



**Die Aufgaben 1 bis 4 beziehen sich auf die folgende Ausgangssituation:**

Die Parkanlage „Wilder Garten“ in Köln konnte u. a. mit ihrer reichhaltigen Pflanzenvielfalt, ihren schönen verwinkelten Wegen, ihren vielen schattigen Plätzen und ihren beliebten Attraktionen in der letzten Saison ihre Anziehungskraft als Naherholungsfläche für viele Besucher steigern. Besucher aus allen Altersgruppen und auch Reiseveranstalter zählen zu den Kunden der Parkanlage.

**1. Aufgabe (25 Punkte)**

Sie werden von Ihrem Ausbilder beauftragt, den Prozess zur Anlage neuer Pflanzungen mittels eines UML-Aktivitätsdiagramms zu dokumentieren. Um den Prozess zu verstehen, sprechen Sie mit mehreren Mitarbeitern.

Sie beginnen mit einem Mitarbeiter der Gartenplanung.

Dieser teilt Ihnen mit, dass er den Gärtnern den Auftrag zu einer neuen Pflanzung erteilt.

Wenn die Gärtner fertig gepflanzt haben, fertigt er Bilder des neuen Bereichs an. Wenn die Bilder gemacht sind und der Einkauf die Rechnung bezahlt hat, ist der Vorgang abgeschlossen.

Informationen eines Gärtners:

Wenn der Gärtner einen neuen Pflanzauftrag vom Gartenplaner erhält, stellt er zuerst die Anforderungen zusammen und sagt dem Einkauf welche Pflanzen zu bestellen sind.

Wenn Pflanzen ankommen, bekommt er diese vom Einkauf und bringt die Pflanzen in die Quarantänestation. Der Gärtner prüft täglich die Pflanzen und kontrolliert das Quarantäneende.

Ist eine Pflanze in Quarantäne erkrankt, so muss diese behandelt und die Quarantäne für alle Pflanzen verlängert werden.

Sind alle Pflanzen gesund und die Quarantäne abgelaufen pflanzt der Gärtner die Pflanzen.

Nach Abschluss der Pflanzarbeiten macht die Gartenplanung noch Bilder.

Der Einkauf gibt Ihnen folgende Hinweise:

Der Gärtner teilt uns mit was er braucht, dann bestellen wir diese Pflanzen bei unserem Lieferanten. Bei Anlieferung übergeben wir die Pflanzen an den Gärtner und bezahlen die Rechnung.

Weiterhin beschreibt der Einkauf den Vorgang beim Lieferanten so, dass der Lieferant den Auftrag annimmt, danach die Lieferung zusammenstellt und dann die Pflanzen ausliefert.

Vervollständigen Sie aufgrund der erhaltenen Informationen das abgebildete Aktivitätsdiagramm.



## 2. Aufgabe (25 Punkte)

Die Verwaltung des Parks möchte eine Auswertung über die Besucherzahlen für alle Tage eines vorgegebenen Monats erhalten.

Für die Besuchertickets werden daher an den Ein – und Ausgängen folgende Informationen festgehalten:

- Das Ticket wird beim Einlass gescannt und damit das Datum, die Uhrzeit und die Anzahl der Personen (Gruppentickets) gespeichert.
- Beim Hinausgehen wird jede Person mit Datum und Uhrzeit einzeln erfasst.

Der Park hat aktuell die Öffnungszeiten von 9:00 Uhr bis 19:00 Uhr.

Zur Auswertung soll für jeden Tag eines Monats festgehalten werden, wie viele Personen sich von 9:00 Uhr bis 9:59 Uhr, von 10:00 Uhr bis 10:59 Uhr, ... und von 18:00 Uhr bis 18:59 Uhr aufgehalten haben.

Die folgende Klasse ist bereits vorhanden:

|                       |
|-----------------------|
| ComeLeave             |
| – date: Date          |
| – comelInOut: Integer |
| – noPeople: Integer   |
| ...                   |

Für jedes Attribut sind öffentliche Zugriffsmethoden (set/get) vorhanden.

Folgende Methoden können verwendet werden:

|                                  |                                                  |
|----------------------------------|--------------------------------------------------|
| getDay() der Klasse Date         | Ermittelt zum Datum den Tag (1 – 31)             |
| getHour() der Klasse Date        | Ermittelt zum Datum die Stunde (0 – 23)          |
| getDaysOfMonth() der Klasse Date | Ermittelt zum Datum die Anzahl der Tage im Monat |

In der Tabelle **entry** vom Typ **ComeLeave** sind die Kommen – und Gehenzeiten der Besucher für den auszuwertenden Monat erfasst (eine Zeile entspricht den Attributen eines Objektes vom Typ ComeLeave):

| date      | time  | comelInOut<br>0 = In/1 = Out | noPeople |
|-----------|-------|------------------------------|----------|
| ...       |       |                              |          |
| 22.5.2023 | 9:00  | 0                            | 1        |
| 22.5.2023 | 9:00  | 0                            | 2        |
| 22.5.2023 | 9:01  | 0                            | 30       |
| ...       |       |                              |          |
| 22.5.2023 | 12:00 | 1                            | 1        |
| 22.5.2023 | 12:01 | 0                            | 2        |
| 22.5.2023 | 12:02 | 1                            | 1        |
| ...       |       |                              |          |

Erstellen Sie eine Funktion `countVisitors(entry: ComeLeave) : Integer[][]`, die folgende Anforderung erfüllt:

- Für jede Stunde eines Tages (Öffnungszeiten!) und für jeden Tag des auszuwertenden Monats soll die Anzahl der anwesenden Personen in einem zweidimensionalen Array vom Typ Integer gespeichert werden.

Für jeden Tag des Monats soll eine Zeile und für jede Stunde innerhalb der Öffnungszeiten eine Spalte verwendet werden.

Dieses Array ist der Rückgabewert der Funktion `countVisitors()`.

Dabei wird eine Person ab einem Stundenabschnitt gezählt, wenn sie in diesem Abschnitt den Park betreten hat. Wenn eine Person den Park verlässt, wird die Anzahl erst für die nachfolgenden Abschnitte entsprechend korrigiert.

Beispielarray für ein Ergebnis:

| Tag\Std | 9:00-<br>9:59 | 10:00-<br>10:59 | 11:00-<br>11:59 | 12:00-<br>12:59 | 13:00-<br>13:59 | 14:00-<br>14:59 | 15:00-<br>15:59 | 16:00-<br>16:59 | 17:00-<br>17:59 | 18:00-<br>18:59 |
|---------|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1       | 23            | 40              | 56              | 120             | 145             | 160             | 140             | 95              | 86              | 45              |
| 2       | 27            | ...             | ...             | ...             | ...             | ...             | ...             | ...             | ...             |                 |
| ...     | ...           | ...             | ...             | ...             | ...             | ...             | ...             | ...             | ...             |                 |
| 31      | ...           | ...             | ...             | ...             | ...             | ...             | ...             | ...             | ...             |                 |

countVisitors(entry: ComeLeave) : Integer[][]

Korrekturrand

### 3. Aufgabe (25 Punkte)

Korrekturrand

Die Verwaltung des Parks hat eine Jahresstatistik für die Nutzung der Monatstickets in Auftrag gegeben. Die implementierte Methode soll einem Test unterzogen werden.

Die Daten liegen in einer Liste, aufsteigend sortiert nach Monat, vor und haben die folgende Struktur:

```
struct TicketData
{
    int Id;           //Ticket-Id
    string Monat;     //Monat
    string Jahr;      //Jahr
    int NutzungsZaehler; //Anzahl der Nutzungen (tagesweise)
}
```

Für den Test sollen die folgenden Testdaten verwendet werden:

| Id | Monat   | Jahr | Nutzungszaehler |
|----|---------|------|-----------------|
| 0  | Januar  | 2023 | 31              |
| 1  | Januar  | 2023 | 31              |
| 2  | Februar | 2023 | 28              |

Aus diesen Daten soll die folgende Ausgabe generiert werden:

#### Nutzungsstatistik Monatstickets

Auswertung für Monat Januar

Minimale Nutzung: 31

Maximale Nutzung: 31

Durchschnitt: 31

Gesamtanzahl Tickets: 2

Auswertung für Monat Februar

Minimale Nutzung: 28

Maximale Nutzung: 28

Durchschnitt: 28

Gesamtanzahl Tickets: 1

Auswertung für Jahr 2023

Minimale Nutzung: 28

Maximale Nutzung: 31

Durchschnitt: 30

Gesamtanzahl Tickets: 3

**Abbildung zur 3. Aufgabe**

```
01 TicketStatistik(List<TicketData> TicketDataList) : void
02   WriteLn("Nutzungssstatistik Monatstickets");
03   WriteLn();
04   TicketData td0 = TicketDataList[0];
05   string monat = td0.Monat;
06   int monatMin = GetMonatstage(monat);
07   int monatMax = 0;
08   int monatTicketZaehler = 0;
09   int monatNutzungsZaehler = 0;
10   int jahrMin = 31;
11   int jahrMax = 0;
12   int jahrTicketZaehler = 0;
13   int jahrNutzungsZaehler = 0;
14   foreach(TicketData td in TicketDataList)
15     if ( monat != td.Monat )
16       //Monatsauswertung:
17       WriteLn("Auswertung für Monat" + monat);
18       WriteLn(" Minimale Nutzung:" + monatMin);
19       WriteLn(" Maximale Nutzung:" + monatMax);
20       WriteLn(" Durchschnitt:" + monatNutzungsZaehler/monatTicketZaehler);
21       WriteLn(" Gesamtanzahl Tickets:" + monatTicketZaehler);
22       WriteLn();
23       //Monatsdaten für Jahresauswertung übernehmen:
24       jahrTicketZaehler = jahrTicketZaehler + monatTicketZaehler;
25       jahrNutzungsZaehler = jahrNutzungsZaehler + monatNutzungsZaehler;
26       if ( jahrMax < monatMax )
27         jahrMax = monatMax;
28       end if
29       if ( jahrMin > monatMin )
30         jahrMin = monatMin;
31       end if
32       //Reset Monat:
33       monatTicketZaehler = 0;
34       monatNutzungsZaehler = 0;
35       monatMax = 0;
36       monat = td.Monat;
37       monatMin = GetMonatstage(monat);
38     end if
39     //Ticketauswertung:
40     monatTicketZaehler += 1;
41     monatNutzungsZaehler += td.NutzungsZaehler;
42     if ( monatMax < td.NutzungsZaehler )
43       monatMax = td.NutzungsZaehler;
44     end if
45     if ( monatMin > td.NutzungsZaehler )
46       monatMin = td.NutzungsZaehler;
47     end if
48   end foreach
49   //Jahresauswertung:
50   WriteLn("Auswertung für Jahr" + td0.Jahr);
51   WriteLn(" Minimale Nutzung:" + jahrMin);
52   WriteLn(" Maximale Nutzung:" + jahrMax);
53   WriteLn(" Durchschnitt:" + jahrNutzungsZaehler / monatTicketZaehler);
54   WriteLn(" Gesamtanzahl Tickets" + jahrTicketZaehler);
55   WriteLn();
56 end TicketStatistik
```



a) Überprüfen Sie den Code auf der perforierten Seite 7 mit Hilfe der auf Seite 6 genannten Testdaten. Verwenden Sie dazu die Datenverlaufsstabelle.

Korrekturrand

Datenverlaufstabelle:

[illegible]

aa) Nennen Sie kurz zwei gefundene Fehler.

2 Punkte

ab) Machen Sie Vorschläge für die Änderung bzw. Erweiterung des Codes, so dass die erwartete Ausgabe erzeugt wird.

13 Punkte

Fortsetzung 3. Aufgabe →

b) Der Test soll alle Anweisungen mindestens einmal durchlaufen (Anweisungsüberdeckung, coverage test).

ba) Erläutern Sie die Anweisungsüberdeckung.

2 Punkte

bb) Mit den vorgegebenen Testdaten werden nicht alle Anweisungen des Codes durchlaufen.

Nennen Sie die Anweisungen, die nicht durchlaufen werden.

4 Punkte

bc) Passen Sie die Testdaten so an, dass alle Anweisungen durchlaufen werden und begründen Sie ihre Veränderungen.

4 Punkte

#### 4. Aufgabe (25 Punkte)

Korrekturrand

Es liegt nachfolgender Datenbankauszug zur Verwaltung der Pflegearbeiten im botanischen Garten vor. Für die Verwaltung und Auswertung der Daten soll die Datenbankabfragesprache SQL zum Einsatz kommen.

##### Objekt

| OID | Bezeichnung     |
|-----|-----------------|
| 1   | Rasenfläche Süd |
| 2   | Rosengarten     |
| 3   | Tropenhaus      |

##### Mitarbeiter

| MID | Name | Vorname |
|-----|------|---------|
| 1   | Rose | Ines    |
| 2   | Baum | Jürgen  |
| 31  | Holz | Walter  |

##### Tätigkeit

| TID | Bezeichnung   |
|-----|---------------|
| 1   | Rosenschnitt  |
| 2   | Rasen mähen   |
| 3   | Unkraut jäten |

##### Pflegearbeit

| PID | OID | TID | MID_Soll | MID_Ist | Datum_Soll | Datum_Abschluss |
|-----|-----|-----|----------|---------|------------|-----------------|
| 1   | 1   | 2   | 31       | 31      | 20.04.2023 | 20.04.2023      |
| 2   | 1   | 2   | 2        | 2       | 11.05.2023 | NULL            |
| 3   | 2   | 1   | 1        | 2       | 20.03.2023 | 21.03.2023      |
| 4   | 2   | 1   | 1        | NULL    | 01.08.2023 | NULL            |
| 5   | 2   | 3   | 31       | NULL    | 28.07.2023 | NULL            |

- a) Erstellen Sie eine SQL-Abfrage, mit der die Anzahl der Pflegearbeiten ermittelt wird, welche im Jahr 2023 geplant aber noch nicht abgeschlossen sind. 5 Punkte

---

---

---

---

---

---

- b) Erstellen Sie eine SQL-Abfrage, mit der Sie ermitteln, an welchen Wochentagen in der Zeit vom 19.06.2023 bis einschließlich 30.06.2023 welche Pflegearbeiten in der „Außenanlage Nord“ stattfinden sollen. Das Ergebnis soll nach dem Datum aufsteigend sortiert angezeigt werden. 8 Punkte

Beispiel-Ergebnistabelle:

| Datum      | Wochentag  | Tätigkeit     |
|------------|------------|---------------|
| 19.06.2023 | Montag     | Heckenschnitt |
| 19.06.2023 | Montag     | Rasen mähen   |
| 22.06.2023 | Donnerstag | Unkraut jäten |
| 26.06.2023 | Montag     | Rosenschnitt  |
| 30.06.2023 | Freitag    | Rosenschnitt  |

---

---

---

---

---

---

---

---

---

---

Fortsetzung 4. Aufgabe →

#### Fortsetzung 4. Aufgabe

Korrekturrand

- c) Erstellen Sie eine SQL-Abfrage mit der Sie die durchschnittliche Anzahl der Tätigkeiten pro Monat im Jahr 2021 von jedem Mitarbeiter auflisten. 8 Punkte

Beispiel-Ergebnistabelle:

| MID | Name  | Vorname | Durchschnitt |
|-----|-------|---------|--------------|
| 1   | Rose  | Ines    | 25,3         |
| 2   | Baum  | Jürgen  | 17,8         |
| 19  | Knoll | Jana    | 0            |
| 31  | Holz  | Walter  | 30,2         |

- d) Geben Sie alle SQL-Anweisungen an, welche notwendig sind, um einen neuen Nutzer „Maier“ mit dem Passwort „5jk2T?“ zu erstellen und diesem die Leserechte an der Tabelle Objekt zuzuweisen. 4 Punkte

#### PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!

Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende Prüfungszeit?

- ☐ 1 Sie hätte kürzer sein können.  
☐ 2 Sie war angemessen.  
☐ 3 Sie hätte länger sein müssen.

☐