Technische Beschreibung

Zweiradverwaltung

Inhaltsverzeichnis

[Verwaltung 3](#_Toc388119130)

[Class FileContents 3](#_Toc388119131)

[Defaultkonstruktor 3](#_Toc388119132)

[Methoden 3](#_Toc388119133)

[BaseImageHandler 4](#_Toc388119134)

[WPFcommunication 5](#_Toc388119135)

[WPFcommunications 5](#_Toc388119136)

[Pfadauswahl 5](#_Toc388119137)

[Defaultkonstruktor 5](#_Toc388119138)

[Hauptkonfiguration 5](#_Toc388119139)

[Defaultkonstruktor 5](#_Toc388119140)

[Strukturen 6](#_Toc388119141)

[FileStructure 6](#_Toc388119142)

[Fahrten 6](#_Toc388119143)

[HauptConfig 6](#_Toc388119144)

# Verwaltung

Hauptsächlich reine C# Klassen ohne oder mit wenig WPF Einfluss.

## Class FileContents

Diese Klasse beschreibt das Innere der Zweirad-Einträge.

Öffentlich verfügbar ist hierbei die Variable Data mit dem Dateninhalt „FileStructure“.

### Defaultkonstruktor

FileContents(string filer)

Filer beschreibt hierbei den Pfad inkl. Dateiname der zu öffnenden Datei. Hierbei wird versucht, aus dem Dateinamen die ID zu extrahieren. Ist das nicht möglich, wird die ID „-1“ vergeben.

### Methoden

void ReadFile()  
Liest die mit dem Defaultkonstruktor angegebene Datei ein und versucht diese in die Variable Data zu speichern.

void WriteFile()  
Schreibt die in Data definierte Struktur in die vom Defaultkonstruktor angegebene Datei.

void Poke()  
Ändert von der Data Struktur die Variable „LetzteBenutzung“ zum aktuellen Zeitpunkt mittels „DateTime.Now“.

void ExportFile(string path, bool image)  
path beschreibt den Standort, wohin die zu exportierende Datei geschrieben werden soll. Image gibt an, ob auch (falls vorhanden) das Bild mit exportiert werden soll.   
Die Daten werden dann in folgender Struktur gespeichert:

ID – Bezeichnung  
  
Servicestichtag: d/M/yyyy  
Kilometer: X km  
Verfügbar Ja/Nein  
  
--- Logbuch ---  
/ X km | d/MM/yyyy (VON) - d/MM/yyyy (BIS)  
~  
  
--- Bild ---  
Base64 encoded Image mit 64 Zeichen je Zeile

## BaseImageHandler

Diese Klasse hilft bei der Verarbeitung und Konvertierung von Bildern in verschiedene Datentypen.

Image Path2Image(string fileName)  
Konvertiert einen über fileName übergebene Datei in ein Bild von Typ „Image“.

string Image2Base64(Image image)  
Konvertiert ein Bild von Typ „Image“ in einen Base64 String.

Image Base642Image(string base64String)  
Konvertiert einen Base64 String in ein Bild von Typ „Image“.

BitmapImage Image2BitmapImage (Image image)  
Konvertiert ein Bild von Typ „Image“ in ein Bild von Typ „BitmapImage“.

# WPFcommunication

Allgemeine Kommunikationsklassen zwischen der WPF Oberfläche und der internen C# Programmierung.

## WPFcommunications

string ConfigurationData()  
Diese Methode öffnet ein Order Auswahlfenster, damit der Benutzer das gewünschte Verzeichnis auswählen kann. Kommt es zum Fehlerfall oder wird Abbrechen gedrückt, wird null zurückgegeben.

DataTable Fahrtenbuch(Fahrten[] data)  
Konvertiert ein Fahrten Element in ein DataTable Element mit den Inhalten „Von“, „Bis“ und „Kilometer“.

double ZKilometer(Fahrten[] data)  
Zählt alle in data.Kilometer übergebenen Kilometer zusammen und gibt diese als double Wert zurück.

DataTable Fahrzeuge(string pfad)  
Liest alle Dateien, die sich in einem Ordner befinden entsprechend der Suchmaske   
„Zweirad\_([0-9]+)\.bin“ aus. Zurückgegeben wird hierbei ein DataTable mit den Spalten „ID“, „Name“ und „Letzte Änderung“.

## Pfadauswahl

Vererbte Klasse von WPFcommunication.

Die Klasse besitzt eine Eigenschaft „Projektpfad“ dessen Wert öffentlich gelesen und nur intern geschrieben werden kann.

### Defaultkonstruktor

Hierbei wird die vererbte Methode „ConfigurationData()“ aufgerufen. Wenn der darin zurückgegebene Ordner existiert, wird dieser in die Eigenschaft „Projektpfad“ geschrieben.

## Hauptkonfiguration

Vererbte Klasse von MainWindow.

Die Eigenschaft „Einstellung“ kann öffentlich gelesen und auch geändert werden. Abbild von struct HauptConfig.  
Die Eigenschaft „Geladen“ kann öffentlich gelesen werden aber nur intern geändert werden.

## Defaultkonstruktor

Hauptkonfiguration(string prpfad)  
Liest die Konfigurationsdatei „configuration.bin“ von dem übergebenen Ordner der Variable „prpfad“.

void Write()  
Schreibt die Daten in der Variable „Einstellung“ in die Datei „configuration.bin“ in den zuvor übergebenen Ordner.

# Strukturen

## FileStructure

string Bezeichnung  
Bezeichnung des Zweirades.

int ID   
Identifikationsnummer des Zweirades

DateTime Servicetermin   
Zeitpunkt des geplanten Services.

int Kilometer   
Anzahl der bereits zurückgelegten Kilometer.

Fahrten[] Logbuch   
Array der aufgezeichneten Fahrten als Fahrten-Struktur.

DateTime LetzteBenutzung   
Zeitpunkt der letzten Benutzung der Datei.

string Bild   
Base64 String des Bildes

bool Verfügbar  
Ist das Zweirad verfügbar oder nicht verfügbar.

## Fahrten

DateTime Von  
Zeitpunkt, ab wann das Zweirad ausgeliehen wurde.

DateTime Bis

Zeitpunkt, bis wann das Zweirad ausgeliehen wurde.

## HauptConfig

int Einträge  
Anzahl der Zweiradeinträge A

DateTime LastUse  
Zeitpunkt der letzten Nutzung einer Niederlassung