

Arbeitsauftrag

Kompetenzcheck

Version 1_08/2021

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. HerausgeberIn und AutorIn übernehmen für fehlerhafte Angaben, Inhalte und Anleitungen und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung. Für Verbesserungsvorschläge und Hinweise auf Fehler sind wir dankbar.

Bitte wenden Sie sich an:

+43 3573 6060

Alle Rechte vorbehalten.

© 2018 Schulungszentrum Fohnsdorf.

Dieses Werk ist urheberrechtlich geschützt.

Kein Teil dieses Werkes darf in irgendeiner Form (durch Fotografie, soziale Medien, Fotokopie, Overheadfolie, Mikrofilm oder ein anderes Verfahren) ohne vorhergehende, schriftliche Genehmigung des Schulungszentrum Fohnsdorf reproduziert oder unter Verwendung elektronischer Systeme gespeichert, verarbeitet, vervielfältigt oder verbreitet werden.

Dies gilt insbesondere für die Erstellung von Kurs- und Seminarunterlagen.

Schulungszentrum Fohnsdorf

Standort Fürstenfeld

Hauptstraße 69

Jahnstraße 30c

8753 Fohnsdorf

8280 Fürstenfeld

Tel.: +43 3573 6060

Tel.: +43 3382 51135

Fax: +43 3573 6060 1009

Fax: +43 3382 51135 9009

office@szf.at

office@szf.at

www.szf.at

www.szf.at



Ablauf

Durch den Arbeitsauftrag sollen Sie durch die Entwicklung eines kleinen Projektes ihre erlangten Kompetenzen präsentieren.

Zur Erstellung des Programmes haben Sie 2 Tage (ca. 15 Stunden) Zeit.

Zu Beginn des zweiten Tages präsentieren Sie Ihrem betreuenden Trainer-Team den momentanen Status in einem kleinen Code-Review. Sie erhalten Feedback vom Team und können dieses nutzen, um fehlende Komponenten bzw. Code Verbesserungen zu implementieren.

Der mündliche Abschnitt des Kompetenzchecks wird nach der praktischen Ausarbeitung am dritten Tag in Form eines Gespräches, Code-Reviews oder Gruppendiskussion erfolgen. Genauere Informationen dazu erhalten Sie von Ihrem betreuenden Team.

Sie erhalten von Ihrem Trainer*in einen Ordner namens

“KompCheck_(Ihr Nachname)_(Ihr Vorname)”.

In diesem befinden sich alle Unterlagen, die Sie zur Erfüllung der Aufgabenstellung benötigen.

Aufgabenstellung

Unser Kunde **GoldIsMoney-Bank** (GIM) hat uns beauftragt ein Programm zu entwickeln um Ihr veraltetes Datenbanksystem besser verwalten zu können. Die Verantwortlichen von GIM haben diesbezüglich alle Daten in CSV Dateien gespeichert. Für die Entwicklung des Programms wurden uns eigens generierte Datensätze zur Verfügung gestellt. Die angesprochenen Dateien befinden sich im Projektordner.

Programmablauf

Erstellen Sie ein Programm mit einer einfachen grafischen Oberfläche, dass diese speziellen CSV - Dateien einlesen und verarbeiten kann. Details zu den CSV Dateien finden Sie weiter unten in der Beschreibung.

1. UML - Ablaufdiagramm
2. Einlesen spezieller CSV Dateien
3. Filtern der Datensätze
4. Grafische Ausgabe
5. Speichern von Kontoauszügen
6. Dokumentation

Zu Punkt 1:

Bevor Sie mit dem Schreiben des Programms beginnen, erstellen Sie mit z.B. draw.io ein Ablaufdiagramm des Programms. Beim Programmieren gehen Sie dann nach Diagramm vor. Das Diagramm sollte Ihnen dabei helfen strukturierter beim Programmieren vorzugehen.

Zu Punkt 2:

Einlesen spezieller CSV Dateien

Im Ordner finden Sie 3 CSV Dateien, lesen Sie alle Datensätze aus.

- *Kunden.csv*
- *Buchungen.csv*
- *Konto.csv*

Rahmenbedingungen:

Ein Kunde kann mehrere Konten haben.

Ein Konto kann mehrere Buchungen beinhalten.

Zu Punkt 3:

Filtern der Datensätze

Erstellen Sie einen Algorithmus der das Filtern von CSV Dateien ermöglicht um eine detaillierte Datenausgabe zu ermöglichen.

Zu Punkt 4:

Grafische Ausgabe

Es soll möglich sein eine gesammelte Liste aller Kunden auf den Bildschirm auszugeben.
Weiters sollen Details* zu einem ausgewählten Kunden dargestellt werden.

*Details = Auflistung aller Konten, für jedes Konto sollen alle registrierten Buchungen und der aktuelle Kontostand aufgelistet werden. Der Kontostand muss erst ermittelt werden (Buchungen auf eigenes Konto)

Informale Angaben

- Das Programm soll in Methoden und Klassen strukturiert sein
- Die Daten sollten in geeigneten Klassen abgebildet werden
- Code Redundanzen sollten, wenn möglich vermieden werden
- Achten Sie auf eindeutige Benennungen und Code Konventionen
- IO Ressourcen (Reader/Writer, Streams etc) sollten im Gut und Fehlerfall geschlossen werden.
- Achten Sie auf ein korrektes Exception-Handling

Allgemeine Beurteilungskriterien

Bei der Beurteilung der Umsetzung des Arbeitsauftrages in ein Software-Programm sind zwei Kriterien wichtig:

Funktionalität

Das Software-Programm soll die in der Angabe stehenden Anforderungen durch die Bearbeitung von möglichen Eingabedaten bzw. durch die Ausführung von Funktionen/Algorithmen lösen, sodass das gewünschte Endergebnis, welches in der jeweils geforderten Form samt den korrekten Daten erbracht werden soll, erzielt wird. *Hierbei wird die technische Umsetzung außer Acht gelassen. Es zählt nur der Vergleich des erbrachten Endresultates zu den gegebenen Anforderungen.*

Technische Umsetzung

Als Software-Programmierer ist es nicht nur wichtig, dass gewünschte Ergebnis zu erzielen. Es ist auch von essenzieller Bedeutung, dass die technische Umsetzung gewissen konzeptionellen und strukturellen Anforderungen entspricht. Der Code wird dadurch testbarer, übersichtlicher/lesbarer und auch bei Fehlerbehebungen wartbarer. Des Weiteren kann zum Beispiel durch den Einsatz von technischen Konstrukten (Schleifen, Entscheidungen, ...) möglicherweise auch der Verbrauch des benötigten Speicherbedarfs minimiert bzw. eine effizientere(schnellere) Ausführung des Programms erzielt werden.

Hierbei geht es nicht um das erzielte Endergebnis, sondern um die Analyse der technischen (Teil-)Umsetzung der Anforderungen.

Allgemeiner Kommentar

Den Teilnehmenden steht es frei zu wählen, welche Programmiersprache Sie zur Umsetzung der Anforderungen verwenden. Sie muss jedoch innerhalb der bisherigen Ausbildung enthalten gewesen sein.

Es ist den Teilnehmenden erlaubt, zur Erfüllung der Anforderungen das Internet bzw. eigene Unterlagen zu benutzen. Alle in der jeweiligen Programmiersprache mitgelieferten Bibliotheken/Konstrukte sowie externe Bibliotheken, die zur Erfüllung notwendig sind und nicht durch einen anderen Teilnehmenden erstellt wurden, sind erlaubt.