

Programmieren 2 – Klausuraufgaben

Grundlagen:

) Der Unterschied zwischen dem JDK (Java Development Kit) und dem JRE (Java Runtime Environment) besteht darin, dass das JDK die Tools und Ressourcen enthält, die zum Entwickeln und Kompilieren von Java-Anwendungen benötigt werden, während das JRE die Umgebung bereitstellt, um Java-Anwendungen auszuführen.

b) Nein, String ist kein Grunddatentyp in Java. Es ist eine Klasse, die verwendet wird, um Zeichenketten darzustellen.

c) Ein Unterschied zwischen den Grunddatentypen von Java und C besteht darin, dass in Java die Größe der Datentypen auf allen Plattformen gleich ist, während in C die Größe der Datentypen von der Plattform abhängen kann.

d) Der Operator "|" (Pipe) ist ein bitweiser Operator, der eine logische ODER-Verknüpfung auf Bit-Ebene durchführt. Der Operator "||" (Double Pipe) ist ein logischer Operator, der eine logische ODER-Verknüpfung auf Ausdrucksebene durchführt. Der Unterschied besteht darin, dass der Operator "|" immer beide Operanden auswertet, während der Operator "||" den zweiten Operanden nur dann auswertet, wenn der erste Operand nicht ausreicht, um den Ausdruck zu bestimmen.

e) Der Unix-Timestamp 3600 entspricht dem Datum und der Uhrzeit 1. Januar 1970, 01:00:00 Uhr.

f) Ein Thread ist ein Ausführungspfad innerhalb eines Prozesses, der unabhängig von anderen Threads ausgeführt werden kann. Ein Prozess hingegen ist eine Instanz eines laufenden Programms, das aus mehreren Threads bestehen kann. Ein Prozess hat seinen eigenen Speicherbereich und Ressourcen, während Threads den Speicherbereich und die Ressourcen des Prozesses teilen.

Klassenmodell, Programmierung:

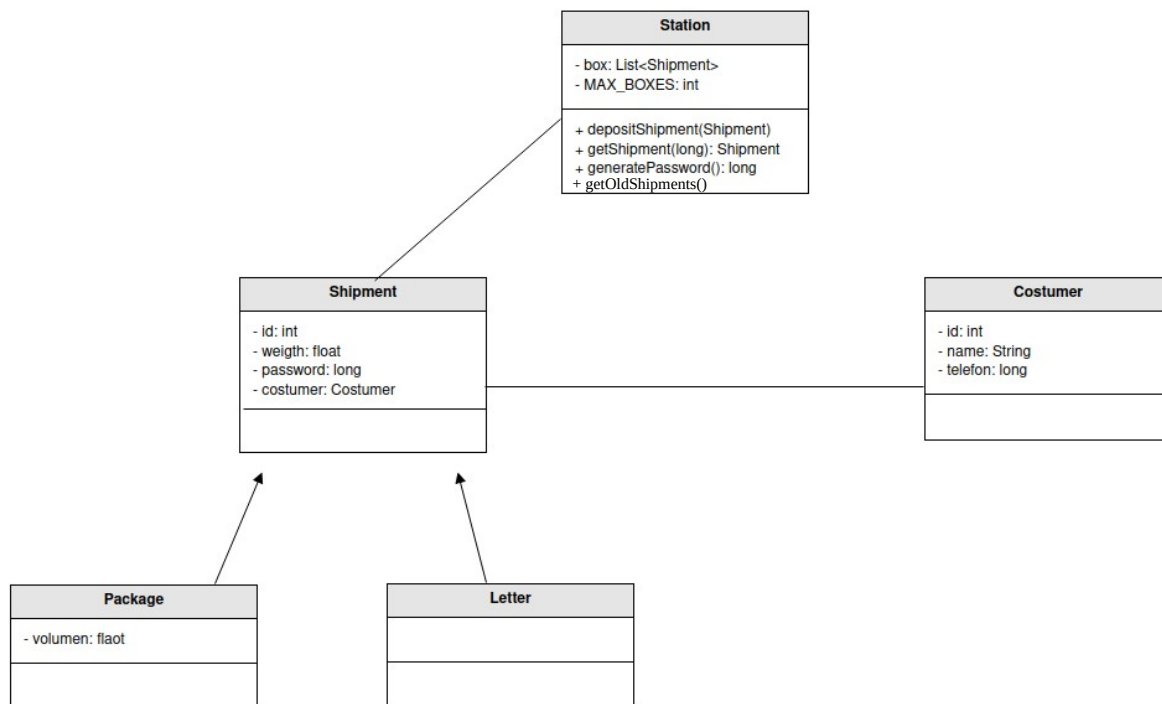
An einem Post-Packstationsautomaten können Kunden hinterlegte Sendungen abholen. Eine Sendung kann dabei ein Brief oder ein Paket sein. Diese werden zu einem bestimmten Zeitpunkt in die Fächer der Station gelegt. Dieser Zeitpunkt wird bei der Sendung gespeichert. Für jeden Brief werden eine eindeutige Nummer und das Gewicht gespeichert. Bei Paketen wird neben dem Gewicht und der eindeutigen Nummer auch zusätzlich das Volumen des Paketes gespeichert.

Die Station hat genau 18 Fächer. Jedes Paket bzw. jeder Brief wird in ein eigenes Fach der Station gelegt. Beim Hinterlegen in der Station wird dem Post-Mitarbeiter ein beliebiges freies Fach vorgeschlagen, sofern noch freie Fächer vorhanden sind. Größenbeschränkungen werden hier nicht betrachtet.

Für jede Sendung wird weiterhin eine zufällige erzeugte Zahl als Passwort gespeichert, sobald die Sendung in ein Fach der Station gelegt wird. Dieses Passwort muss der Kunde nennen, um die Sendung aus der Station entnehmen zu können. Hier können Sie davon ausgehen, dass der Kunde das Passwort per SMS bekommt.

Von jeder Sendung ist bekannt, welchem Kunden die Sendung gehört. Ein Kunde hat einen Namen, eine Kundennummer und eine Telefonnummer.

1. Erstelle ein vereinfachtes Klassenmodell mit Attributen, Methoden und Datentypen, das für diese Programmieraufgabe geeignet ist.



2. Programmieren Sie die Methode(n), welche das Einlegen einer Sendung in die Station abbildet bzw. abbilden. Dabei soll das zufällige Passwort erzeugt und bei der Sendung gespeichert werden.

```

public class Poststation {
    private static final int MAX_BOXES = 30;
    private List<Shipment> boxes;

    public Poststation() {
        boxes = new ArrayList<>();
    }

    public void depositShipment(Shipment s) {
        if (boxes.size() < MAX_BOXES) {
            boxes.add(s);
            s.setPassword(generateRandomPassword());
        } else {
            System.out.println("Kein freies Fach mehr vorhanden.");
        }
    }
}

```

3. Ergänzen und programmieren Sie eine Methode, die alle Sendungsnummern zusammen mit den Namen des jeweiligen Kunden in der Command-Line anzeigt, die zum Zeitpunkt des Aufrufes der Methode länger als 8 Tage in der Station liegen.

```

public void displayOldShipments() {
    Calendar eightDaysAgo = Calendar.getInstance();
    eightDaysAgo.add(Calendar.DAY_OF_YEAR, -8);

    for (Shipment box : boxes) {
        if (box.getTimestamp().before(eightDaysAgo.getTime())) {
            System.out.println("Sendungsnummer: " + box.getId() + ", Kunde: " + box.getCustomer().getName());
        }
    }
}

```

- d) Programmieren Sie weiterhin eine Methode, welche den Benutzer in der Command-Line dazu auffordert, das Passwort einer Sendung einzugeben. Die Methode sucht die zum Passwort passende Sendung in der Station und entfernt sie aus der Station, sofern sie gefunden wird. Wenn keine passende Sendung gefunden wird, erhält der Benutzer eine entsprechende Meldung.

Anmerkung: Sie können davon ausgehen, dass jedes erzeugte Zufallspasswort nur ein einziges Mal existiert.

```

public Shipment getShipment() {
    System.out.print("Bitte geben Sie das Passwort ein: ");
    long enteredPassword = sc.nextLong();

    Shipment pack = null;
    for (Shipment box : boxes) {
        if (box.getPassword() == enteredPassword) {
            pack = box;
            boxes.remove(box);
            return pack;
        }
    }

    System.out.println("Falsches Passwort oder Sendung nicht gefunden.");
    return null;
}

```

Codeverständnis:

1. Beschreiben Sie die

Hauptaufgabe des folgenden Java Programms in zwei bis drei Sätzen. Nennen

```
import java.util.Scanner;

public class WasMacheIch {
    private static int size = 2;
    private static long[][] array;

    public static void main (String[] args) {
        array = new long[size][size];
        Scanner sc = new Scanner(System.in);
        for(int i = 0; i < size; i++) {
            for(int j = 0; j < size; j++) {
                System.out.println("Geben Sie Zahl " + i + "/" + j + " ein: ");
                array[i][j] = sc.nextLong();
            }
        }
        System.out.println("Ergebnis: ");
        System.out.println(anonymous(array));
    }

    private static float anonymous(long[][] array) {
        long result = 0;
        for(int i = 0; i < array.length; i++) {
            long curNumber = array[i][i];
            if(curNumber % 2 == 0) {
                result += curNumber;
            }
        }
        return result;
    }
}
```

Das Java-Programm erstellt eine quadratische Matrix der Größe 2x2 (definiert durch die Variable `size`). Der Benutzer wird aufgefordert, Zahlen für jedes Element der Matrix einzugeben. Anschließend wird eine Methode namens `anonymous` aufgerufen, die die Hauptdiagonale der Matrix durchläuft und die Summe aller geraden Zahlen in der Hauptdiagonale berechnet. Das Ergebnis wird dann auf der Konsole ausgegeben.

Beispiel:

Eingabe:

...

Geben Sie Zahl 0/0 ein: 2

Geben Sie Zahl 0/1 ein: 3

Geben Sie Zahl 1/0 ein: 4

Geben Sie Zahl 1/1 ein: 6

...

Ergebnis:

...

Ergebnis:

8.0

...

In diesem Beispiel sind die geraden Zahlen in der Hauptdiagonale 2 und 6, und ihre Summe ist 8.0.

- b) In der folgenden Klasse sind zwei Fehler enthalten. Nennen Sie die beiden Fehler und begründen Sie Ihre Wahl.

```
1  import java.util.List;
2
3  public class Fehlerhaft {
4
5      public static void main(String[] args){
6
7          List<Integer> list = new int[10];
8
9          for (Integer element : list){
10             if (element > 10){
11                 list.remove(element);
12                 System.out.println("Removed");
13             }
14         }
15     }
16 }
```

List<Integer> list = new ArrayList<>();

oder

```
Integer[] array = new Integer[10];
List<Integer> list = Arrays.asList(array);
```

In der Liste sind keine Elemente enthalten

Man kann nicht gleichzeitig über eine Liste iterieren und die elemente löschen. Das würde zu einer Exception führen. : Der Iterator arbeitet mit einem internen Index, der auf das aktuelle Element zeigt. Wenn Sie ein Element entfernen, ändert sich die Größe der Liste und die Indexverwaltung könnte durcheinander geraten.

a) Erklären Sie den Begriff der "quasi-parallelen" Ausführung mehrerer Threads auf einem Einprozessorsystem.

Der Begriff "quasi-parallele Ausführung" bezieht sich auf die Situation, in der mehrere Threads auf einem Einprozessorsystem scheinbar gleichzeitig ausgeführt werden. Obwohl der Prozessor nur einen Thread gleichzeitig ausführen kann, wechselt er zwischen den Threads hin und her, um den Eindruck einer parallelen Ausführung zu erzeugen.

b) Programmieren Sie eine Klasse mit dem Namen "Test", die sich als Thread starten lässt und 5000 Zufallszahlen in der Command-Line ausgibt. Ergänzen Sie eine main-Methode, welche zwei dieser Threads startet. Sobald beide Threads fertig sind, soll die Meldung "fertig" in der Command-Line angezeigt werden.

```
import java.util.Random;

public class Test extends Thread {
    public static void main(String[] args) {
        Thread thread1 = new Test();
        Thread thread2 = new Test();

        thread1.start();
        thread2.start();

        try {
            // Wartet bis beide beendet
            thread1.join();
            thread2.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("fertig");
    }

    @Override
    public void run() {
        Random random = new Random();
        for (int i = 0; i < 5000; i++) {
            System.out.println(Thread.currentThread().getName() + ": " + random.nextInt());
        }
    }
}
```