

## Aufgabe zur APL in Programmierung 1 im WS 2019/2020

Die APL (Alternative Prüfungsleistung) besteht aus einem eigenständig zu bearbeitenden Programmierprojekt.

Es ist ein C-Programm zu entwickeln, das zur Organisation eines Fahrradverleihs benutzt werden kann. Das Programm soll **Reservierungen** speichern, sowie den Verleih mit **Ausgabe** und **Rücknahme** der **Fahrräder verwalten**. Es sollen jeweils  **$n$  Damen- und Herrenfahrräder** zum Verleih angeboten werden. Der Wert von  **$n$**  soll über eine **Konstante** im Programm festgelegt werden, **ganzzahlig** und **größer als 1** sein. Zum **Testen** des Programms wird  **$n=2$**  empfohlen, für den anvisierten Einsatz soll das Programm mit größeren  **$n$ -Werten** arbeiten können. Eine **Kategorie D** für Damenrad bzw. **H für Herrenrad** wird genutzt, um die Wahl eines geeigneten Rads bei der Reservierung zu steuern. Die einzelnen **Fahrräder** sollen mit ihrer **Kategorie** und einer **Nummer** (**Bereich 1 bis  $n$** ) identifiziert werden. Die **Zeitangaben** für Verleihbeginn und -ende erfolgen mit **Tag, Monat, Jahr, Stunde, Minute** entsprechend der Wertevorräte für kalendarische Datumsangaben und Uhrzeitangaben. **Minuten** sind nur in einem Raster von vollen **15 Minuten** zu erfassen. Alle **Reservierungen** und **Verleihvorgänge** werden mit **Zeiträumen versehen**, die jeweils eine **Zeitangabe für Anfang und für Ende** enthalten.

Für den Verleiher, der als Bediener des Programms fungiert, sind die unten beschriebenen Funktionalitäten bereitzustellen:

Die **Reservierung** soll für ein **einzelnes Fahrrad** einer **Fahrradkategorie** (D oder H) erfolgen. Bei der Reservierung sind eine **fünfstellige Kundennummer**, eine **fünfstellige Reservierungsnummer**<sup>1</sup>, die gewünschte Kategorie, die geplante **Ausgabezeit und Rückgabezeit** zu erfassen. Reservierungen werden vom **Programm bestätigt**, wenn ein Fahrrad in der gewünschten Kategorie **verfügbar ist**, anderenfalls abgelehnt. Programmintern wird für ein freies Fahrrad in dieser Kategorie die Reservierung vorgenommen, dabei die Kundennummer, die Reservierungsnummer, der Zeitraum für das Fahrrad und die Nummer des Fahrrads in den Datenstrukturen des Programms gespeichert.

Eine **Stornierung** zu einer Reservierung wird unter **Angabe** einer **Kundennummer** und einer Reservierungsnummer vorgenommen. Vom Programm soll **geprüft** werden, ob tatsächlich ein Fahrrad reserviert wurde. Wenn das der Fall ist, dann wird programmintern die Reservierung für das Fahrrad entfernt.

Die **Ausgabe** eines reservierten Fahrrads soll mit Angabe der **Reservierungsnummer** erfolgen. Das **Kennzeichen** des in der Reservierung festgelegten Fahrrads soll **angezeigt** werden. Programmintern wird das betroffene Fahrrad als verliehen gekennzeichnet. Die Herausgabe zum Verleih soll immer nur mit einer vorangehenden Reservierung erfolgen.

Bei **Rücknahme** wird die Kategorie und die **Nummer des Fahrrads**, sowie die **Rückgabezeit** erfasst. Dabei wird **geprüft**, ob das angegebene **Fahrrad tatsächlich ausgeliehen** war und eine entsprechende Meldung erzeugt. In den Datenstrukturen des Programms ist der betreffende Eintrag zur Reservierung und zum Verleih des Fahrrads zu entfernen. Auch eine verspätete Rückgabe soll gehandhabt werden.

---

<sup>1</sup> Aus Gründen der Vereinfachung sollen die Reservierungsnummern hier als Eingabe bereitgestellt werden. Die Nummern müssen aus Gründen der Zuordenbarkeit einmalig (ohne Doppelungen) vergeben werden. In echten Programmsystemen würde eine Nummer mit dieser Eigenschaft vom System selbst generiert werden.

Eine **Anzeigefunktion** soll alle Reservierungen und die zum Zeitpunkt der Anzeige ausgeliehenen Fahrräder auf der Konsole ausgeben. Dabei sind alle Daten zu einer Reservierung bzw. einer Ausleihe zeilenweise darzustellen, beispielsweise so:

Knr	Rnr	Kennz	Art	Ausgabe	Rückgabe
11201	00012	H2	Reservierung	10-08-2019 14:00	10-08-2019 18:00
11399	00014	D1	Ausgeliehen	08-08-2019 17:00	10-08-2019 17:00

### Funktionale Anforderungen:

- Das Programm muss mehrere Reservierungen und Verleihvorgänge für ein ausgewähltes Fahrrad zulassen, sofern die Zeiträume nicht überlappen.
- Unterschiedliche Fahrräder sollen für gleiche oder überlappende Zeiträume reserviert, bzw. verliehen werden können.
- Die begrenzte Menge der Fahrräder ist zu berücksichtigen.
- Die in den Datenstrukturen des Programms eingetragenen Daten sind bei Programmende in eine Textdatei zu schreiben. Beim nächsten Start des Programms sind die Daten wieder aus der Datei einzulesen.
- Alle Eingaben (Zeitangaben, Kundennummer, Reservierungsnummer usw.) sind bezüglich Format zu prüfen (z.B. Anzahl der Stellen). Datums- und Zeitangaben müssen darüber hinaus tatsächliche Zeitpunkte ausdrücken. Für Zeiträume muss der Beginn vor dem Ende liegen.
- Das Programm soll gegenüber Eingaben im fehlerhaften Format robust reagieren, d.h. sollte keine fehlerhaften Eingaben übernehmen und keine Fehlfunktionen hervorrufen.

### Anforderungen an die technische und programmiersprachliche Umsetzung:

- Zur Ein- und Ausgabe von einzelnen Zeitangaben und von Zeiträumen sind C-Funktionen zu nutzen, insb. um Code-Dopplungen zu vermeiden.
- Funktionalitäten wie z.B. Reservierung, Ausgabe, Rücknahme, Stornierung sind durch C-Funktionen abzubilden, die von der Nutzerinteraktion getrennt sind.
- Die gespeicherten Daten sind in Struktur-Elementen (struct) zu organisieren und wahlweise durch eine Form der dynamischen Speichertechnik (Feld variabler Länge, dynamische Feld oder lineare verkettete Liste) zu handhaben. Die Nutzung eines statischen Feldes (fester Länge) führt bei der Bewertung zu Punktabzug.
- Das C-Programm muss unter Windows innerhalb Visual Studio übersetzt und gestartet werden können. Fremdkomponenten (z.B. für Datenbankzugriff, Containerklassen) sind für diese Aufgabe nicht erlaubt.

Für den Betreiber des Systems sollen zusätzlich ausgewählte Funktionen angeboten werden, die von Studierenden individuell zu bearbeiten sind.

Dazu wird jede/r Studierende einer Gruppe (A bis D) zugeordnet (siehe unten) und hat die Funktion/en für seine Gruppe zu programmieren.

A: Studierende mit gerader Matrikelnummer und gerader s-Nummer.

B: Studierende mit gerader Matrikelnummer und ungerader s-Nummer.

C: Studierende mit ungerader Matrikelnummer und gerader s-Nummer.

**D:** Studierende mit ungerader Matrikelnummer und ungerader s-Nummer.

## Aufgaben für Gruppen

### Gruppe A:

Bei Reservierungen soll die Wahl des Fahrrads so erfolgen, dass die Kategorie eingehalten wird, aber ein Fahrrad ausgewählt wird, für das bisher am wenigsten verliehen worden ist. Dazu sind die Verleihvorgänge für jedes Fahrrad zu zählen. Die Zählerstände sind zusätzlich zu den Daten zu den Reservierungs- und Verleihvorgängen zwischen den Programmausführungen auf einer Datei zu speichern.

### Gruppe B:

Bei Rücknahme eines Fahrrads ist ein Rechnungseintrag zu erzeugen und auf einer weiteren Textdatei abzulegen. Die Datei für die Rechnungen kann im s.g. Append-Mode immer für das Anhängen eines Eintrags geöffnet, beschrieben und wieder geschlossen werden. In der Rechnung sind Kundennummer, Reservierungsnummer, der Leihzeitraum, die Kategorie und Nummer des Fahrrads und ein Preis abzulegen. Der Preis wird über einen Stundenpreis von 4 Euro je angefangene Stunde berechnet, bzw. einen Tagespreis (24h) von 15 Euro. Es gilt der für den Kunden günstigere Preis.

### Gruppe C:

Es ist eine Funktionalität bereitzustellen, um Sammelreservierungen für mehrere Fahrräder in einem gemeinsamen Zeitraum abzugeben. Dabei wird anstelle einer Kategorie (D oder H) die Anzahl der Fahrräder in jeder Kategorie abgefragt. Sind alle gewünschten Fahrräder verfügbar, so sollen alle reserviert werden. Sind nicht alle verfügbar, dann wird kein Fahrrad reserviert und die Sammelreservierung verworfen. Die Ausgabe und Rückgabe der Fahrräder soll weiterhin einzeln erfolgen.

### Gruppe D:

Der Betreiber soll bei Eingabe einer Zeitangabe (Datum, Uhrzeit) prüfen können,

- welche Fahrräder mit ihrer Reservierung in die angegebene Zeit fallen und nicht ausgegeben wurden.
- welche Reservierungen ohne Ausgabe abgelaufen sind.
- für welche ausgegebenen Fahrräder die Rückgabezeit überschritten wurde.

Dafür sind entsprechende Funktionalitäten bereitzustellen.

### **Termine und Bewertung:**

- Ausgewählte Programmbausteine sind 2019, in Kalenderwoche 51 innerhalb des Praktikums vorzuführen. Dazu gehören nötige Typdefinitionen und Funktionen
  - zur Ein- und Ausgabe von Zeitangaben, von Zeiträumen,
  - das Prüfen der Gültigkeit von Zeitangaben und Zeiträumen
  - die Feststellung von Gleichheit, der Früher-als-, sowie der Später-als-Relation von Zeitangaben.
  - eine Funktion zum Erkennen von Überlappungen von Zeiträumen.
- Das vollständige Programm ist 2020, in Kalenderwoche 3 im OPAL-Kurs hochzuladen.
- Die APL wird mit maximal 30 Punkten bewertet. Die erreichte Punktzahl wird auf eine Note abgebildet.
- Sie müssen die Aufgabe selbständig bearbeitet haben. Lösungen die offensichtlich von anderen Studierenden kopiert wurden, können automatisiert erkannt werden und können je nach Umfang des Kopierens zu Punktabzug führen.
- In die Bewertung fließt zu einem Teil die Teilabnahme der Programmbausteine in KW 51 ein. Der andere Teil der Bewertung wird ausgehend vom abgegebenen Programm in seiner Endfassung vorgenommen.
- Das Programm wird nach funktionalen Kriterien und nach nichtfunktionalen Kriterien bewertet. Die funktionalen Kriterien umfassen das Vorhandensein der vollständigen Funktionalität laut Aufgabenstellung, sowie die Fehlerfreiheit bei der Ausführung.
- Nichtfunktionale Kriterien umfassen beispielsweise eine sinnvolle Gliederung des Programms und der Daten, die Abwesenheit von Code-Dopplungen, eine fehlerfreie Programmierung, sowie die Nutzung geeigneter Datenstrukturen.

Das zur Lösung der Aufgabe notwendige Wissen wird schrittweise bis zur Kalenderwoche 51 (vor der Weihnachtspause) in der Vorlesung behandelt.