

Diese Aufgaben decken ein breites Spektrum an wichtigen Konzepten ab und sollten dir eine solide Grundlage bieten, um die relevanten Themen zu verstehen und anzuwenden. Hier ist eine detaillierte Zusammenstellung der Aufgaben, um sicherzustellen, dass du gut vorbereitet bist:

GK-Aufgabe: Klassen und Objekte

Erstelle eine Java-Klasse `Student`, die folgende Attribute hat: `name` (String), `studentId` (int) und `gpa` (double). Schreibe Methoden, um:

1. Einen neuen Studenten zu erstellen.
2. Die Details eines Studenten auszugeben.
3. Den `gpa` eines Studenten zu aktualisieren.

Zusätzlich, schreibe eine Hauptklasse `StudentManagement`, die ein Array von `Student`-Objekten verwaltet und Methoden enthält, um:

1. Einen neuen Studenten hinzuzufügen.
2. Einen Studenten nach seiner ID zu suchen.
3. Alle Studenten auszugeben.
4. Den Durchschnitts-GPA aller Studenten zu berechnen.
5. Den Studenten mit dem höchsten GPA zu finden.

EK-Aufgabe 1: Sortieralgorithmen

Implementiere die Sortieralgorithmen: Bubble Sort und Merge Sort. Schreibe ein Programm, das ein Array von Ganzzahlen sortiert und die sortierten Arrays ausgibt. Füge außerdem Methoden hinzu, um:

1. Eine Liste von Studenten nach ihrem GPA aufsteigend zu sortieren.
2. Eine Liste von Studenten nach ihrem Namen alphabetisch zu sortieren (verwende hierfür eine geeignete Sortiermethode).

EK-Aufgabe 2: Überschreiben von `hashCode` und `equals`, Java-Records

1. Implementiere eine Klasse `Person` mit den Attributen `name` (String) und `birthDate` (LocalDate). Überschreibe die Methoden `hashCode` und `equals` auf korrekte Weise.
2. Verwende Java-Records, um eine ähnliche Struktur zu implementieren. Erstelle ein Record `PersonRecord` mit denselben Attributen und implementiere eine Methode, um die Gleichheit der Record-Instanzen zu überprüfen.
3. Implementiere eine Methode, um eine Liste von `Person`-Objekten nach dem Geburtsdatum zu sortieren.

EK-Aufgabe 3: Zusätzliche knifflige Methoden

1. Implementiere eine Methode in der `StudentManagement`-Klasse, die alle Studenten zurückgibt, deren GPA über einem bestimmten Schwellenwert liegt.
2. Implementiere eine Methode, die überprüft, ob ein bestimmter Name in der Liste der Studenten enthalten ist.
3. Implementiere eine Methode, die zwei Listen von Studenten vereint und doppelte Einträge (Studenten mit derselben `studentId`) entfernt.

Hinweise zur Vorbereitung

- **Grundlegende Konzepte:** Stelle sicher, dass du die Grundlagen von Klassen, Objekten, Arrays und den grundlegenden Datenstrukturen gut verstanden hast.
- **Sortieralgorithmen:** Lerne die Funktionsweise der Sortieralgorithmen (Bubble Sort, Merge Sort) und übe deren Implementierung.
- **hashCode und equals:** Verstehe, warum und wie man diese Methoden überschreibt, um Objekte korrekt in Hash-basierten Strukturen zu verwenden.
- **Java-Records:** Mache dich mit der neuen Syntax und den Vorteilen von Records vertraut.
- **Erweiterte Methoden:** Übe die Implementierung komplexerer Methoden, die mehrere Objekte und Sammlungen manipulieren.

Wenn du diese Aufgaben und Konzepte beherrschst, solltest du gut auf die Prüfung vorbereitet sein. Viel Erfolg!