

Beantworten Sie die Fragen in den Aufgaben 1 und 2 mit einer kurzen, prägnanten Antwort.

Aufgabe 1 (8 Punkte)

1. Wie sieht -13 in der 8Bit 2-er Komplementdarstellung aus?

2. Woraus ergibt sich die *Totale Korrektheit* eines Programms?

3. Wie lauten die Bestandteile eines endlichen Automaten?

4. Woraus besteht ein Graph?

5. Welche Operationen stellt der ADT Liste zur Verfügung?

6. Wie ist die Laufzeit (O-Notation) der binären Suche im *Worst Case* abhängig von der Array-Länge n ?

7. Wie lauten die ersten sieben Zahlen der Fibonacci-Folge?

8. Welcher Traversierung entspricht die Tiefensuche im ADT *Baum*?

Aufgabe 2 (8 Punkte)

Alle Fragen beziehen sich auf die Programmiersprache Java 5.

1. Nach welchem Prinzip werden Parameter bei einem Methodenaufruf übergeben?

2. Die Java-Klasse `Buecherei` enthalte eine Methode `public static int getAnzahlBuecher()`. Mit welchem Ausdruck können Sie die Methode aus einer anderen Klasse aufrufen, ohne eine Instanz dieser Klasse zu verwenden?

3. Mit welchem Ausdruck erzeugen Sie in Java eine typische Liste für `Strings` ?

4. Was kennzeichnet das Schlüsselwort `static` in Java?

5. Wie nennt man Methoden, die in einer abgeleiteten Klasse denselben Namen wie in der Superklasse haben, aber unterschiedlich viele Parameter besitzen?

6. Wie lautet der Code, mit dem Sie eine `RuntimeException` mit der Meldung `Fehler!` werfen?

7. Wie nennt man das Ermitteln der aufzurufenden Methode zur Laufzeit bei Instanzmethoden?

8. Gegeben ist folgender Ausdruck: `boolean b = false==false ? 2 > 1 : 1 > 2;`
Welchen Wert hat `b`?

Beantworten Sie die Aufgaben 3 und 4, indem Sie unten genau ein Kreuz machen. Ist Ihre Antwort richtig, erhalten Sie alle Punkte. Ist sie falsch, erhalten Sie keinen Punkt.

Aufgabe 3 (6 Punkte)

Gegeben sei folgende Java-Klasse:

```
import AlgoTools.IO;

public class Mitarbeiter{

    public Mitarbeiter chef;
    public Mitarbeiter untergebener;
    public Mitarbeiter kollege;
    public String name;

    public void baueMitarbeiter(Mitarbeiter chef, Mitarbeiter untergebener,
                                Mitarbeiter kollege, String name){

        this.chef = chef;
        this.untergebener = untergebener;
        this.kollege = kollege;
        this.name = name;
    }

    public static void main(String[] args){

        Mitarbeiter oliver = new Mitarbeiter();
        Mitarbeiter ralf = new Mitarbeiter();
        Mitarbeiter patrick = new Mitarbeiter();
        Mitarbeiter doro = new Mitarbeiter();

        oliver.baueMitarbeiter(ralf, patrick, doro, "Ralf");
        ralf.baueMitarbeiter(oliver, patrick, doro, "Doro");
        patrick.baueMitarbeiter(oliver, doro, ralf, "Oliver");
        doro.baueMitarbeiter(ralf, patrick, oliver, "Patrick");

        ralf.untergebener = oliver.kollege;
        ralf.name = doro.untergebener.name;
        patrick.chef = doro.untergebener;

        IO.println(ralf.untergebener.untergebener.chef.kollege.name);
    }
}
```

Welche Ausgabe hat die Methode main dieser Klasse?

- Doro Oliver Patrick Ralf

Aufgabe 4 (6 Punkte)

Gegeben sei folgende Java-Klasse:

```
import AlgoTools.IO;
public class Aehnlich {

    private int a,b;

    public Aehnlich(int a, int b){
        this.a = a;
        this.b = b;
    }
    public void aehnlich(int a){
        this.b += a;
        aehnlich();
    }
    public void aehnlich(int a, int b){
        this.a += b;
        this.b += a;
        aehnlich(this.b);
    }
    public void aehnlich(){
        a *= 2;
        b *= 2;
    }
    public static void main(String[] args) {

        Aehnlich aehnlich = new Aehnlich(5,1);
        Aehnlich aehnlicher = new Aehnlich(4,2);

        aehnlich.ahnlich(aehnlich.a,aehnlich.b);
        aehnlicher.ahnlich(aehnlicher.b);

        IO.println(aehnlich.a + aehnlich.b + aehnlicher.a + aehnlicher.b);
    }
}
```

Welche Ausgabe hat die Methode main dieser Klasse?

- | | | | |
|--------------------------|--------------------------|--------------------------|--------------------------|
| <input type="radio"/> 22 | <input type="radio"/> 23 | <input type="radio"/> 32 | <input type="radio"/> 40 |
| <input type="radio"/> 42 | <input type="radio"/> 52 | <input type="radio"/> 56 | <input type="radio"/> 64 |

Tragen Sie bei den Aufgaben 5 bis 13 Ihre Lösungen in den vorgesehenen Platz ein.

Aufgabe 5 (6 Punkte)

Gegeben sei folgende Java-Klasse:

```
import AlgoTools.IO;

public class Raetsel {

    public static int frage(int a, int b) {

        if(b == 0) return a;

        return frage(b, a - b * (a/b));
    }
}
```

a) Was berechnet die Methode `public static int frage(int a, int b)` der Klasse `Raetsel` für $a, b \geq 1$? (4 Punkte)

b) Wie ist die Laufzeit der Methode `public static int frage(int a, int b)` in der O-Notation, wenn $a = b$? (2 Punkte)

Aufgabe 6 (7 Punkte)

Gegeben sei folgende Java-Klasse:

```
public class Fraglich {
    public static int methode (int n){
        int k = 1;
        int i = 0;
        int j= n+1;

        while(++i < --j){
            k *= i*j;
        }

        if(n % 2 != 0) k *= (n + 1) / 2;

        return k;
    }
}
```

a) Was berechnet die Methode `public static int methode(int n)` der Klasse `Fraglich`? (5 Punkte)

b) Wie ist die Laufzeit der Methode `public static int methode(int a)` in der O-Notation? (2 Punkte)

Aufgabe 7 (8 Punkte)

Sie passen einen Tag auf den Hund eines Bekannten auf, dabei müssen Sie einige Regeln beachten. Sie müssen mit dem Hund Gassi gehen, ihn füttern und können mit ihm spielen. Damit Sie wissen, ob Sie alles richtig machen, sollen Sie einen Automaten erstellen, der den korrekten Tagesablauf überprüft.

- Sie müssen mindestens zweimal Gassi gehen.
- Sie müssen den Hund genau einmal füttern.
- Bevor Sie den Hund füttern, müssen sie mindestens einmal Gassi gegangen sein.
- Sie können mit dem Hund zu jeder Zeit und beliebig oft spielen.

a) Geben Sie das Eingabealphabet und die Zustandsmenge an. (2 Punkte)

Eingabealphabet:

Zustandsmenge:

b) Zeichnen Sie den Zustandsüberföhrungsgraphen des Automaten, der die Gültigkeit des Tagesablaufes überprüft. Markieren Sie den Start- und den oder die korrekten Endzustände. (6 Punkte)

Aufgabe 8 (5 Punkte)

gegeben sei folgendes char-Array:

Z E U G

Sortieren Sie das char-Array mit dem Heapsort-Verfahren aus der Vorlesung lexikographisch.

a) Geben Sie für die erste Phase den Baum sowie den daraus erzeugten Heap an. (2 Punkte)

Ursprungsbaum:

Heap:

b) Für Phase 2 geben Sie jeweils den Baum nach Entfernen der Wurzel, den reorganisierten Heap, sowie das zugehörige aktuelle char-Array für jeden Sortierschritt nach dem Reorganisieren an. (3 Punkte)

Aufgabe 10 (6 Punkte)

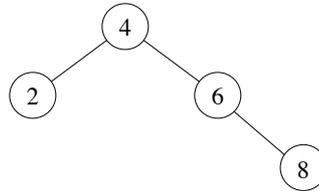
Gegeben sei folgende Postorder-Traversierung eines binären Suchbaums:

c r o t u s m

- a) Zeichnen Sie den Suchbaum. (4 Punkte)
- b) Löschen Sie den Buchstaben *s* aus dem Baum und zeichnen Sie ihn erneut. (2 Punkte)

Aufgabe 11 (6 Punkte)

Gegeben sei folgender AVL-Baum:



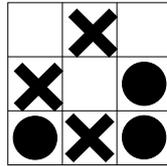
a) Fügen Sie in den Baum die Zahl 9 ein, indem Sie den Knoten direkt in den obigen AVL-Baum einzeichnen. Versuchen Sie den Baum mit Balancen. (1 Punkt)

b) Führen Sie eine eventuell notwendige Rotation aus und zeichnen Sie den reorganisierten Baum erneut. Vermerken Sie, welche Rotation Sie verwenden haben. (2 Punkte)

c) Fügen Sie in den neu gezeichneten Baum die Zahl 5 ein und versuchen Sie den Baum mit Balancen. Führen Sie die notwendige Rotation aus und zeichnen sie den Baum unten nochmals. Versuchen Sie den Baum wieder mit Balancen und vermerken Sie die Art der Rotation (3 Punkte).

Aufgabe 12 (7 Punkte)

Betrachten Sie folgende Situation eines TicTacToe-Spiels auf einem 3×3 -Feld:



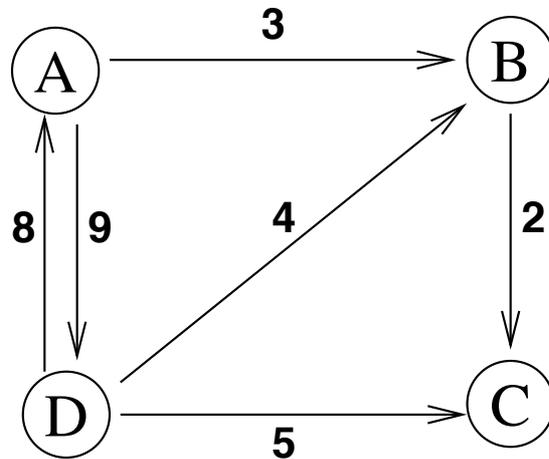
Ihre Spielzüge sind mit \times , die Ihres Gegners mit einem Kreis eingezeichnet. Ihr Gegner ist am Zug.

a) Zeichnen Sie einen Spielbaum für den restlichen Verlauf dieses TicTacToe-Spiels. Bewerten Sie die möglichen Spielzüge und schreiben Sie die entsprechenden *minmax*-Werte für jeden Spielzug auf. Verwenden Sie die Werte -1 (Kreis/Gegner gewinnt), 0 (unentschieden) und $+1$ (Kreuz/man selbst gewinnt). (6 Punkte)

b) Wenn sowohl ihr Gegner als auch Sie geschickt spielen, wer wird gewinnen? (1 Punkte)

Aufgabe 13 (4 Punkte)

Sei folgender gerichteter Graph mit gewichteten Kanten gegeben:



Geben Sie die Adjazenzmatrix zum Graphen an:

	A	B	C	D
A				
B				
C				
D				

Bei den Aufgaben 14 bis 16 müssen Sie selbst einige Zeilen Java-Code schreiben. Nutzen Sie den vorgesehenen Platz zwischen den geschweiften Klammern.

Aufgabe 14 (5 Punkte)

Ergänzen Sie in der gegebenen Klasse `BaumTools` die Methode `printEbene`, die eine bestimmte Ebene des Baumes von links nach rechts ausgibt. Achten Sie auf eine geeignete Fehlerbehandlung. Hinweis: Verwenden Sie Rekursion!

```
public class BaumTools {  
    public static void printEbene(Baum b, int e){
```

```
    }  
}
```


Aufgabe 16 (7 Punkte)

Eine `DraengelSchlange` arbeitet nach dem Prinzip einer Schlange. Man kann Objekte wie in einer normalen Schlange am Ende einfügen. Außerdem gibt es aber noch die Möglichkeit, ein Objekt beim Eintragen "vordrängeln" zu lassen, das bedeutet, dass es am Anfang der Schlange eingefügt wird. Vervollständigen Sie dazu die Methode `draengeln(Object x)` in der gegebenen Klasse.

Implementieren Sie die von `VerweisSchlange` abgeleitete Klasse `DraengelSchlange`, indem Sie die Methode `public void draengeln(Object x)` ergänzen, die das `Object x` ganz vorn in die Schlange einfügt.

Verwenden Sie zum Zwischenspeichern von Elementen die Hilfsschlange `tmp`, die in der Methode bereits vorgegeben ist. Greifen Sie auf die abgeleitete Schlange `DraengelSchlange` und auf die Hilfsschlange nur mit den Schlangenoperationen zu und implementieren Sie keine weiteren Methoden.

```
public class DraengelSchlange extends VerweisSchlange{
    public void draengeln(Object x) {
```

```
        Schlange tmp = new VerweisSchlange();
```

```
    }
}
```