}
☐ x=0, y={1,2,3,4,5}
☒ x=6, y={0,0,0,0,0}

45.

Was sind typische Methoden für verkettete Listen?

- ☒ **get** ✓
- ☐ **insert** ✗
- ☐ **delete** ✗
- ☐ **advance** ✗

Die bestmögliche Lösung lautet::

- ☒ **get**
- ☒ **insert**
- ☒ **delete**
- ☒ **advance**

46.

Gegeben sei eine leere, geschlossene Hashtabelle mit der Behältergröße $b = 1$ und der Behälteranzahl $k = 8$. Die verwendete Hashfunktion ist $h(x) = (x * 2 - 3) \bmod k$ und Kollisionen sollen durch lineares Sondieren in beide Richtungen mit Schrittweite $c = 2$ aufgelöst werden. Fügen Sie der Reihe nach die Elemente 93, 68, 41, 71, 78, 16, 25, 3 in die Hashtabelle ein!

0	1	2	3	4	5	6	7
✗	41 ✗	✗	✗	71 ✓	✗	68 ✓	✗

Die bestmögliche Lösung lautet::

0	1	2	3	4	5	6	7
41	78	25	71	3	68	16	93

47.

In welchem der folgenden Programmabschnitte wird eine anonyme Klasse definiert?

- ☐ `anonymous class AC{private int i = 1};`
- ☒ `public anonymous class AC{public int i = 1};`
- ☐ `return new AC(){public int i = 1};`
- ☐ `new AC{private int i = 1};`

Die bestmögliche Lösung lautet::

- ☐ `anonymous class AC{private int i = 1};`
- ☐ `public anonymous class AC{public int i = 1};`
- ☒ `return new AC(){public int i = 1};`
- ☐ `new AC{private int i = 1};`

48.

Implementieren Sie eine Warteschlange (Queue), die zur Speicherung der Daten ein Array fester Größe verwendet. Diese Struktur wird auch Ringpuffer genannt, da das Array wie ein Ring behandelt wird: sobald man mit dem Einfügen neuer Elemente am Ende des Arrays ankommt, fängt man wieder am Anfang an.



Ergänzen Sie den folgenden Programmcode so, dass die Methode `enqueue` neue Elemente zur Liste einfügt. Die Methode soll `true` zurückgeben, wenn das neue Element erfolgreich eingefügt wurde und `false`, falls der Puffer bereits voll ist.

ACHTUNG: Ergänzen Sie alle Lücken OHNE die Verwendung jeglicher Leerzeichen. Ansonsten wird Ihre Antwort durch die automatische Korrektur als falsch bewertet.

```
public class Queue {

    private final int CAPACITY;

    private int[] buffer;

    private int first = 0;

    private int last = 0;

    private int size = 0;

    public Queue(int capacity) {

        CAPACITY = capacity;

        buffer = new int[CAPACITY];

    }

    public void clear() {

        last = first;

        size = 0;

    }

    public boolean isEmpty() {

        return size == 0;

    }

    public int size() {
```

```
        public boolean isEmpty() {

            return size == 0;

        }

        public int size() {

            return size;

        }

        public boolean enqueue(int element) {

            if (size==CAPACITY) {

                return false ;

            }

            buffer[first++] = element ;

            first %= CAPACITY ;

            size++;

            return true ;

        }

    }
```

49.

Was sind Beispiele für eine Ereignisquelle in Java?

- ☒ Mausclick
- ☐ Mausbewegung
- ☒ Tastaturklick
- ☐ Textfeld

Die bestmögliche Lösung lautet::

- ☐ Mausclick
- ☐ Mausbewegung
- ☐ Tastaturklick
- ☒ Textfeld

50.

Welche der folgenden Aussagen zur Polymorphie ist/sind korrekt?

- ☐ Die folgende Deklaration mit Initialisierung ist in Java erlaubt: `Object x = 42;`
- ☐ Eine als `final` deklarierte Methode kann auch aus einer Instanzmethode heraus aufgerufen werden, sofern sie die dafür notwendige Sichtbarkeit hat.
- ☒ Im Falle überladener Methoden erfolgt die Auswahl der richtigen Methode nur zur Laufzeit und nur über die Datentypen der Argumente und den dynamischen Typ des Zielobjekts.
- ☐ In Java dürfen die beiden Methoden `long foo(int x)` und `int foo(long x)` nicht in der gleichen Klasse deklariert werden.

Die bestmögliche Lösung lautet::

- ☒ Die folgende Deklaration mit Initialisierung ist in Java erlaubt: `Object x = 42;`
- ☒ Eine als `final` deklarierte Methode kann auch aus einer Instanzmethode heraus aufgerufen werden, sofern sie die dafür notwendige Sichtbarkeit hat.
- ☐ Im Falle überladener Methoden erfolgt die Auswahl der richtigen Methode nur zur Laufzeit und nur über die Datentypen der Argumente und den dynamischen Typ des Zielobjekts.
- ☐ In Java dürfen die beiden Methoden `long foo(int x)` und `int foo(long x)` nicht in der gleichen Klasse deklariert werden.

51.

Was sind Grundprinzipien der objektorientierten Programmierung?

- ☒ Vererbung
- ☒ Kapselung
- ☒ Polymorphie
- ☒ Abstraktion

52.

Objekte welcher Java-Klasse bzw. welcher Java-Datentyps können in einen `DataOutputStream` geschrieben werden?

- ☒ String ✔
- ☐ Integer ✘
- ☐ double ✘
- ☒ char ✔

Die bestmögliche Lösung lautet::

- ☒ String
- ☒ Integer
- ☒ double
- ☒ char

53.

Wann entsteht das Diamond-Problem?

- ☒ Wenn eine Klasse von zwei Klassen erbt, die wiederum von derselben Oberklasse abgeleitet sind. ✔
- ☐ Wenn zwei Klassen von derselben Oberklasse abgeleitet sind, die wiederum von zwei Oberklassen erbt.
- ☐ Wenn bei der Mehrfachvererbung eine Klasse von mehreren Oberklassen erbt.
- ☐ Wenn von einer Klasse mehr als eine Unterklasse gebildet wird.

Die bestmögliche Lösung lautet::

- ☒ Wenn eine Klasse von zwei Klassen erbt, die wiederum von derselben Oberklasse abgeleitet sind.
- ☐ Wenn zwei Klassen von derselben Oberklasse abgeleitet sind, die wiederum von zwei Oberklassen erbt.
- ☐ Wenn bei der Mehrfachvererbung eine Klasse von mehreren Oberklassen erbt.
- ☐ Wenn von einer Klasse mehr als eine Unterklasse gebildet wird.

54.

Welche Komplexität hat das Suchen in einem AVL-Baum mit n Knoten?

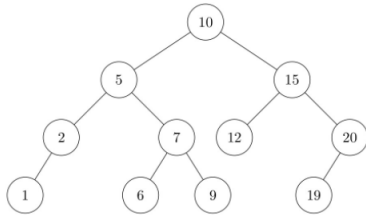
- ☒ $O(\log(n))$ ✔
- ☐ $O(n)$
- ☐ $O(n \log(n))$
- ☐ $\Theta(\log(n))$
- ☐ $\Omega(n \log(n))$

Die bestmögliche Lösung lautet::

- ☒ $O(\log(n))$
- ☐ $O(n)$
- ☐ $O(n \log(n))$
- ☐ $\Theta(\log(n))$
- ☐ $\Omega(n \log(n))$

55.

Bringen Sie die Zahlen in die Reihenfolge, die beim Postorder-Traversieren des Binärbaums entsteht!



1 2 6 9 7 5 12 19 20 15 10 ✓

56.

Ordnen Sie folgende Funktionen **aufsteigend** nach ihrer kleinsten oberen Schranke im \mathcal{O} -Kalkül!

$$f_1(n) = 2n + n; f_2(n) = \log \frac{n^2}{n}; f_3(n) = 9 \frac{1}{2}n; f_4(n) = \pi^n; f_5(n) = \sin(n) \cdot 2^n; f_6(n) = 3 \sin n$$

f1 f6 f2 f3 f5 f4 ✗

Die bestmögliche Lösung lautet::

f6 f2 f1 f5 f3 f4

57.

Sie haben eine Java Klasse **Fahren** geschrieben, welche die main-Funktion beinhaltet. Außerdem ruft diese Klasse die Klasse **Fahrzeug** auf, welche wiederum die Unterklasse **Auto** aufruft. Alle drei Klassen sind in Dateien mit dem entsprechenden Namen gespeichert. Welchen Kommandozeilen-Befehl können Sie benutzen, um das gesamte Programm zu kompilieren?

- ☐ javac Fahren.java
- ☐ javac Fahrzeug.java
- ☒ javac Auto.java ✗
- ☐ javac Fahrzeug.java Auto.java

Die bestmögliche Lösung lautet::

- ☒ javac Fahren.java
- ☐ javac Fahrzeug.java
- ☐ javac Auto.java
- ☐ javac Fahrzeug.java Auto.java

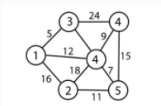
58.

In einem Postzentrum werden die eingegangenen Briefe zunächst nach der ersten Ziffer der Postleitzahl auf verschiedene Behälter verteilt. Welcher Sortieralgorithmus implementiert dieses Vorgehen als ersten Schritt?

- ☐ Sortieren durch Auswählen (Selection Sort)
- ☐ Sortieren durch Einfügen (Insertion Sort)
- ☐ Sortieren durch Austauschen (Bubble Sort)
- ☐ Sortieren durch Mischen (Merge Sort)
- ☐ Sortieren durch Zerlegen (Quick Sort)
- ☐ BucketSort
- ☒ RadixSort

59.

Wie viele Kanten hat der minimale Spannbaum des folgenden Graphen?



- ☒ 4 Kanten
- ☐ 5 Kanten
- ☐ 6 Kanten
- ☐ 7 Kanten

Die bestmögliche Lösung lautet::

- ☐ 4 Kanten
- ☒ 5 Kanten
- ☐ 6 Kanten
- ☐ 7 Kanten

60.

Welches sind Merkmale von Java?

- ☒ ist objektorientiert
- ☐ besitzt vergleichbare Syntax zu C und C++
- ☒ erlaubt Mehrfachvererbung
- ☐ enthält Zeigerarithmetik

Die bestmögliche Lösung lautet::

- ☒ ist objektorientiert
- ☒ besitzt vergleichbare Syntax zu C und C++
- ☒ erlaubt Mehrfachvererbung
- ☒ enthält Zeigerarithmetik