

## Programmieren II – Übung

### Dynamische Datenstrukturen

#### Übungsfragen:

- a) Wieso werden Arrays auch als statische Datenstrukturen bezeichnet?
- b) Welche Datenstruktur-Operationen können Arrays gut, und welche nicht?
- c) Wie kann man die „dynamischen Operationen“ mit Arrays simulieren?
- d) Wie ist „dynamische Datenstruktur“ definiert?
- e) Wie kann man eine lineare Verkettung von Daten implementieren?
- f) Wie findet man den „Anfang“ einer verketteten Liste?
- g) Welche Detailschritte sind notwendig zum Anhängen und Einfügen von neuen Listendaten?
- h) Welche Detailschritte sind notwendig zum Löschen von Listendaten?
- i) Was bedeutet Listentraversierung und wie kann dies implementiert werden?
- j) Wie kann man Daten in einer Liste suchen?
- k) Was sind die Limitationen und Probleme einer einfachen verketteten Liste?
- l) Wie ist eine Liste zu erweitern, damit Daten in „richtiger“ Reihenfolge abgespeichert werden?

#### Übungsaufgabe:

Implementieren Sie einen „Token-Ring“-Netzwerk-Simulator:

Ein „Token-Ring“-Netzwerk (auch „Ringpuffer“ genannt) basiert auf einer zirkulären Netzwerkstruktur, d. h. die Verbindungen aller miteinander kommunizierenden Geräte bilden einen Kreis bzw. einen „Ring“, bei dem die einzelnen Netzwerkgeräte die Knoten (bzw. Elemente) der zugrundeliegenden (Daten-)Struktur darstellen –  
Siehe z.B.:

<http://einstein.informatik.uni-oldenburg.de/rechnernetze/token.htm>

Das „Token“ steht für ein Datenpaket, das „durch“ den Ring gesendet wird. Typischerweise enthält es die...

- (1.) eigentlichen Informationen bzw. „Daten(wert)“ (= in unserer Aufgabe einfach ein String), ...
- (2.) die Netzwerkadresse des Senders und...
- (3.) die Netzwerkadresse des Empfängers.

Die Netzwerk-Adressen sollen in dieser Aufgabe einfach mit natürlichen Zahlen („Integer“) implementiert werden. Wenn ein neues Gerät zu dem Netzwerk hinzugefügt wird, dann soll ihm eine neue und eindeutige Netzwerk-Adresse zugewiesen werden (Die Eindeutigkeit kann einfach mittels eines fortlaufenden Inkrementierens der zuletzt vergebenen Adresse realisiert werden).

Gehen Sie bei der Lösung Aufgabe am besten folgendermaßen vor:

**Schritt 1** – Implementieren Sie zunächst die interne Datenstruktur („Network“). Diese Datenstruktur soll auf einer verketteten Liste basieren, aber eine ringförmige Struktur aufweisen (d. h. das letzte Element soll wieder auf das erste Element verweisen)! Ein Listen-, oder besser gesagt „Ringelement“, soll dabei – wie oben beschrieben – ein Netzwerkgerät („Device“) darstellen. Jedes Netzwerkgerät soll neben den Informationen zur Verkettung (= eine „Nachfolger-Referenz“), die eigene Netzwerk-Adresse enthalten sowie eine Statistik (= zwei Zähler) wie viele Datenpakete es selbst gesendet („Upload“) und (erfolgreich) empfangen („Download“) hat. Die Netzwerk-Datenstruktur soll – neben eventueller weiterer benötigter Hilfsfunktionalität – die folgenden Operationen (Methoden) implementieren:

- (i) Hinzufügen eines neuen Netzwerk-Gerätes (basierend auf einer angegebenen Netzwerk-Adresse des Vorgänger-Gerätes), ...
- (ii) Entfernen eines Netzwerk-Gerätes (basierend auf einer angegebenen Netzwerk-Adresse), ...
- (iii) Ausgabe des Netzwerkes auf der Konsole (d. h. Auflistung aller enthaltenen Geräte, inklusive der jeweiligen Netzwerk-Adresse und der Statistikwerte)

→ Siehe dazu auch die Beispielausgaben auf den folgenden Seiten.

**Schritt 2** – Implementieren Sie dann eine zweite Klasse „Simulator“, die das Versenden eines Datenpaketes von einer Sender- zu einer Empfänger-Adresse für ein vorgegebenes Netzwerk simuliert. Die Simulation kann über einer Methode („Start“) angestoßen werden, bei der die Sender- und Empfängeradressen sowie die eigentlichen Daten (also in dieser Aufgabe ein String) übergeben werden können. Die Methode solle dann das zugrundeliegende Netzwerk durchlaufen und dabei schrittweise ausgeben...

- welches Gerät sendet, ...
- welches Gerät das Datenpaket weiterleitet und...
- welches Gerät das Datenpaket (erfolgreich) empfängt oder...
- wenn ein Datenpaket nicht zugestellt werden konnte.

→ Siehe dazu auch die Beispielausgabe auf den folgenden Seiten.

**Schritt 3** – Entwickeln Sie dann eine textbasierte Benutzeroberfläche, mit der Sie die Verwaltung des Netzwerkes (z. B. Hinzufügen und Entfernen von Geräten) sowie den oben beschriebenen Token-Ring-Netzwerk-Simulator steuern können. Die entsprechenden Menüpunkte, sowie damit verbunden Funktionalitäten, können Sie aus den Beispielausgaben auf den folgenden Seiten entnehmen. Eine Seite stellt dabei immer den Ablauf einer abgeschlossenen Benutzerinteraktion dar. Zum besseren Verständnis wurde dabei die folgende Farbkodierung verwendet:

- Schwarz: Ausgabe des Programms auf der Konsole
- Rot: Geänderte bzw. neu hinzugekommene(n) Ausgabe seit der letzten Benutzerinteraktion
- Blau: Eingabe(n) des Benutzers

Status des Ringpuffer-Netzwerks:

Leer

Menü:

(1) Neue Geräte hinzufügen

(2) Gerät entfernen

(3) Datenpaket versenden

(4) Programm beenden

> 1

Neue Geräte hinzufügen...

Anzahl neuer Geräte > 3

### Status des Ringpuffer-Netzwerks:

Gerät mit Adresse 1: Anzahl Uploads/Downloads = 0/0

Gerät mit Adresse 2: Anzahl Uploads/Downloads = 0/0

Gerät mit Adresse 3: Anzahl Uploads/Downloads = 0/0

### Menü:

(1) Neue Geräte hinzufügen

(2) Gerät entfernen

(3) Datenpaket versenden

(4) Programm beenden

> 1

Neue Geräte hinzufügen...

Adresse des Vorgänger-Geräts > 2

Anzahl neuer Geräte > 1

### Status des Ringpuffer-Netzwerks:

Gerät mit Adresse 1: Anzahl Uploads/Downloads = 0/0

Gerät mit Adresse 2: Anzahl Uploads/Downloads = 0/0

Gerät mit Adresse 4: Anzahl Uploads/Downloads = 0/0

Gerät mit Adresse 3: Anzahl Uploads/Downloads = 0/0

### Menü:

(1) Neue Geräte hinzufügen

(2) Gerät entfernen

(3) Datenpaket versenden

(4) Programm beenden

> 2

Gerät entfernen...

Adresse des Geräts > 2

Status des Ringpuffer-Netzwerks:

Gerät mit Adresse 1: Anzahl Uploads/Downloads = 0/0

Gerät mit Adresse 4: Anzahl Uploads/Downloads = 0/0

Gerät mit Adresse 3: Anzahl Uploads/Downloads = 0/0

Menü:

(1) Neue Geräte hinzufügen

(2) Gerät entfernen

(3) Datenpaket versenden

(4) Programm beenden

> 3

Datenpaket versenden...

Datenwert: Testpaket

Adresse des Senders: 4

Adresse des Empfängers: 1

Starte Simulation...

Gerät 4 sendet Datenpaket („Testpaket“, 4, 1)

Gerät 3 leitet Datenpaket („Testpaket“, 4, 1) weiter

Gerät 1 empfängt Datenpaket („Testpaket“, 4, 1)

Simulation beendet. Bitte eine Taste drücken...

Status des Ringpuffer-Netzwerks:

Gerät mit Adresse 1: Anzahl Uploads/Downloads = 0/**1**

Gerät mit Adresse 4: Anzahl Uploads/Downloads = **1**/0

Gerät mit Adresse 2: Anzahl Uploads/Downloads = 0/0

Menü:

(1) Neue Geräte hinzufügen

(2) Gerät entfernen

(3) Datenpaket versenden

(4) Programm beenden

> **3**

Datenpaket versenden...

Datenwert: **Anderes Testpaket**

Adresse des Senders: **1**

Adresse des Empfängers: **3**

Starte Simulation...

Gerät 1 sendet Datenpaket („Anderes Testpaket“, 1, 3)

Gerät 4 leitet Datenpaket („Anderes Testpaket“, 1, 3) weiter

Gerät 2 leitet Datenpaket („Anderes Testpaket“, 1, 3) weiter

Gerät 1 vernichtet Datenpaket („Anderes Testpaket“, 1, 3)

Simulation beendet. Bitte eine Taste drücken...

### Status des Ringpuffer-Netzwerks:

Gerät mit Adresse 1: Anzahl Uploads/Downloads = 1/1

Gerät mit Adresse 4: Anzahl Uploads/Downloads = 1/0

Gerät mit Adresse 2: Anzahl Uploads/Downloads = 0/0

### Menü:

(1) Neue Geräte hinzufügen

(2) Gerät entfernen

(3) Datenpaket versenden

(4) Programm beenden

> 4