

c# Programmieren (c) HTL-Leonding

UpdateProduct-CSV

Lehrziele

- Programmargumente
- Dateien - Lesen/Schreiben/Umbenennen
- Lokalisierung: Dezimalpunkt als Punkt und nicht als Komma.
- Datentyp `decimal` für eine Währung

Aufgabenstellung

Das gesuchte Programm **UpdateProductCsv** soll in einer Csv Datei (diese enthält Artikel mit je einem Verkaufspreis) Änderungen vornehmen können. Dabei wird die gesamte Datei gelesen, im Speicher verändert und dann wieder geschrieben. Der Dateiname wird als erstes Argument übergeben. Das zweite Argument wird als Prozentsatz interpretiert, mit dem die Verkaufspreise angepasst werden sollen.

Programmablauf

Das Programm soll die folgenden Funktionalitäten aufweisen.

- Beim Programmstart wird geprüft, ob exakt zwei Argumente übergeben wurden.
- Das erste Argument muss ein gültiger Dateiname **mit** der Erweiterung CSV sein.
- Das zweite Argument muss in eine Kommazahl konvertiert werden können (Punkt als Dezimaltrennzeichen).
- Das Programm liest anschließend alle Artikel aus der CSV Datei in eine interne Liste (=Array).
- Alle Preise werden mit dem übergebenen Prozentsatz (zweites Argument) umgerechnet. Achten sie auf eine zweistellige Rundung.
- Die Artikelliste wird in eine neue Datei gespeichert:
 - Zuerst wird eine Datei mit dem Originalnamen und der Erweiterung `$$$` erstellt.
 - Die Originaldatei wird in `bak` umbenannt.
Eine eventuell vorhandene `bak` Datei wird zuvor gelöscht.
 - Als letzter Schritt wird die `$$$` in `csv` verschoben.
- Schreiben sie eine Methoden zum Konvertieren einer CSV Zeile. Mit **out** Parametern werden die Werte der Spalten an den Aufrufer zurück übergeben.
- Eine weitere zu implementierende Methode erstellt eine CSV Zeile. Die Spaltenwerte werden als Parameter übergeben.

Programmdesign

- Achten Sie auf die richtigen Datentypen. Eine Währung (eine Zahl mit 2 Nachkommastellen) sollte als `decimal` gespeichert werden.
- In der CSV Datei ist der Verkaufspreis mit einem Punkt als Dezimaltrennzeichen gespeichert.

Unittests

In dem bereitgestellten Programm-Template sind Unittests vorhanden. Implementieren Sie die Methoden so, dass **alle** Unittests erfolgreich ausgeführt werden können. Änderungen an den *Unittests* sind nicht erlaubt.

Bildschirmausgabe

Die Steuerung des Programms erfolgt über die Programm-Argumente. Eine Eingabe mit `Console.ReadLine()` wird daher **nicht** benötigt. Fehlermeldungen (z.B. falscher Dateiname) werden auf der Konsole ausgegeben.