Übungsblatt 8: Komplexität und Vertiefung zu Klassen

Deadline: 04.12.2025 12:00 Uhr
Besprechung: 08.–12.12.2025

Hinweise für die Bearbeitung der Aufgaben

- Bitte lesen Sie die Aufgabenstellungen genau durch.
- Testen Sie Ihre Lösungen vor der Abgabe selbst!
- Halten Sie sich an die Java Coding Konventionen.
- Damit die automatischen Tests korrekte Ergebnisse liefern können, achten Sie bitte auf die folgenden Punkte:
  - Halten Sie sich bitte genau an die Vorgaben/Vorlagen aus der Aufgabenstellung.
  - Wenn Sie Umlaute verwenden, stellen Sie die Kodierung Ihrer Dateien unbedingt auf UTF-8.
  - Achten Sie darauf, dass Ihr Projekt mit Java Version 21 kompiliert und ausgeführt werden kann.
  - Ihre Java-Klassen müssen sich im default package befinden (es darf kein package angegeben werden).

Aufgabe 1 Abgabe in GATE
Komplexität

a) Auf Blatt 4 haben wir ein Dreieck auf der Konsole ausgegeben. Bestimmen Sie die Zeitkomplexität dieses Algorithmus'. In welche $\mathcal{O}$-Komplexitätsklasse fällt er? Begründen Sie Ihre Lösung mit einem Satz.

```
static void drawTriangle(int sizeOfTriangle) {
  for (int i = 0; i < sizeOfTriangle; i++) {
    for (int j = 0; j <= i; j++) {
      System.out.print("*");
    }
    System.out.println();
  }
}
```

b) Bestimmen Sie die Zeitkomplexität des folgenden Algorithmus'. In welche $\mathcal{O}$-Komplexitätsklasse fällt er? Begründen Sie Ihre Lösung mit einem Satz.

```
static void berechneB(int[] array) {
  for (int i = 0; i < Math.min(1000,array.length-1); ++i) {
    array[i] = array[i+1];
  }
}
```

c) Bestimmen Sie die Zeitkomplexität des folgenden Algorithmus'. In welche $\mathcal{O}$-Komplexitätsklasse fällt er? Begründen Sie Ihre Lösung mit einem Satz.

```
static void berechneC(int[] array) {
    for (int i = 0; i < 5; ++i) {
        for (int j = i; j < array.length; ++j) {
            array[j] = array[j] + i;
        }
    }
}
```

Aufgabe 2 Abgabe in GATE
Verkettete Listen

Diese Aufgabe bezieht sich auf die abstrakten Datentypen "einfach-verkettete Liste" und "Warteschlange", die in der Vorlesung eingeführt wurden. Implementieren Sie das Interface `Queue` gemäß der Spezifikation (im Interface) für eine Warteschlange mit der Klasse `LinkedQueue` unter Verwendung einer einfach-verketteten Liste. Ihre Klasse `LinkedQueue` benötigt einen leeren Konstruktor.

Queue.java

```
public interface Queue {
  final int EMPTY_VALUE = -1;

  /**
   * Fuegt den Wert value (am Ende) in die Wartschlange ein
   * @param value
   */
  void append(int value);

  /**
   * Prueft, ob die Warteschlange leer ist
   * @return true, genau dann wenn die Warteschlange leer ist
   */
  boolean isEmpty();

  /**
   * Entfernt das erste Element aus der Warteschlange
   * Macht nichts, wenn die Warteschlage leer ist
   */
  void remove();

  /**
   * Gibt den Wert des ersten Elements der Liste zurück
   * Wenn die Liste leer ist, wird EMPTY_VALUE zurückgegeben
   * @return Wert des ersten Elements
   */
  int peek();

  /**
   * Erzeugt ein Array mit den Werten der Warteschlage
   * @return neues Array mit den Werten der Warteschlange
   */
  int[] toArray()
```

```
}
```

Aufgabe 3 Abgabe in GATE
Binärer Suchbaum

Zeichnen Sie einen binären Suchbaum, in den Sie die folgenden Werte in dieser Reihenfolge einfügen: 7, 5, 13, 11, 1, 8, 6, 15, 2, 3

Aufgabe 4 Abgabe in GATE mit Peer-Review
Zahlen Raten

Der Computer soll eine von Ihnen gedachte Zahl zwischen 1 und 1.000 raten in dem er Ihnen Fragen der Form „Ist die von Ihnen gedachte Zahl größer als 10?“ stellt, die Sie per „ja“ oder „nein“ beantworten. Schließlich soll er die gefundene Zahl in der Form „Die gedachte Zahl ist: 5“ ausgeben.

a) Welcher aus der Vorlesung bekannte Algorithmus eignet sich dafür?

b) Implementieren Sie das oben beschriebene Programm mithilfe des Algorithmus aus a) in Java. Erstellen Sie hierzu eine Klasse `NumberGuesser` mit einer Methode `public static void guess(int maxNumber)`. Die Methode `guess` soll so lange "ja" oder "nein" einlesen bis sie die gedachte Zahl (zwischen 0 (inklusive) und `maxNumber` (inklusive)) weiß und ausgeben kann.

   Beispiel Programmausgabe für mit der gedachten Zahl `4` (A, B, C und D sind dabei Platzhalter für konkrete Zahlen):

```
Ist die gedachte Zahl groesser als A?
nein
Ist die gedachte Zahl groesser als B?
nein
Ist die gedachte Zahl groesser als C?
ja
Ist die gedachte Zahl groesser als D?
ja
Die gedachte Zahl ist: 4
```

   Halten Sie sich exakt an dieses Format.

c) Wie oft muss der Computer für Zahlen zwischen 1 und 15.000 raten, um eine Lösung zu erhalten? Geben Sie den Rechenweg an!

d) Welchen Zahlenbereich würde der Computer abdecken, wenn er 4-mal raten dürfte? Geben Sie den Rechenweg an!

Abgabe

Die Abgabe erfolgt bis zum 04.12.2025 12:00 Uhr auf GATE, https://gate.ifi.lmu.de.

Zuletzt aktualisiert: 17. November 2025