

Klausur

MMI und GUI-Programmierung

Prüfer: Prof. Dr. Malte Weiß

Vorname	Nachname	Matrikelnummer
---------	----------	----------------

Bewertung:

1	2	3	4	Gesamt	Note
10	15	50	25	100	

Zum Ablauf der Prüfung

- **Zeit für die Lösung:** 4 Stunden
- **Beginn:** 8:00 Uhr, **Ende:** 12:00 Uhr am 23.03.2021
- **Lösung wo hochladen:** in E-Learning im [Kursraum der Vorlesung](#)
- **Lösung wie hochladen (Form):** Zip-Datei im Format *mmi-exam-VORNAME-NACHNAME-MATRIKELNUMMER.zip*.
Beispiel: Student *Mathias Bauer* mit Matrikelnummer *10001234* gibt eine Zip-Datei *mmi-exam-Mathias-Bauer-10001234.zip* ab.
- **Ordnerstruktur:** Jede Aufgabe liegt in einem entsprechenden Unterordner in der Zip-Datei. Siehe aufgabenspezifische Konventionen.
- **Nicht erlaubte Hilfsmittel/Hilfe:** Kommunikation digital oder persönlich, direkt oder indirekt mit anderen Menschen (**keine** SMS, WWW-Kommunikation, Telefon, E-Mail, Chat, etc.)
- **Kompilierbarkeit:** Lösungen *müssen* kompilieren, ansonsten werden 50 % der Aufgabenpunkte abgezogen.
- **Fragen:** Für Fragen während der Klausurzeit nutzen Sie bitte das Forum im E-Learning-Bereich der Veranstaltung. Preisgabe von Lösungen oder Ansätzen führt zu Nichtbestehen der Veranstaltung inkl. Fehlversuch.
- Die Klausur gilt als **nicht angetreten**, wenn Sie bei Beginn der Klausur nicht die vorgeschriebene eidestattliche Versicherung unterschrieben haben.

Aufgabe 1: Templates (10 Punkte)

Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A1 gespeichert werden.

Schreiben Sie die folgenden Template-Funktionen:

- a) Eine Funktion `printAll` mit Platzhalter T. Sie erhält einen Vektor(`std::vector`) vom Typ T und gibt alle Elemente des Vektors Zeile für Zeile auf der Konsole aus.

Schreiben Sie ein Hauptprogramm, das die Funktion testet.

(5 Punkte)

- b) Eine Funktion `countOccurrences` mit Platzhalter T. Die Funktion erhält drei Parameter: ein Array vom Typ T, die Anzahl der Elemente des Arrays und ein Suchelement vom Typ T. Die Funktion zählt, wie oft das Element im Array vorkommt und gibt diese Anzahl zurück.

Schreiben Sie ein Hauptprogramm, das die Funktion testet.

(5 Punkte)

Aufgabe 2: Qt Grundlagen (15 Punkte)

Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A2 gespeichert werden.

Schreiben Sie in Qt eine GUI-Applikation. Im Hauptfenster gibt es folgende GUI-Elemente:

- Zwei beschriftete Eingabefelder. Diese erfassen jeweils einen String.
- Eine Schaltfläche (Button).
- Ein Ausgabefeld zur Darstellung eines Ergebnisses.

Die GUI-Applikation besitzt nun folgende Eigenschaften:

- Die Schaltfläche ist deaktiviert, wenn mindestens eines der Eingabefeld leer ist.
- Bei Betätigung der Schaltfläche werden die Strings aus den Eingabefeldern ausgelesen und konateniert (zu einem String zusammengesetzt). Für diesen zusammengesetzten Wert wird der SHA1-Hash berechnet. Der Hash-Wert wird im Ausgabefeld dargestellt.

Beispiel:

- Eingabe 1: "Hallo"
- Eingabe 2: "Welt"
- Zusammengesetzt: "HalloWelt"
- SHA1-Hash: b747f7237c23a6b970e141529f68c9363e6dbb10

Tipp: Nutzen Sie für die Generierung von Hash-Werten QHash (siehe Qt-Dokumentation).

Für die Aufgabe gelten die folgenden Auflagen:

- Setzen Sie entsprechend alle notwendigen Signals & Slots sowie deren Connections um.
- Achten Sie auf sinnvolle Beschriftungen im Designer.
- Beachten Sie gängige C++ Konventionen (Projekt-/Struktur, Namensgebung, etc.)

Aufgabe 3: Qt Programmierung (50 Punkte)

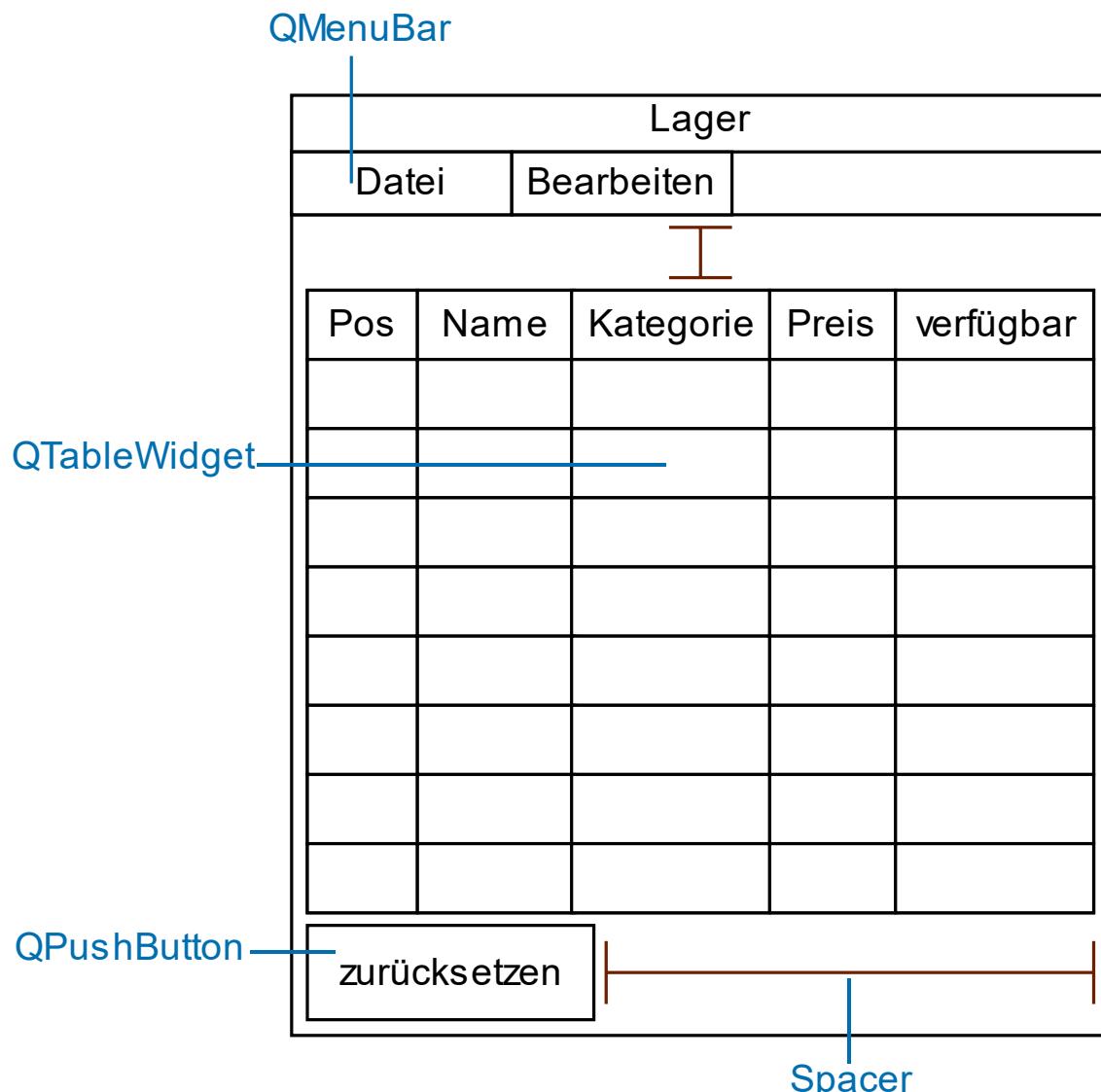
Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A3 gespeichert werden.

Es soll ein Programm zur Verwaltung von Produkten geschrieben werden. Die Produkte können innerhalb der Tabellenansicht (QTableWidget) angepasst und in der Datei aktualisiert (gespeichert) werden. Die beigegebene Datei shop.txt enthält eine beispielhafte Produkttabelle.

Ein Menü (QMenuBar) stellt eine Funktion zum Öffnen einer Produktdatei sowie zum Beenden des Programms zur Verfügung.

Des Weiteren gibt es ein Menü, um ein neues Produkt hinzuzufügen, zu ändern oder zu entfernen.

Das folgende Wireframe zeigt den Aufbau des Hauptfensters:



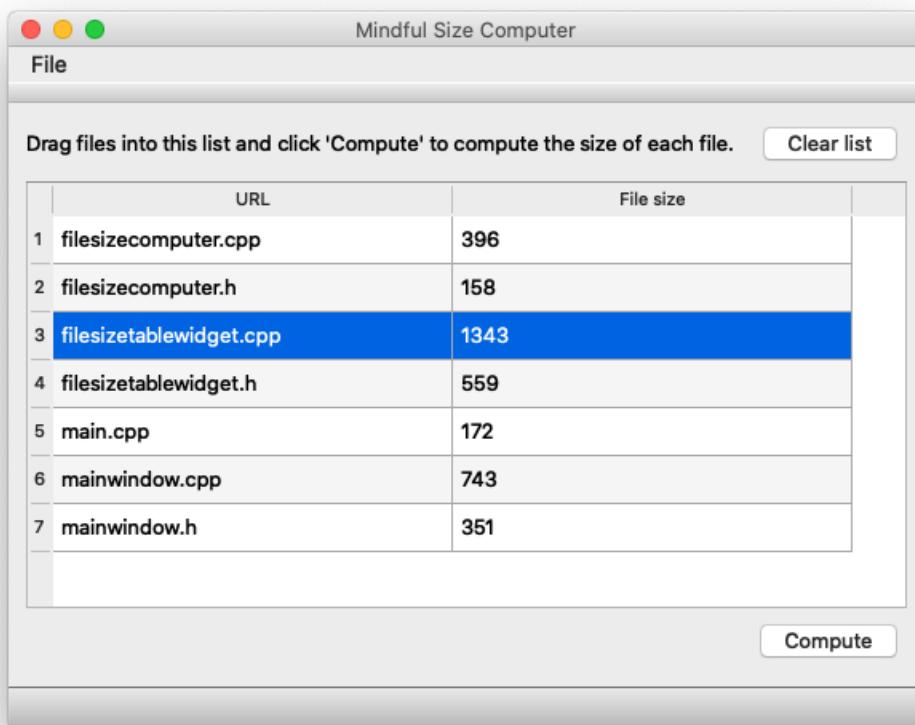
In der Tabelle können einzelne Zeilen markiert werden (keine Mehrfachauswahl möglich).

- a) Implementieren Sie mit Hilfe von Layout-Managern das obige Wireframe unter Einhaltung der gängigen Qt- und C++-Konventionen.
(11 Punkte)
- b) Fügen Sie dem Programm den Menüpunkt "Beenden" im Menü "Datei" hinzu. Wird dieser Menüpunkt angewählt, wird das Programm beendet.
(3 Punkte)
- c) Fügen Sie den Menüpunkt "Datei öffnen" im Menü "Datei" hinzu. Klickt man auf den Menüpunkt, öffnet sich ein Standard-Dialog zur Auswahl einer Textdatei. Nach Auswahl einer Textdatei wird diese eingelesen und die darin enthaltenen Einträge im QTableWidget angezeigt. Die beigegebene shop.txt zeigt den Aufbau der Textdatei.
- Sie können davon ausgehen, dass geöffnete Dateien immer ein korrektes Format besitzen. Es ist keine Formatprüfung erforderlich.
(12 Punkte)
- d) Fügen Sie einen Menüpunkt "Hinzufügen" im Menü "Bearbeiten" hinzu. Beim Klick auf den Menüpunkt öffnet sich ein Unterdialog, der es erlaubt, die Daten für ein neues Produkt zu erfassen.
- Nach Bestätigung des Dialogs wird das neu definierte Produkt der Liste hinzugefügt.
(12 Punkte)
- e) Fügen Sie einen Menüpunkt "Löschen" im Menü "Bearbeiten" hinzu. Der Menüpunkt ist nur verfügbar, wenn ein Produkt ausgewählt ist.
- Wird der Menüpunkt ausgewählt, erscheint eine Frage, ob das gewählte Produkt gelöscht werden soll. Wird dies bestätigt, wird das Produkt aus der Liste gelöscht.
(8 Punkte)
- f) Die Schaltfläche "zurücksetzen" entfernt alle Produkte aus der Liste. Zuvor wird der/die Nutzer:in aber über einen Standard-Dialog gefragt, ob er/sie das wirklich möchte.
(4 Punkte)

Aufgabe 4: Responsiveness (25 Punkte)

Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A4 gespeichert werden.

Die *Putrid Software AG* hat das preisgekrönte Programm "Mindful Size Computer" entwickelt. Es berechnet die Größe von Dateien auf besonders langsame Art, um bei den Nutzer:innen Achtsamkeit zu fördern:



Nutzer:innen können Dateien per Drag-and-Drop in die Liste ziehen und auf Compute drücken. Dann werden – auf besonders achtsame Art – die Dateigrößen berechnet und in der Spalte "File size" angezeigt.

Sie finden das entsprechende Qt-Projekt in der beigefügten ZIP-Datei `MindfulSizeComputer.zip`.

Hinweis: Unter MacOSX kann es vorkommen, dass die Dateigrößen nach der Berechnung erst angezeigt werden, wenn man auf die entsprechenden Zeilen klickt. (Qt-Bug)

- Schreiben Sie das Programm so um, dass der GUI-Thread während der Berechnung der Dateigrößen nicht mehr blockiert.

Auflagen: Das Problem muss mit Hilfe eines zusätzlichen Threads gelöst werden. Die Funktion `computeFileSize` darf in dieser Teilaufgabe nicht verändert werden.

(11 Punkte)

- b) Erweitern Sie das Programm so, dass während der Berechnung der aktuelle Fortschritt in Form einer Progress Bar angezeigt wird. Die Progress Bar sollte sich jedes Mal ändern, wenn die Größe einer Datei berechnet wurde. **(10 Punkte)**

- c) Erweitern Sie die Funktion **computeFileSize** so, dass maximal 3 Threads gleichzeitig darauf zugreifen können. **(4 Punkte)**