**Advanced workbook**

Aufgabenstellung zum Kurs:

DLBCSDSJCL02\_D – Datenstruktur und Java-Klassenbibliotheken

**Bitte überprüfe die Gültigkeit dieser Workbookaufgaben im Kurs auf myCampus.**

**Hinweis zum Urheberrecht und zur Plagiatsprüfung:**

**Es wird darauf hingewiesen, dass der IU Internationale Hochschule GmbH das Urheberrecht der Prüfungsaufgaben/Aufgabenstellungen obliegt. Einer Veröffentlichung der Aufgabenstellungen auf Drittplattformen wird ausdrücklich widersprochen. Im Falle einer Zuwiderhandlung stehen der Hochschule u.a. Unterlassungsansprüche zu. Zudem weisen wir darauf hin, dass jede eingereichte schriftliche Ausarbeitung mittels einer Plagiatssoftware überprüft wird. Wir empfehlen daher auch, keinesfalls ausgearbeitete Lösungen zu teilen, da dies den Verdacht eines Plagiates begründen kann.**

****Aufgabenstellung 1:****

Angenommen, Du entwickelst eine Anwendung, die eine Datenbank mit Kundendaten verwaltet. Dabei werden Informationen wie Name, Alter und E-Mail-Adresse erfasst. Beschreibe, wie Du Wrapper-Klassen in Java verwenden kannst, um diese Kundendaten effizient zu speichern und zu verarbeiten. Erkläre, welche Wrapper-Klassen Du verwenden würdest und wie Du mit den primitiven Datentypen und den entsprechenden Wrapper-Klassen umgehen würdest, um die Funktionalität Ihrer Anwendung zu verbessern. Gib Beispiele für die Verwendung von Autoboxing und Unboxing sowie der zusätzlichen Funktionen der Wrapper-Klassen, um die Kundendaten zu manipulieren und auszuwerten. Diskutiere die Vorteile der Verwendung von Wrapper-Klassen in diesem Szenario und stelle mögliche Herausforderungen und Lösungen bei der Verwendung von Wrapper-Klassen dar.

****Aufgabenstellung 2:****

Implementiere eine eigene Map-Implementierung namens "ArrayMap" in Java, die auf einem Array basiert. Die ArrayMap soll die grundlegenden Funktionen der Map-Schnittstelle unterstützen, einschließlich put(), get(), remove() und size(). Stelle sicher, dass die ArrayMap Schlüssel-Wert-Paare speichern und bei Bedarf erweitert werden kann. Berücksichtige dabei auch die Behandlung von Kollisionen, wenn mehrere Schlüssel denselben Hash-Wert haben. Verwende generische Typen, um die Flexibilität der ArrayMap zu erhöhen.

****Aufgabenstellung 3:****

Stelle Dir vor, Du bist Teil eines Entwicklerteams, das an einer Anwendung zur Simulation eines Kassensystems arbeitet. Die Anwendung soll eine Warteschlange für Kunden implementieren, die an verschiedenen Kassenstellen anstehen. Für die Implementierung der Warteschlange möchtest Du die Datenstruktur ArrayDeque in Java verwenden. Beschreibe ausführlich, wie Du die ArrayDeque nutzen und welche Vorteile sie für die Umsetzung dieses Szenarios bietet.

Erkläre zunächst, was eine ArrayDeque ist und wie sie im Vergleich zu anderen Datenstrukturen wie einer einfachen ArrayList oder einer LinkedList funktioniert. Diskutiere die besonderen Merkmale der ArrayDeque, einschließlich ihrer Fähigkeit, Elemente sowohl am Anfang als auch am Ende der Warteschlange effizient hinzuzufügen und zu entfernen.

Beschreibe dann, wie Du die ArrayDeque verwenden würdest, um die Warteschlange für Kunden in Deinem Kassensystem zu implementieren. Erkläre, wie Du Kunden am Ende der Warteschlange hinzufügst und wie Du Kunden bedienst, indem Du sie am Anfang der Warteschlange entfernst. Diskutiere die Vorteile der Verwendung einer ArrayDeque für diese Aufgabe, wie zum Beispiel die Effizienz beim Hinzufügen und Entfernen von Elementen und die Unterstützung von gleichzeitigen Operationen am Anfang und Ende der Warteschlange.

Gehe auch auf die erweiterten Funktionen der ArrayDeque ein, wie z. B. die Möglichkeit, Elemente am Anfang oder Ende der Warteschlange abzurufen oder zu ändern. Erörtere, wie diese Funktionen in Deinem Kassensystem nützlich sein könnten, beispielsweise um Kundeninformationen abzurufen oder um spezielle Anforderungen bei der Bearbeitung von Kunden zu erfüllen.

****Aufgabenstellung 4:****

**Erläutere das Konzept einer Javadoc-Dokumentation an einem selbst gewählten Beispiel. Inwiefern kann Javadoc zur Vereinfachung und Verkürzung von Codedokumentation beitragen? Gehe bei Deinen Ausführungen u. a. auf die drei Javadoc-Tags @param, @return sowie @throws ein.**

****Aufgabenstellung 5:****

Angenommen, Du entwickelst ein Online-Shopsystem und möchtest die Warteschlangen (Queues) in Java verwenden, um Bestellungen zu verarbeiten. Beschreibe, wie Du Queues in Deinem System verwenden würdest, um die Bestellungen effizient zu verarbeiten und die Reihenfolge der Bestellungen zu gewährleisten. Diskutiere die Implementierungsdetails, einschließlich der Verwendung der java.util.Queue-Schnittstelle und einer geeigneten Queue-Implementierung wie LinkedList. Erkläre, wie Du Bestellungen in die Warteschlange einfügen und sie in der richtigen Reihenfolge verarbeiten würdest. Diskutiere auch, wie Du mit verschiedenen Szenarien wie Priorisierung von Bestellungen, Stornierungen oder Rückgaben umgehen würdest. Gib Beispiele für die Verwendung von Methoden wie offer(), poll() und peek() in der Queue-Schnittstelle, um die Bestellverarbeitung zu steuern. Diskutiere die Vorteile der Verwendung von Queues in diesem Szenario und stelle mögliche Herausforderungen und Lösungen bei der Verwendung von Queues in einem Online-Shopsystem dar.

****Aufgabenstellung 6:****

Angenommen, Du entwickelst ein Restaurantbuchungssystem in Java, das die Verwaltung von Buchungen für verschiedene Tische ermöglicht. Eine wichtige Funktion des Systems ist es, Kundenbenachrichtigungen zu generieren und zu formatieren. Erkläre, wie Du die Manipulation von Strings in Java verwenden würdest, um benutzerdefinierte Benachrichtigungen zu erstellen. Beschreibe die verschiedenen Methoden und Funktionen, die Du verwenden würdest, um den Text der Benachrichtigungen zu formatieren, Kundeninformationen einzufügen und dynamische Daten zu generieren. Gib konkrete Beispiele für die Verwendung von String-Manipulationsfunktionen wie concat(), format(), replace(), substring() und trim(). Diskutiere die Vor- und Nachteile der Verwendung von String-Manipulationstechniken für die Generierung von Benachrichtigungen in Bezug auf Lesbarkeit, Flexibilität und Leistung. Stelle außerdem mögliche Herausforderungen und Lösungen bei der Manipulation von Strings in diesem Szenario dar.