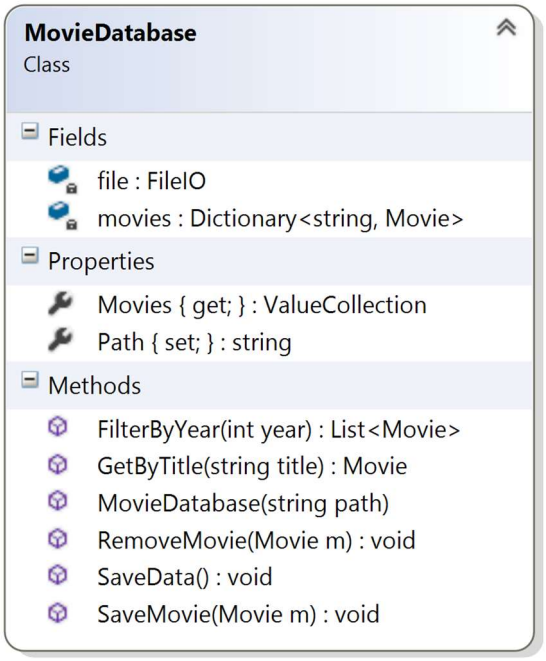
17_MovieCollection

Erstelle ein C#-Programm, mit dem man Kinofilme verwalten kann.

Die Klasse Movie speichert alle Daten eines Films und macht sie über Properties zugänglich. **Die Klasse Movie ist schon fertig implementiert** und kann sofort eingesetzt werden.



Die Klasse MovieDatabase verwaltet alle Filme mit einem Dictionary movies mit Title als Key.

Properties:

Movies: Liefert alle Filme in der MovieDatabase

Path: damit kann der Dateipfad nachträglich gesetzt werden.

Methoden:

FilterByYear: liefert eine Liste mit jenen Movies, die das Jahr year gesetzt haben.

GetByTitle: liefert zu einem title das Movie-Objekt oder null, falls es keines gibt.

MovieDatabase: Konstruktor, der den Pfad zur Standard-Datei mitbekommt.

RemoveMovie: enfernt den Movie m aus dem Dictionary

SaveData: speichert alle Movies in der Datei.

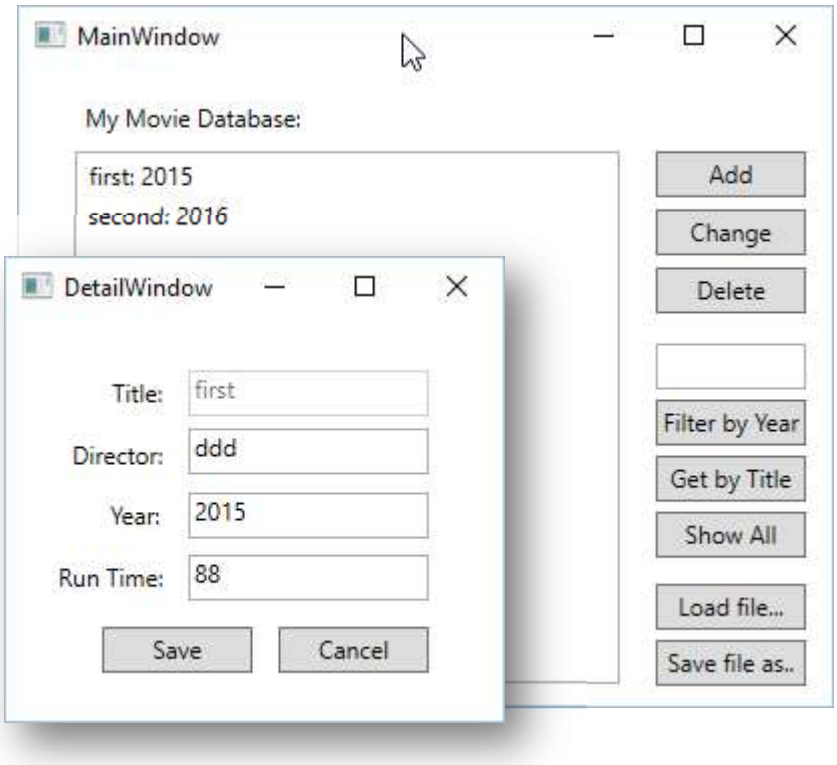
SaveMovie: speichert den Movie m im Dictionary. Wenn der Film mit dem Title bereits vorhanden ist, so soll er überschrieben werden.

Die Speicherung der Daten erfolgt mit der Klasse **FileIO**, die ebenso wie die Movie Klasse bereits **fertig vorhanden** ist.

GUI:

Die **XAML-Daten für die GUI** (MainWindow und DetailWindow) sind bereits **fertig vorhanden**!

Bei Start des Programms werden alle Movies aus der Standard-Datei geladen und angezeigt.



Der Benutzer klickt auf den Button Add und bekommt ein DetailWindow als Dialog angezeigt. In diesem Dialog trägt er alle Daten des Films ein und wenn wer auf den Button Save klickt, so wird der neue Movie in der MovieDatabase abgespeichert und in der Liste aller Movies angezeigt.

Beim Button Change erscheint ebenso das DetailWindow wie beim Button Add, nur die Eingabefelder mit den bestehenden Daten sind bereits befüllt und der Titel darf nicht geändert werden. Nach Klick auf den Button Save wird der bestehende Film in der Datenbank mit den neuen Daten überschrieben.

Beim Button Delete erhält der Benutzer eine Sicherheitsabfrage, ob er wirklich den in der Liste ausgewählten Film löschen möchte. Wenn er diese mit Yes bestätigt, wird der Film aus der MovieDatabase entfernt und in der Liste nicht mehr angezeigt.

Hinweis:

Die Sicherheitsabfrage für den Löschvorgang wird folgendermaßen erstellt und ausgewertet:

```
MessageBoxResult result = MessageBox.Show("Do you really want to delete " + movie.Title + " ?",
"Question", MessageBoxButton.YesNo, MessageBoxImage.Warning);

if (result == MessageBoxResult.Yes)
{
      // delete the movie
}
```

Beim Klick auf den Button Filter by Year wird aus der darüberliegenden Textbox das Jahr ausgelesen. Danach werden in der ListBox nur jene Filme angezeigt, die dieses Jahr eingetragen haben.

Beim Klick auf den Button Get by Title wird aus der darüberliegenden Textbox der Titel ausgelesen. Danach wird das Detailfenster mit dem Film zu diesem Titel geöffnet. Falls kein Film zum Titel vorhanden ist, wird stattdessen eine Fehlermeldung ausgegeben.

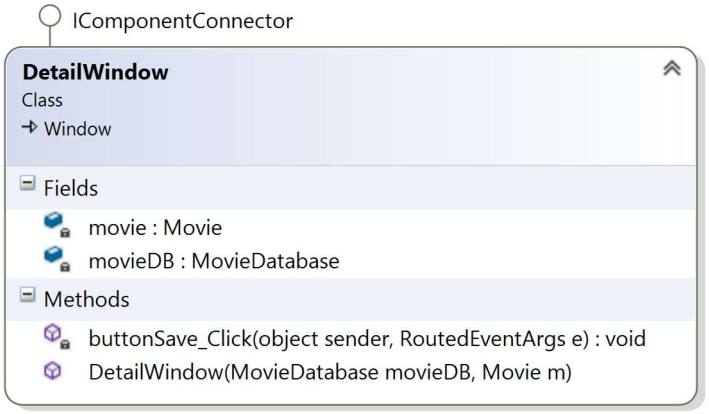
Beim Klick auf den Button Show All werden wieder alle Filme ungefiltert angezeigt.

Beim Klick auf den Button Load file … erscheint OpenFileDialog, mit dem die Datei, wo die Filme gespeichert sind, ausgewählt werden kann. Nach dem Schließen des Dialogs werden die Filme aus dieser Datei in der ListBox aufgelistet.

Beim Klick auf den Button Save file as … erscheint ein SaveFileDialog, mit dem die Datei ausgewählt werden kann, wo die Filme gespeichert werden sollen.

DetailWindow

Die Klasse DetailWindow soll folgendermaßen implementiert werden:



Beim Erzeugen eines neuen DetailWindows wird mit dem Konstruktor vom MainWindow die MovieDatabase sowie der anzuzeigende Film mitgegeben.

Bei einem neu zu erfassenden Film wird stattdessen als Movie null mitgegeben.

Somit kann im DetailWindow festgestellt werden, ob ein bestehender Film angezeigt und geändert werden soll, oder ob ein neuer Film erfasst und in die MovieDatabase hinzugefügt werden soll.

Gutes Gelingen!