

1 Praktikum 1: Queues und Bäume

Die Code-Vorlagen können Sie im ILIAS-Kurs zu ADS herunterladen.

1.1 Aufgabenstellung Queue

In dieser Aufgabe simulieren wir eine Warteschlange für einen gemeinsamen Drucker in einem Büro. In einem Büro, insbesondere in größeren Firmen, wird der Drucker meist von allen Computern auf dem Büroflur genutzt und befindet sich oft in einem separaten Druckerraum. Alle Computer in diesem Raum sind mit dem gleichen Drucker verbunden. Allerdings kann der Drucker immer nur einen Druckauftrag gleichzeitig ausführen, und dieser benötigt auch eine gewisse Zeit, um abgeschlossen zu werden. In der Zwischenzeit können andere Benutzer ebenfalls Druckaufträge senden. In einem solchen Fall muss der Drucker alle anstehenden Aufträge speichern, damit er sie nacheinander abarbeiten kann, sobald der aktuelle Auftrag abgeschlossen ist. Das Code-Template kann im ILIAS-Kurs heruntergeladen werden. Die benötigten Dateien finden Sie im Verzeichnis **P1.1.Printer**.

1. Lesen und verstehen Sie den vorgegebenen Code

Kommentieren Sie den Code in *printer.cpp* mit Doxygen im Javadoc-Format (**Link Dokumentation**).

```
1  /**
2   * Beispielkommentar zur Veranschaulichung der Funktion bsp.
3   *
4   * @param x In diesem Teil des Kommentars kann die Variabl x beschrieben
5   *         werden, die der Funktion übergeben werden.
6   */
```

2. Erweiterung um eine Priority-Queue

Standardmäßig verwendet das Programm eine FIFO-Queue aus der C++ Stl-Library. Modifizieren Sie den Programmcode, sodass stattdessen eine Priority-Queue genutzt wird. Hierfür können Sie die Priority-Queue der Stl benutzen oder auch nach Vorlesung selbst implementieren. Es soll der Druckauftrag mit der kürzesten Dauer zuerst gedruckt werden.

3. Erstellung des Doxygen-Outputs

Anhand der Doxygen-Kommentare kann eine interaktive Übersicht des Codes im HTML-Format erstellt werden. Installieren Sie dafür Doxygen auf Ihrem PC ([Link Doxygen Download](#)). Fügen Sie die *doxyConfig*-Datei aus ILIAS Ihrem Projektordner hinzu. Öffnen Sie ein Terminal-Fenster und navigieren Sie in den Projektordner. Unter Windows kann ein Ordner per Rechtsklick und *In Terminal öffnen* direkt aufgerufen werden.

Durch Eingabe des Befehls `doxygen .\doxyConfig` wird Doxygen anhand der Konfiguration in *doxyConfig* einen Ordner *html* erstellen. In diesem Ordner befindet sich eine Datei *index.html* welche mit einem Browser geöffnet werden kann.

4. Simulieren Sie ein Szenario, in dem mehrere Personen Druckaufträge hinzufügen

Erstellen Sie ein Szenario, indem mehrere Benutzer einen Druckauftrag erstellen.

John möchte 10 Seiten drucken, *Jerry* 4, *Jimmy* 5, *George* 2, *Bill* 11 und *Kenny* 10.

1.2 Aufgabenstellung Baum

Erweitern Sie das vorgegebene Codegerüst eines BST in *bst.cpp* um folgende Funktionalitäten und kommentieren Sie diese mit Doxygen. Die benötigten Dateien finden Sie im Verzeichnis **P1_2_BST**.

- **Höhe der Knoten**

Zu jedem Knoten soll die Höhe im Baum gespeichert werden. Modifizieren Sie dafür die Knotenstruktur und die **insert(int value)** Methode.

- **Suchen im BST**

Implementieren Sie eine Funktion **find(int value)**, die nach einem Integer-Wert sucht und den gefundenen Knoten zurückgibt.

- **Löschen im BST**

Entwickeln Sie eine Funktion **deleteValue(int value)**, die einen Knoten aus dem BST löscht. Behandeln Sie dabei alle Fälle (keine, ein oder zwei Nachfolger). Wenn zwei Nachfolger existieren, soll das Minimum des rechten Teilbaums den gelöschten Knoten ersetzen. Aktualisieren Sie außerdem die Höhen der Knoten des betroffenen Teilbaums.

- **Ausgabe des Baumes**

Die Struktur des Baumes soll auf der Konsole in In-, Pre, Post- und Levelorder ausgegeben werden können. Implementieren Sie eine oder mehrere Funktionen, die den Baum in den 4 verschiedenen Traversierungsmethoden visuell ansprechend ausgibt. Nutzen Sie dabei die Knotenhöhe aus.

- **Main Methode**

Erstellen Sie ein User-Menü, das das Einfügen, Suchen und Löschen von Knoten, sowie die Ausgabe des Baumes ermöglicht. Fügen Sie die folgenden Elemente nacheinander ein:

46, 83, 95, 22, 17, 68, 11, 82, 21, 31, 54, 99, 74

Geben Sie den Baum in Inorder-Reihenfolge aus.

Suchen Sie nach Knoten 17 und 11.

Löschen Sie die folgenden Knoten:

68, 21, 17, 95

Suchen Sie erneut nach Knoten 17 und 11 und geben Sie den Baum in allen Ausgabeformen aus.

Ihr BST die folgende Struktur haben:

