

Hausarbeit

Anwendungsprogrammierung

WS 21/22

Vor- und Nachname: Cedrik Hoffmann

Matrikelnummer: 670339

Prüfungszeitraum: WS-2021/22-II oder SS-2022-I oder SS-2022-II

Einzelprojektnummer: 011

Gruppennummer: 017

Hinweise zur Verwendung dieser Vorlage: Sie können diese Vorlage verwenden oder mit einer Textverarbeitung Ihrer Wahl nachbauen. Ändern Sie auf keinen Fall die Formatierung! Schriftart (Liberations Serif oder TimesNewRoman), Schriftgröße, Abstände, Schriftfarbe, Seitenabstand, Einrückung etc. sind beizubehalten; oder im Falle eines "Nachbaus" auf identische Weise herzustellen. Auch dieser Hinweis darf nicht entfernt werden.

Inhaltsverzeichnis

1.	Einleitung.....	1
2.	Übersicht des Programms aus Nutzersicht.....	1
2.1	<i>Walkthrough.....</i>	<i>1</i>
2.2	<i>Ergänzende Hinweise</i>	<i>2</i>
3.	Übersicht über die Solution	2
3.1	<i>UML-Diagramm.....</i>	<i>3</i>
3.2	<i>Die View Schicht.....</i>	<i>4</i>
3.2.1	<i>Layout.....</i>	<i>4</i>
3.2.2	<i>Funktion</i>	<i>4</i>
3.3	<i>Die ViewModel Schicht.....</i>	<i>6</i>
3.3.1	<i>Verfügbare Properties.....</i>	<i>6</i>
3.3.2	<i>Funktion</i>	<i>7</i>
3.4	<i>Die Service Schicht.....</i>	<i>8</i>
3.4.1	<i>Layout.....</i>	<i>8</i>
3.4.2	<i>Funktion</i>	<i>9</i>

1. Einleitung

Auf dem bereits bekannten MVVM-Grundgerüst wird ein neues Feature eingebaut. Das Feature soll eine Druckfunktion der gespeicherten Bücher ermöglichen. Dazu wird ein neues Ui-Fenster erstellt. In der Ui kann der Benutzer eine Auswahl an Büchern in einer `ListView` auswählen. Das angeklickte Buch wird im Bereich Book Preview im Detail angezeigt. Mit dem Button Add wird das gewählte Buch zum Drucken markiert. Es können mehrere Bücher ausgewählt werden. Die ausgewählten Bücher können im Bereich „**Selected Books**“ eingesehen werden. Zudem hat der Benutzer die Möglichkeit, ein Buch aus der Liste wieder zu entfernen.

In der folgenden Dokumentation wird das Projekt zum Feature „PrintService“ genauer beschrieben. Es folgt eine Walkthrough aus Benutzerperspektive, sowie eine Übersicht über die Solution wo Funktions- und Designentscheidungen genauer erläutert werden.

2. Übersicht des Programms aus Nutzersicht

Wird das Programm gestartet, ladet der User auf der Startseite des Programmes. Siehe hierzu die gemeinsame Kurzanleitung. Möchte der Benutzer, die bereits angelegten Bücher in einer Liste ausdrücken, so bietet das Feature „PrintService“ die Möglichkeit, eine sortierte Liste der gewählten Bücher auszudrucken.



2.1 Walkthrough

Wird das `PrintServiceWindow` gestartet, sieht der Benutzer links oben in der Liste „**All Books**“ alle Bücher, welche in der Anwendung bereits angelegt wurden. Unter der Liste befinden sich zwei Buttons, um die gewählten Bücher zur Auswahl hinzuzufügen [ADD], oder das Buch aus

der Liste wieder zu entfernen [REMOVE]. Die ausgewählten Bücher, welche in der Liste „Selected Books“ zu sehen sind, werden anschließend in einer Liste ausgedruckt. Unter dieser Liste verbergen sich wiederum zwei Buttons. Zum einen, um den Druckvorgang zu starten [PRINT] und zum anderen, um das Fenster zu schließen [CLOSE], um wieder zur Startseite zu gelangen. Wählt der Benutzer ein Buch in der Liste „All Books“ aus, wird die Vorschau mit den entsprechenden Metadaten des Buches angepasst, welches angeklickt wurde ([siehe Book Preview](#)). Hat der Benutzer die Bücher, welche ausgedruckt werden sollen, der Liste „Selected Books“ hinzugefügt, kann der Benutzer über den Button [PRINT] den Druckvorgang starten. Es öffnet sich nun das Standard Windows Druckerfenster, wo der Benutzer seinen Drucker auswählen kann und somit die Liste der Bücher ausgedruckt wird.

2.2 Ergänzende Hinweise

Der Benutzer hat die Möglichkeit, mittels STGR + <KLICK AUF BUCH> mehrere Bücher in der Liste „All Books“ auszuwählen. Somit können mehrere Bücher gleichzeitig zur „Selected Book“ Liste hinzugefügt werden ([siehe Add Books](#)). Es muss zwingend ≥ 1 Buch in der Liste „Selected Books“ beim Starten des Druckvorgangs vorhanden sein, ansonsten erscheint eine Meldung, dass nicht genug Bücher in der Liste vorhanden sind und der Druckvorgang wird abgebrochen.

3. Übersicht über die Solution

Auf die Solution, welche in der Vorlesung erstellt wurde, wurden folgende Dateien / Klassen / Klassenbibliotheken hinzugefügt:

Logic

```
|-- Logig.Ui
|   |-- MessageBusMessages
|   |   |--+ OpenPrintServiceWindowMessage.cs
|   |
|   |-- ViewModel
|   |   |--+ PrintServiceWindowViewModel.cs
```

Ui

```
|-- Ui.Desktop
|   |--+ PrintServiceWindow.xaml
|   |--+ PrintServiceWindow.xaml.cs
```

Services

```
|--+ Services.PrintService
|   |--+ PrintBookService.cs
```

Listing 1 - Solution Explorer Ansicht

In der View Schicht:

Hier wurde in Ui.Desktop ein neues WPF Fenster mit den Namen PrintServiceWindow angelegt. Daraus folgen die Dateien PrintServiceWindow.xaml und PrintServiceWindow.xaml.cs. PrintServiceWindow.xaml beschreibt die XAML-Datei für das WPF Fenster bzw. der UI. PrintServiceWindow.xaml.cs beschreibt die *Code-Behind* und bleibt im MVVM-Pattern zwingend leer, da die Logik der UI im ViewModel erstellt wird.

In der ViewModel Schicht:

Jede View hat auch ein ViewModel, somit wurde in dem Projekt `Logic.Ui` im Ordner `ViewModel` die Klasse bzw. das ViewModel `PrintServiceWindowViewModel.cs` erstellt. Hierhinter verbirgt sich die Logik hinter der View `PrintServiceWindow.xaml`.

Zudem wurde die Klasse `OpenPrintServiceWindow.cs` erstellt, diese dient für den `MessageBus`, damit aus dem Hauptfenster das Fenster der Druckfunktion geöffnet werden kann.

Als Service:

Da die Druckfunktion als Service zu sehen ist, wurde im Ordner `Services` eine neue Klassenbibliothek `Service.PrintService` angelegt. In dieser Klassenbibliothek befindet sich die Klasse `PrintBookService.cs`, welche die Logik des Service beinhaltet. Hier wurde der Namespace `De.HsFlensburg.ClientApp011.Services.PrintService` verwendet.

3.1 UML-Diagramm

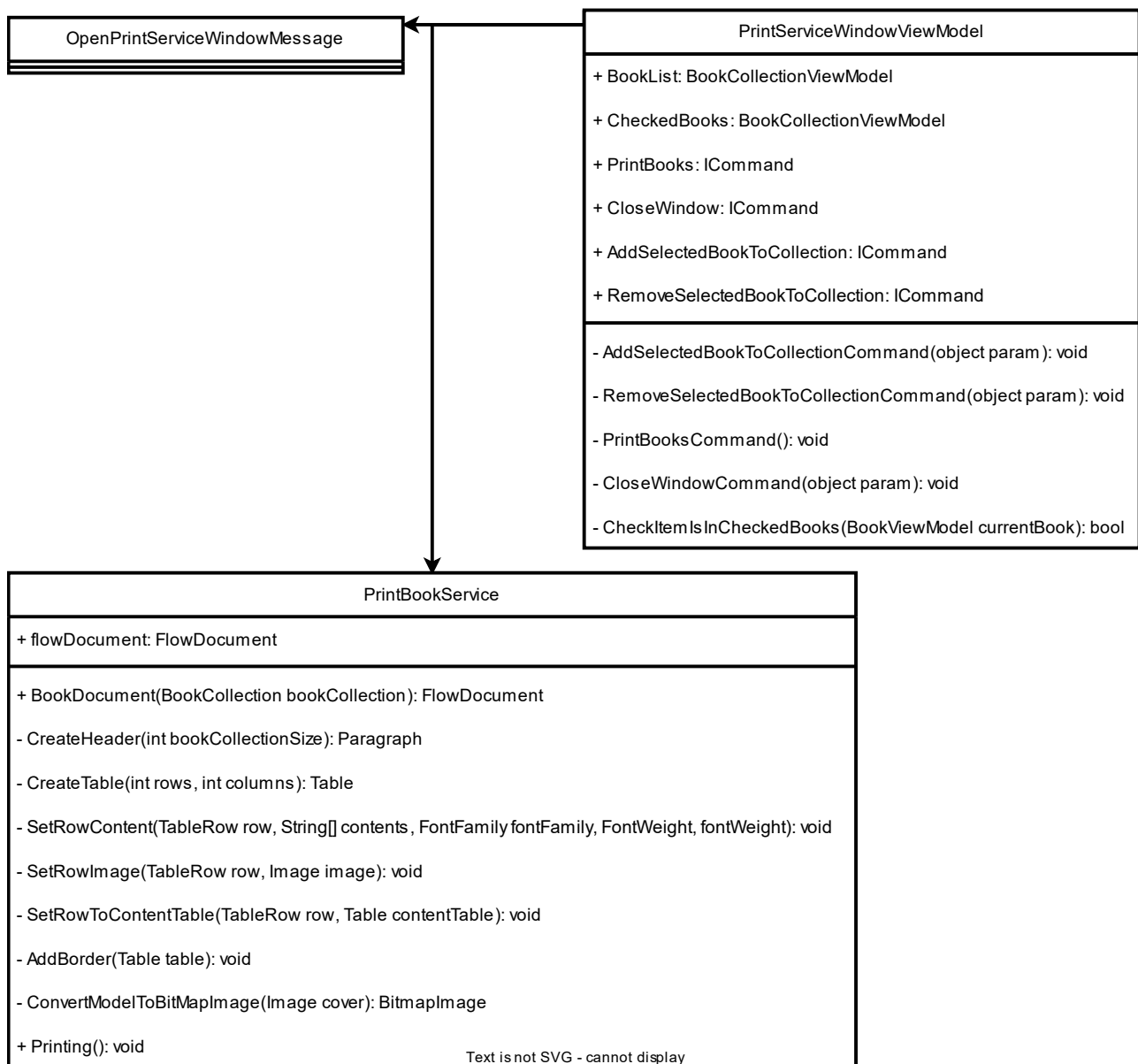


Abbildung 1 - UML Diagramm

3.2 Die View Schicht

Die View der Druckfunktion wurde mittels XAML Definiert. Das Layout teilt sich in zwei Spalten auf (BookControl links und BookPreview rechts) und wurde mittels Grid realisiert.

3.2.1 Layout

Layout BookControl:

Die **BookControl** besteht aus den UI Elementen, womit der User interagieren kann. In diesem Fall die Listen „**All Books**“ und „**Selected Books**“, sowie die einzelnen Buttons. Das Layout besteht aus einer Spalte und sechs Reihen. Die Reihen werden jeweils den Komponenten zugeteilt, siehe Abbildung Layout Row.

Layout BookPreview:

Die **BookPreview** dient nur zur Vorschau der Daten des Buches, welches vom User in der BookControl ausgewählt wird. Dieses Layout besteht aus 2 Spalten mit 3 Reihen. Die entsprechenden Komponenten wurden hier auch dem Layout zugeteilt, siehe Abbildung Layout Row.

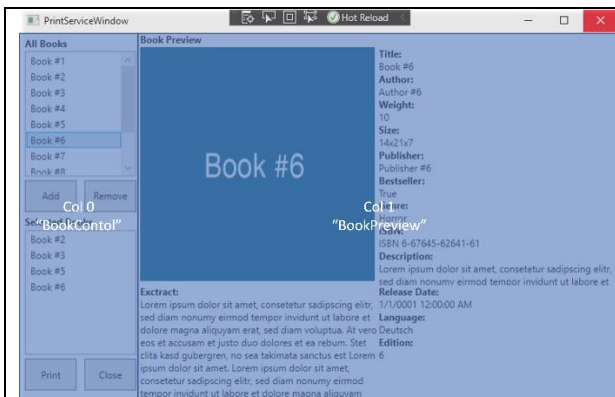


Abbildung 2 - Globales Layout

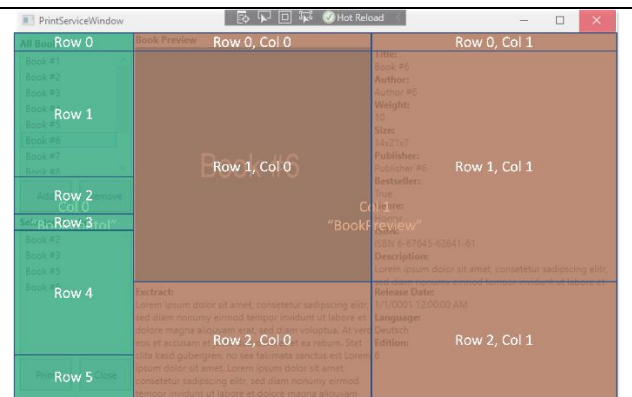


Abbildung 3 - Layout detailliert

3.2.2 Funktion

Die Bücher in der Liste „**All Books**“ sind in einer BookCollectionViewModel im ViewModel gespeichert. Beim Starten des UI-Fensters, wird die BookCollectionViewModel mit den angelegten Büchern an das PrintServiceWindowViewModel im ViewModelLocator übergeben. Da die UI übers *Binding* Zugriff auf das PrintServiceWindowViewModel hat, kann die UI mittels einer ListView die Daten von der Collection in einer Liste anzeigen.

Die Liste „**Selected Books**“ ist ähnlich wie die Liste „**All Books**“ aufgebaut. Diese besteht auch aus einer Collection des Typen BookCollectionViewModel. Wird über den Button [ADD] ein oder mehrere Bücher ausgewählt, werden diese Bücher im ViewModel der Collection CheckedBooks hinzugefügt, welche in der View dann angezeigt wird. Dazu übergibt die View mittels des RelayCommand mit Parametern die Bücher, welche vom Benutzer selektiert wurden, an das ViewModel. Diese werden in der ListView als

System.Windows.Controls.SelectedItemCollection an das ViewModel gesendet. Ähnliche Logik verbirgt sich auch hinter den Button [REMOVE], nur das hier entsprechend die übergebenen Bücher der Collection CheckedBooks entfernt werden. Die genauere Logik hierhinter wird in dem Kapitel 3.3 genauer erläutert.

```

<Button Content="Add" Margin="0,5,5,5"
    CommandParameter="{Binding ElementName=BookListBox, Path=SelectedItem}"
    Command="{Binding AddSelectedItemToCollection}"/>
<Button Content="Remove" Margin="5,5,0,5"
    CommandParameter="{Binding ElementName=BookListBox, Path=SelectedItem}"
    Command="{Binding RemoveSelectedItemToCollection}"/>

```

Listing 2 - Button Funktion in XAML

Die Liste „**Selected Books**“ zeigt wie schon erwähnt, die Bücher der Collection `CheckedBooks` an. Diese `ListView` hat die Eigenschaft, dass diese nicht angeklickt werden kann. Diese Eigenschaft wurde mittels des Parameters `<ListView.ItemContainerStyle>` deaktiviert.

```

<ListView Grid.Row="4" ItemsSource="{Binding CheckedBooks}">
    <ListView.ItemContainerStyle>
        <Style TargetType="ListViewItem">
            <Setter Property="Focusable" Value="False"/>
        </Style>
    </ListView.ItemContainerStyle>
    ...
</ListView>

```

Listing 3 - ListView Focusable deaktiviert

Der Bereich der `BookPreview` besteht größtenteils aus der Komponente `<TextBlock>`, welche den Wert des Buches ausgibt, welches aktuell vom Benutzer in der Liste „**All Books**“ angeklickt wurde. Dazu werden die `TextBlock` Komponenten über das Binding an die `ListView` gebunden. Wird nun ein anderes Buch in der `ListView` angeklickt, passt sich die Preview zu dem angeklickten Buch an.

```

<TextBlock FontWeight="Bold" Text="Title: "/>
<TextBlock Text="{Binding ElementName=BookListBox,
    Path=SelectedItem.Title}"/>

<TextBlock FontWeight="Bold" Text="Author: "/>
<TextBlock Text="{Binding ElementName=BookListBox,
    Path=SelectedItem.Author}"/>

<TextBlock FontWeight="Bold" Text="Weight: "/>
<TextBlock Text="{Binding ElementName=BookListBox,
    Path=SelectedItem.Weight}"/>

```

Listing 4 - TextBlock Funktion XAML

3.3 Die ViewModel Schicht

Das *ViewModel* beinhaltet die Logik der UI. Die Klasse *PrintServiceWindowViewModel* ist somit das zugehörige ViewModel zur View *PrintServiceWindow*.

Im ViewModel muss folgende Logik enthalten sein:

1. User fügt ein oder mehrere Bücher der Liste „**Selected Books**“ hinzu.
2. User entfernt ein oder mehrere Bücher aus der Liste „Selected Books“.
3. User möchte die ausgewählten Bücher ausdrucken.
4. User schließt das Fenster.

3.3.1 Verfügbare Properties

Das ViewModel enthält mehrere Properties, wo die View drauf Zugriff hat.

Property	Funktion	Typ
BookList	Enthält die bereits angelegten Bücher der Anwendung.	BookCollectionViewModel
CheckedBooks	Beinhaltet die Bücher, welche der Benutzer ausdrucken möchte.	BookCollectionViewModel
PrintBooks	RelayCommand, welches ausgeführt wird, wenn der Benutzer den Button [PRINT] in der UI auswählt. Dies startet den Druckvorgang.	ICommand
CloseWindow	RelayCommand, welches ausgeführt wird, wenn der Benutzer den Button [CLOSE] in der UI auswählt. Schließt das aktuelle Fenster.	ICommand
AddSelectedBookToCollection	RelayCommand, welches ausgeführt wird, wenn der Benutzer den Button [ADD] in der UI auswählt. Fügt der Collection <i>CheckedBooks</i> die übergebenden Bücher der View hinzu.	ICommand
RemoveSelectedBookToCollection	RelayCommand, welches ausgeführt wird, wenn der Benutzer den Button [REMOVE] in der UI auswählt. Entfernt die übergebenden Bücher von der View in der Collection <i>CheckedBooks</i> .	ICommand

3.3.2 Funktion

Konstruktor:

wird das Fenster in der Ui geöffnet, wird über den ViewModelLocator die bereits angelegten Bücher, welche sich im ViewModelLocator in der Property TheBookCollectionViewModel befinden, über den Konstruktor an das ViewModel übergeben. Diese Collection wird dann schließlich beim Konstruktor Aufruf der Property BookList zugewiesen. Neben dieser Zuweisung werden den ICommand Properties mittels RelayCommand den entsprechenden Methoden zugeteilt.

User fügt ein oder mehrere Bücher der Liste „Selected Books“ hinzu:

in diesem Moment wird von der View eine Collection vom Typen SelectedItemCollection an das ViewModel geschickt. Die Methode AddSelectedBookToCollectionCommand(object param) wird dabei aufgerufen. Die Methode castet den Inhalt der Collection zum Typen BookViewModel. Im Anschluss werden die Bücher, welche sich in der Collection befinden, der Collection CheckedBooks hinzugefügt. In diesem Zuge wird zudem überprüft, ob sich bereits ein Buch in der Collection CheckedBooks befindet. Dies wird mit der Hilfsmethode CheckItemIsInCheckedBooks(BookViewModel currentBook) überprüft.

```
// Command to add selected book to BookCollection "CheckedBooks"
private void AddSelectedBookToCollectionCommand(object param) {
    // Save System.Windows.Controls.SelectedItemCollection
    // to IList and Cast to List<BookViewModel>
    System.Collections.IList items = (System.Collections.IList)param;
    var collection = items.Cast<BookViewModel>();
    foreach (BookViewModel book in collection) {
        if(!CheckItemIsInCheckedBooks(book)) {
            CheckedBooks.Add(book);
        }
    }
}

// Helper Function to check if book is present in collection
private bool CheckItemIsInCheckedBooks(BookViewModel currentBook) {
    foreach(BookViewModel book in CheckedBooks) {
        if(book.Equals(currentBook)) {
            return true;
        }
    }
    return false;
}
```

Listing 5 - ViewModel Buch wird hinzugefügt

User entfernt ein oder mehrere Bücher aus der Liste „Selected Books“:

wählt der User ein Buch aus, welches aus der Liste entfernt werden soll, ist die Logik ähnlich wie das hinzufügen eines Buches. Die View schickt auch hier eine Collection des Typen SelectedItemCollection. Auch hier wird der Inhalt der Collection zu BookViewModel gecastet. Anschließend werden die Bücher in der Collection CheckedBooks über die Methode Remove() entfernt.

User möchte die ausgewählten Bücher ausdrucken:

Ist die Liste „Selected Books“ gefüllt und der User startet den Druckvorgang, wird die Methode `PrintBooksCommand()` ausgeführt. Hier ist die Besonderheit, dass überprüft wird, ob sich mindestens ein Buch in der Collection `CheckedBooks` befindet. Ist dies der Fall, wird die Collection an das Objekt `PrintBookService` übergeben, welches die Daten der Collection in ein `FlowDocument` konvertiert. (Näheres dazu im Kapitel 3.4). Befindet sich kein Buch in der Collection, wird eine `MessageBox` erstellt, welche dem User mitteilt, dass sich mindestens ein Buch in der Liste befinden muss.

User schließt das Fenster:

Schließt der User über den Button **[CLOSE]** das `PrintService` Fenster, wird die Methode `CloseWindowCommand(object param)` ausgeführt. Diese ist analog zu der Methode, welche im Skript vorgestellt wurde. Allerdings wird hier noch vor dem Schließen die Collection `CheckedBooks` mittels der Methode `Clear()` geleert, damit diese beim neu Aufruf des Fensters leer ist.

3.4 Die Service Schicht

Das schlussendliche Feature „PrintService“ verbirgt sich als Klassenbibliothek im Ordner `Services`. In der Klasse `PrintBookService.cs` wird das `FlowDocument` erstellt, welches anschließend an den Drucker gesendet wird. `FlowDocument` bietet eine gute Möglichkeit, ein dynamisches Dokument zu erstellen, welches sich dem Text entsprechend anpasst.

Print BookList of 6 Books

Book #1	Title:	Author:	Genre:	Preis:
	Book #1	Author #1	Horror	2,3
	Kurzbeschreibung:			
	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At			
Book #2	Title:	Author:	Genre:	Preis:
	Book #2	Author #2	Horror	2,3
	Kurzbeschreibung:			
	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At			

Abbildung 4 - Auszug eines Druckes

3.4.1 Layout

Das Layout besteht aus zwei Tabellen. Die erste Tabelle (Main Table), besteht aus einer Reihe und zwei Spalten. In der ersten Spalte befindet sich das Cover Image vom Buch. In der zweiten Spalte befindet sich eine weitere Tabelle (Content Table). Diese Tabelle wird dazu verwendet, die Metadaten des Buches strukturiert anzeigen zu können. Die Content Table besteht dabei aus 4 Reihen und 4 Spalten, wo sich die jeweiligen Überschriften wie Title, Author, Genre,

Preis und Kurzbeschreibung, sowie die entsprechenden Attribute beinhaltet. Die Reihen Kurzbeschreibung und das dazugehörige Attribut erstrecken sich über die 4 Spalten.

Book #2	Title:	Author:	Genre:	Preis:
	Book #2	Author #2	Horror	2,3
	Kurzbeschreibung:			
	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At			

Abbildung 5 - FlowDocument Layout

Die Überschrift der Attribute ist in der Schriftart Arial und Fett. Die jeweiligen Attribute in der Schriftart Consolas.

3.4.2 Funktion

Im ViewModel wird die Liste mit den Büchern, welche ausgedruckt werden sollen, an den Konstruktor des Objektes PrintBookService übergeben. Zudem wird im Konstruktor der Klassenvariable flowDocument, das entsprechende zugehörige Objekt FlowDocument() zugewiesen. Anschließend wird die Methode BookDokument(BookCollection bookCollection) aufgerufen. Diese Methode ist dafür zuständig, die Daten aus der Collection in ein FlowDocument umzuwandeln. Die Methode iteriert durch die Collection und legt für jeden Eintrag bzw. jedes Buch eine Tabelle an, welche dann dem FlowDocument hinzugefügt wird. Durch das Refactoring wurden mehrere Hilfsmethoden entwickelt, um den Code zu strukturieren.

Eine der wichtigsten Hilfsmethode ist hier SetRowContent(). Diese Methode befüllt die jeweilige Reihe der Content Table. Als Parameter erhält diese Methode die aktuelle TableRow, den Content, die FontFamily sowie die FontWeights.

```
// First Column fill with Headers
String[] headers = { "Title:", "Author:", "Genre:", "Preis:" };
SetRowContent(contentTable.RowGroups[0].Rows[0], headers, arial, FontWeights.Bold);
// Second Column fill with Content
String[] content = { book.Title, book.Author, book.Genre.ToString(), book.Price.ToString() };
SetRowContent(contentTable.RowGroups[0].Rows[1], content, consolas, FontWeights.Normal);
```

Listing 6 - Auszug aus Code, Tabelle wird gefüllt

Mit der Methode Printing() wird nun das Windows eigene Druckfenster geöffnet und das FlowDocument wird an den Drucker übergeben, welche der User über das PrintDialog ausgewählt hat.