Fragenkatalog ESOP

1 Einleitung

1.1 Was ist Programmieren?

1.2 Was ist ein Programm?

1.3 Welche Schritte werden bei der Programmerstellung benötigt?

1.4 Was ist ein Algorithmus?

1.5 Was sind Variablen im Kontext der Programmierung?

1.6 Welche verschiedenen Arten von Anweisungen kennen Sie?

1.7 Geben Sie einen Algorithmus an, der zwei Variablen x und y vertauscht (Java-ähnlicher Pseudocode)

```
int x = 5;
int y = -10;
```

1.8 Was ist ein Compiler, was ist ein Interpreter und was sind die Unterschiede?

1.9 Was ist Maschinencode, was ist Assembler und was sind die Unterschiede?

1.10 Nennen Sie eine Möglichkeit einen Algorithmus zu spezifizieren bzw. zu beschreiben, bevor er als Programm umgesetzt wird und geben Sie ein Beispiel für einen einfachen Algorithmus Ihrer Wahl an.

2 Einfache Programme

2.1 Grundsymbole: Dürfen Schlüsselwörter als Namen verwendet werden? Begründen Sie Ihre Antwort.

2.2 Grundsymbole: Was sind Schlüsselwörter und wofür werden Sie in einer Programmiersprache benötigt? Geben Sie zwei Beispiele für Schlüsselwörter in Java an.

2.3 Grundsymbole: Was benennen Namen und was sind die Vorgaben für Namen in Java?

2.4 Bestimmen Sie den Typ (int, float, String, usw.) der folgenden Symbole unter der Annahme, dass die Symbole sich an die Java-Notation halten:

- 10
- 12.3
- 13.1f
- true
- “false“
- `3`
- “hello world“

2.5 Wie werden Variablen in Java deklariert und initialisiert? Geben sie ein Beispiel für eine ganzzahlige Variable mit dem Namen x und dem Wert 10 an.

2.6 Welche Werte kann eine Variable vom Typ int in Java annehmen? (Hinweis: wie viele Bits werden bei der Speicherung von einer Variable vom Typ int genutzt?)

2.7 Was sind Kommentare und wofür werden Kommentare beim Programmieren benötigt?

2.8 Was ist eine Zuweisung? Geben Sie ein Beispiel an.

2.9 Was passiert bei folgender Zuweisung? Was ist der finale Wert von x?

```
int x = 12.3f
```

2.10 In welcher Form sind die Typen int, byte, short und long zuweisungskompatibel? Geben Sie die Typhierarchie an.

2.11 Nennen Sie mindestens fünf Operatoren für arithmetische Ausdrücke und geben Sie an ob diese unär oder binär sind.

2.12 Welches Ergebnis liefert der arithmetische Ausdruck 7-3-2 in Java und warum?

2.13 Wie werden verschiedene Typen bei arithmetischen Ausdrücken behandelt, welche Typregeln kennen Sie?

2.14 Einer der folgenden arithmetischen Ausdrücke liefert einen Fehler. Geben Sie an welcher und begründen Sie ihre Antwort.

```
short s; int i;
i = i + 1;
i = i + s;
s = s + 1;
```

2.15 Was ist der Unterschied zwischen while(x++<10) und while(++x<10)?

2.16 Was sind shift Operationen und was kann man damit machen?

2.17 Welcher der folgenden Ausdrücke liefert den Wert 8 für x = 2?

- x << 1

- x >> 1
- x << 2
- x >> 2

2.18 Welchen Wert hat x nach folgenden Operationen:

int x = 4;

x %= 3;

3 Verzweigungen und Schleifen

3.1 Welche Arten eine Verzweigung in Java zu programmieren kennen Sie?

3.2 Mit welchen Schlüsselwörtern kann man eine Schleife in Java programmieren?

3.3 Was ist der Unterschied zwischen einer while und einer do ... while Schleife?

3.4 Ist Einrückung in Java verpflichtend? Wenn ja, warum, wenn nein, warum wird es trotzdem gemacht?

3.5 Welche Vergleichsoperatoren für Zahlen kennen Sie?

3.6 Welche Vergleichsoperatoren für Bool'sche Variablen kennen Sie?

3.7 Was ist an folgendem Beispiel falsch?

```
int i = 0
while (i = 0) {
	System.out.println("i ist 0");
	if (Math.random() > 0.8) {
		i++;
	}
}
```

3.8 Was wird bei nachfolgendem Beispiel ausgegeben?

```
int i = 3;
boolean y = true;
if ( (i < 3) && !y) {
	System.out.println("1");
} else if (i > 3) {
	System.out.println("2");
} else {
	System.out.println("2");
}
```

3.9 Nennen Sie mindestens eine der beiden DeMorgan'schen Regeln.

3.10 Warum wird die break-Anweisung in der switch-Verzweigung benötigt?

3.11 Wie oft werden folgende Schleifen ausgeführt?

- for (i = 0; i < n; i++)
- for (i = 10; i > 0; i--)
- for (int i = 0; i <= n; i = i + 1)
- for (int i = 0, j = 0; i < 10n && j < 10; i = i + 1, j = j + 2)
- for (;;)

3.12 Was sind die Teile x, y und z eine Zählschleife for (x;y;z)

3.13 Was macht die break-Anweisung, wenn sie innerhalb einer Schleife aufgerufen wird?

3.14 Wann ist eine break-Anweisung in einer Schleife typischerweise vertretbar?

4 Gleitkommazahlen, Methoden und Arrays

4.1 Was sind Gleitkommazahlen, wieso heißen sie „Gleitkommazahlen" und wie werden sie in Java benutzt?

4.2 Wie hoch ist die Genauigkeit von double / float (in Bit)?

4.3 Welchen Typ haben folgende Zahlen?

- 3.14
- 3.14f
- 3.14d
- 0.314E1

4.4 Was passiert, wenn Zahlen verschiedenen Typs durch arithmetische Operatoren verknüpft werden. Erklären Sie anhand folgender Beispiele und geben Sie Typ und Wert der Ergebnisse an.

- 3.14 * 2
- 3.14f *2.0
- 4 / 16

4.5 Was sind Methoden?

4.6 Was ist der Unterschied zwischen Methoden und Funktionen?

4.7 Wie definiert man in einem Programm die main-Methode einer Klassen? Geben Sie die Methodendefinition an und erläutern Sie was die einzelnen Schlüsselwörter bedeuten.

4.8 Was ist der Unterschied zwischen formalen und aktuellen Parametern?

4.9 Welches Schlüsselwort wird benutzt um einen Rückgabewert aus einer Methode zu liefern und wie muss ein Rückgabewert in der Methodendeklaration definiert werden?

4.10 Was bedeutet das Schlüsselwort void in einer Methodendefinition?

4.11 Was bedeutet das Schlüsselwort static in einer Methodendefinition?

4.12 Was ist der Unterschied zwischen einer Funktion und einer Prozedur?

4.13 Was ist der Gültigkeitsbereich (Scope) einer Variablen?

4.14 Erläutern Sie für nachfolgendes Beispiel was die Ausgabe des Programms ist (1, 2 oder 3) und in welchem Block (A, B, C oder D) die ausgegebene Variable gültig ist.

```
{ // Block A
        int k = 0;
        if (true) { // Block B
                int k = 2;
                for (int i = 0; i < 10; i++) { // Block C
                        k = 3;
                        if (i==5) { // Block D
                                System.out.println(k);
                        }
                }
        }
}
```

4.15 Erläutern Sie das Lokalitätsprinzip.

4.16 Was versteht man unter „Überladen von Methoden“?

4.17 Was sind varargs in Java?

4.18 Was sind Arrays?

4.19 Erläutern Sie den Unterschied zwischen ein- und zweidimensionalen Arrays.

4.20 Wie groß ist der maximale Index eines Arrays?

4.21 Warum ist folgendes Beispiel falsch?

```
int[] arr = {1, 2, 3};
System.out.println(arr[1d]);
```

4.22 Warum ist folgendes Beispiel falsch?

```
int[] arr = {1, 2, 3};
System.out.println(arr[1*3]);
```

4.23 Warum ist folgendes Beispiel falsch?

```
int[] arr = {1, 2, 3, 4, 5};
System.out.println(arr[2.0*2.0]);
```

4.24 Was ist eine „for each“-Schleife?

4.25 Was sind Kommandozeilenparameter und wie werden diese in der main-Methode ausgelesen?

5 Klassen und Objekte

5.1 Was sind Referenzdatentypen und Basisdatentypen? Geben Sie jeweils ein Beispiel an.

5.2 Was ist der Unterschied zwischen „call by reference“ und „call by value“?

5.3 Erläutern Sie den Zusammenhang zwischen Klasse und Instanz.

5.4 Was ist eine Klasse in Java und wofür kann sie benutzt werden?

5.5 Was ist ein Konstruktor?

5.6 Was bedeutet das Schlüsselwort „static“?

5.7 Geben Sie ein Beispiel für Vererbung in Klassenhierarchien an.

5.8 Wofür wird Vererbung benötigt?

5.9 Was bedeutet das Schlüsselwort „abstract“ und wofür setzt man es ein?

6 Information Hiding

6.1 Was bedeutet das Schlüsselwort „super“?

6.2 Geben Sie ein Beispiel an, das das Schlüsselwort „super“ einsetzt (Hinweis: Konstruktor).

6.3 Erläutern Sie das Geheimnisprinzip.

6.4 Welche Schlüsselwörter werden für die Umsetzung von Information Hiding in Java benötigt.

6.5 Geben Sie an für wen eine Klasse, Variable oder Methode sichtbar ist, wenn sie als private/public/protect oder gar nicht gekennzeichnet ist.

6.6 Was sind dynamische Datenstrukturen?

6.7 Welchen Vorteil hat eine dynamische Liste gegenüber einem Array?

6.8 Erläutern Sie den typischen Aufbau von dynamischen Datenstrukturen in kurzen Worten und graphisch (Hinweis: Knoten und Kanten)

6.9 Erläutern Sie (graphisch bzw. schrittweise) wie bei einer dynamischen Liste (einfach verkettet) etwas zu Beginn / am Ende / in der Mitte eingefügt werden kann.

7 Rekursion, Interfaces

7.1 Was heißt rekursiv, direkt rekursiv und indirekt rekursiv?

7.2 Beantworten Sie folgende Fragen für nachfolgende Methode: Ist die Methode iterativ oder rekursiv? Was ist das Ergebnis für k = 3?

```
int foo(int k) {
  if (k > 0)
        return foo(--k) + 1;
  else
        return 1;
}
```

7.3 Was ist ein Interface in Java, und wie wird es definiert?

7.4 Nennen Sie einen Anwendungsfall, in dem ein Interface sehr nützlich ist.

7.5 Nennen Sie ein beliebiges Interface aus der Standard-Java-Klassenbibliothek und erläutern Sie in kurzen Worten wofür es benötigt wird.

7.6 Welches Ergebnis liefert Math.floor(Math.random())?

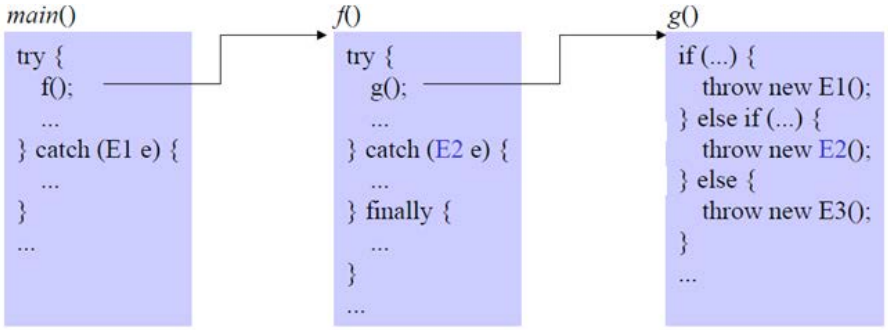
8 Exceptions

8.1 Wofür braucht man in Java Exceptions?

8.2 Was ist bei der Verwendung von Exceptions zu bedenken (hinsichtlich Laufzeit, übermäßige Nutzung, usw.)?

8.3 Was ist der Unterschied zwischen Laufzeitfehlern und geprüften Ausnahmen?

8.4 in welcher Methode werden die Ausnahmen behandelt wenn die Ausnahmen (i) E1, (ii) E2, (iii) E3 auftreten?



8.5 Im folgenden Beispiel tritt eine IOException in Zeile 6 auf. Markierren Sie alle Zeilen, die ausgeführt werden, geben Sie die Ausgabe des Programms (beispielhaft) an und erläutern Sie ob der InputStream korrekt geschlossen wird.

```
class Test {
    public static void main(String[] args) {
        try {
            File file = new File("C:\\Temp\\file.txt");
            InputStream fis = new FileInputStream(file);
```

```
            byte b = fis.read();   // IOException tritt hier auf
            System.out.println("Result = " + b);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            fis.close();
        }
    }
}
```

8.6 Was ist der Unterschied zwischen einem InputStream und einem Reader?

8.7 Was ist der Unterschied zwischen einem OutputStream und einem Writer?

8.8 Was ist ein InputStreamReader?

8.9 Was ist der Unterschied zwischen einem BufferedReader und einem Reader?

9 Pakete und Collections

9.1 Was sind Pakete und wofür kann man Sie in Java nutzen?

9.2 Wie wird Paketzugehörigkeit in Java Quelltext definiert?

9.3 Wo sind folgende Variablen sichtbar (nur in der Klasse, im Paket, in Unterklassen, überall)?

```
private int a;
        int b;
protected int c;
public int d;
```

9.4 Was sind die Java Collections? Welche Klassen aus den Java Collections kennen Sie?

9.5 Was ist der Unterschied zwischen List und Set?

9.6 Was ist der Unterschied zwischen ArrayList und LinkedList?

9.7 Welche Klasse bietet schnelleren Zugriff auf ein beliebiges Element der Liste (random access)? LinkedList oder ArrayList?

9.8 Welche Klasse erlaubt schnelleres Löschen von einem beliebigen Listenelement? LinkedList oder ArrayList?

10 Innere Klassen und Threads

10.1 Was sind innere Klassen?

10.2 Sind anonyme Klassen innere Klassen?

10.3 Was unterscheidet lokale Klassen von anonymen Klassen?

10.4 Wann werden anonyme Klassen typischerweise benutzt? Erläutern Sie ein kurzes Beispiel.

10.5 Was sind Threads und welchen Vorteil hat die Nutzung von mehreren Threads?

10.6 Hat ein Java Programm Threads, auch wenn keine Threads explizit erzeugt werden? Begründen Sie ihre Antwort.

10.7 Was macht die join() Methode eines Threads?

10.8 Können mehrere Threads gleichzeitig auf ein und dieselbe Festplatte zugreifen? Begründen Sie Ihre Antwort.

10.9 Wie viele CPU-Kerne können von folgendem Programm gleichzeitig genutzt werden? Begründen Sie ihre Antwort.

```
public class ThreadExample implements Runnable {
    String text;
```

```
    int count = 500;

    public ThreadExample(String text) {
        this.text = text;
    }

    public static void main(String[] args) {
        new Thread(new ThreadExample("-")).start();
        new Thread(new ThreadExample("O")).start();
        new Thread(new ThreadExample("/")).start();
        new Thread(new ThreadExample("|")).run();
    }

    @Override

    public void run() {
        while (count-- > 0) System.out.print(text);
    }
}
```