

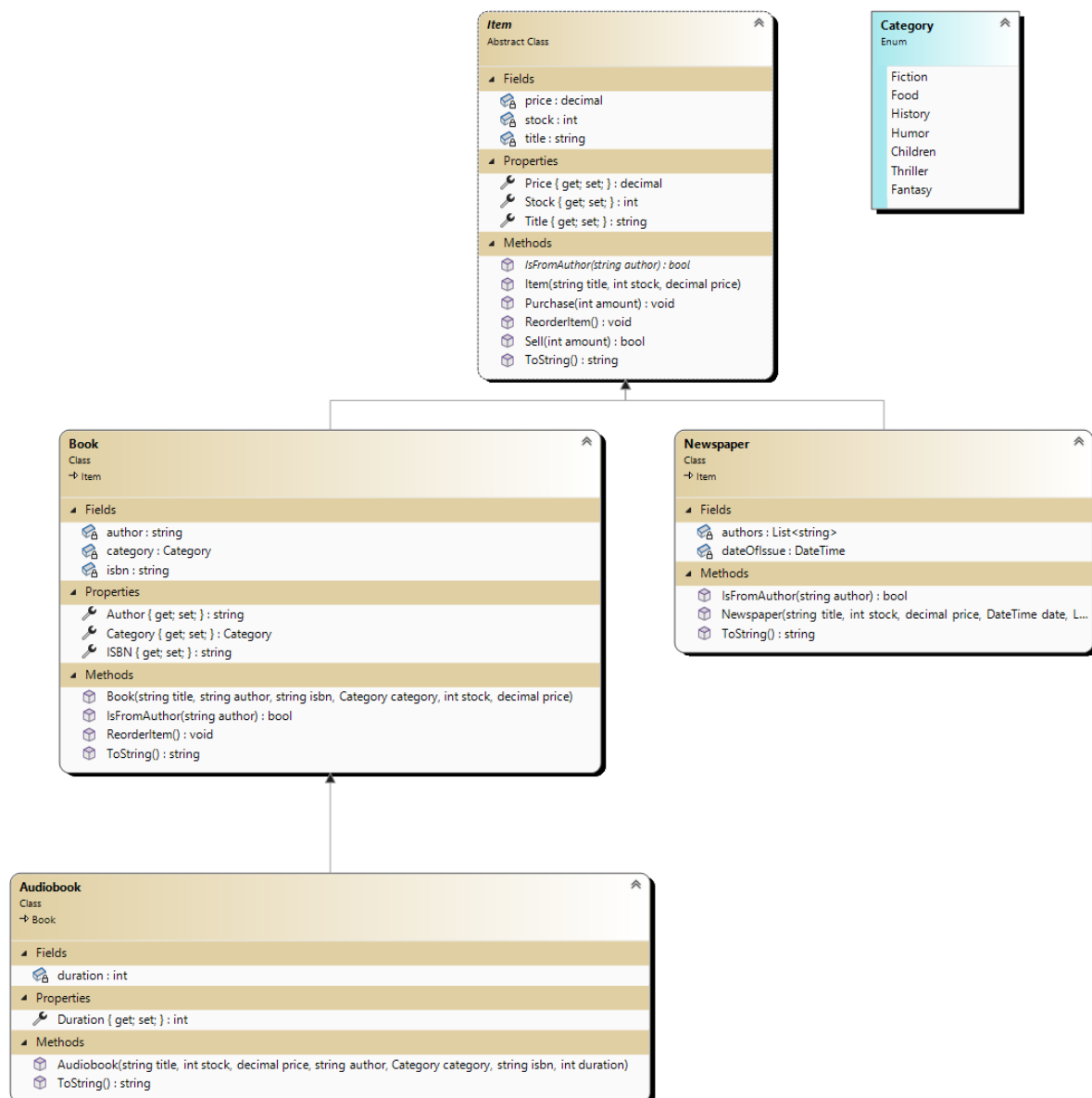
Lab – Ein objekt-orientiertes Buchgeschäft

In dieser Übung erstellen wir eine .Net Core (!) Klassenbibliothek für ein Buchgeschäft. Danach werden wir diese in einer WPF-Anwendung verwenden.

Klassenbibliothek

Erstelle das unten gezeigte Klassendiagramm in der Klassenbibliothek BookStoreLibrary. Achte dabei auf die folgenden Fragen:

- Welche Sichtbarkeit (`private`, `protected`, `public`) haben die Felder, Properties, Methoden der einzelnen Klassen? Schränke die Sichtbarkeit so stark wie möglich ein.



Klasse Item

Jedes *Item* in einer Buchhandlung hat einen Preis (**Price** - darf nicht negativ sein), einen Titel (**Title** - darf nicht leer sein) und einen aktuellen Lagerbestand (**Stock** - darf nicht negativ sein). Wirf die passende *Exception* bei ungültigen Werten. Alle diese Eigenschaften können von außen nicht verändert werden.

- **Purchase** – bedeutet den Zukauf von Items. Dabei erhöht sich der Lagerbestand (stock) um die gekaufte Menge.
- **Sell** – bedeutet den Verkauf von Items. Daher verringert sich der Lagerbestand um die als Parameter mitgegebenen Menge. Die Methode liefert *false*, wenn die gewünschte Menge nicht im Lager ist, ansonsten *true*.
- **ReorderItem** – ist eine Methode für die automatische Nachbestellung von Items. Beim Aufruf dieser Methode wird der Lagerbestand überprüft und wenn er kleiner 20 ist, werden 50 Stück vom Item „nachbestellt“, d.h. der Lagerbestand erhöht sich um 50.
- **IsFromAuthor** – ist eine Methode für die Suche nach Items von einem bestimmten Autor. Da Autor aber keine Eigenschaft in dieser Klasse ist, kann diese Methode hier nicht implementiert werden.
- **ToString** – liefert die String-Darstellung eines Items, dem Screenshot (siehe weiter unten) entsprechend.

Klasse Book

Diese Klasse erbt von Item und hat zusätzlich einen Autor (*author* – darf nicht leer sein), gehört zu einer Kategorie (*category* – vom Typ *Category*) und hat auch eine Isbn-Nummer (*isbn* – darf nicht leer sein).

- **IsFromAuthor** – liefert *true*, wenn der Name des Buchautors dem gesuchten Namen (Parameter) entspricht, ansonsten *false*.
- **ReorderItem** – die geerbte Methode für die Automatische Nachbestellung soll überschrieben werden. Da von Büchern ein nicht so hoher Lagerbestand gehalten wird, soll erst bei einem Lagerbestand kleiner 10 20 Stück „nachbestellt“ werden.
- **ToString** – ergänzt die geerbte Methode entsprechend dem Screenshot (siehe weiter unten).

Klasse Audiobook

Diese Klasse erbt von Book, hat zusätzlich eine Abspieldauer (*duration* – darf nicht negativ sein), die von außen nicht verändert werden darf.

- **ToString** – ergänzt die geerbte Methode entsprechend dem Screenshot (siehe weiter unten).

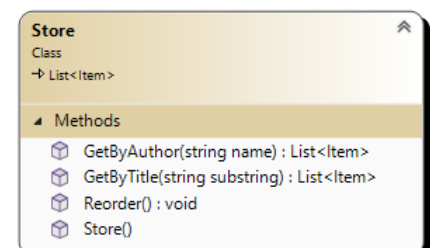
Klasse Newspaper

Diese Klasse erbt von Item und hat zwei zusätzliche Felder. Eine Liste, in dem die Namen der Autoren (*authors* – darf nicht leer sein) gespeichert werden, die einen Artikel veröffentlicht haben, und das Datum der Ausgabe (*dateOfIssue*).

- **IsFromAuthor** – liefert *true*, wenn der gesuchte Name (Parameter) in der Liste der Autoren enthalten ist, ansonsten *false*.
- **ToString** – ergänzt die geerbte Methode entsprechend dem Screenshot (siehe weiter unten).

Klasse Store

Um die diversen Bücher und Zeitschriften im Buchladen speichern zu können, brauchen wir eine Liste von Item Objekten, in der wir nach verschiedenen Kriterien suchen können. Um die WPF-Anwendung auch von dieser Logik zu befreien, erstellen wir für diese Funktionalität eine eigene Klasse **Store**, die von **List<Item>** erbt. Mit dieser einfachen Methode können wir die GUI in der WPF-Anwendung deutlich vereinfachen.

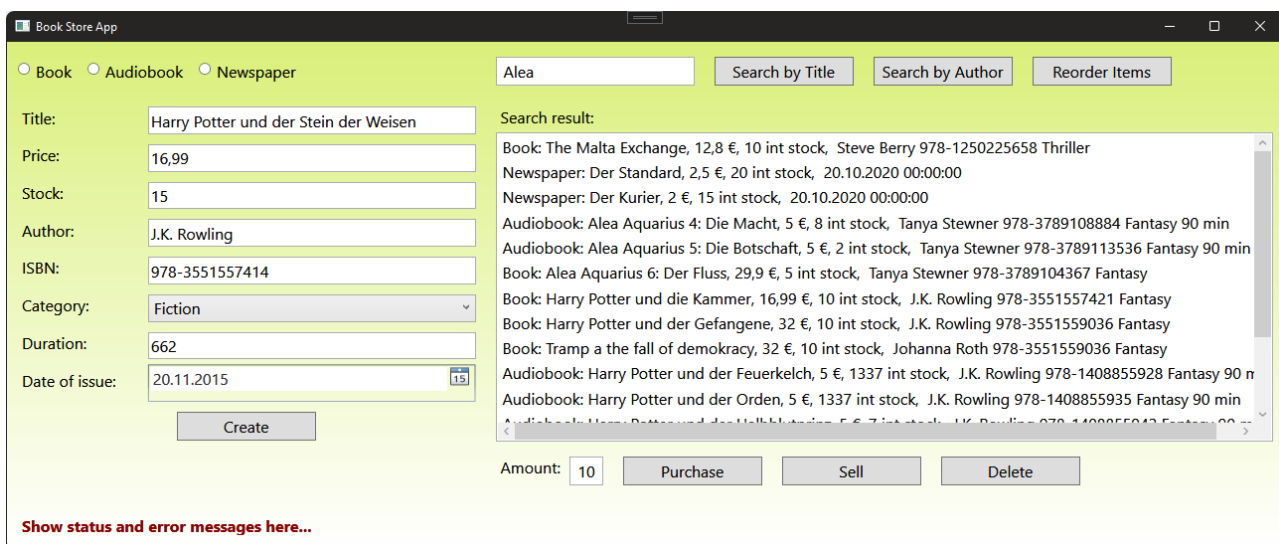


- **Konstruktor** – hier wird die Liste mit Hilfe des Code-Snipptes mit Testdaten befüllt.
- **GetByAuthor** – liefert eine Liste jener Items, die die Arbeit eines gesuchten Autors enthalten (*IsFromAuthor*).
- **GetByTitle** – liefert eine Liste jener Items, die im Titel das gesuchte Wort enthalten.
- **Reorder** – bestellt jedes Item nach, falls notwendig ist (*ReorderItem*).

WPF-Anwendung

Nachdem wir sehr viel Logik in der Klassenbibliothek implementiert haben, müssen wir in der WPF-Anwendung diese verwenden.

Gegeben ist die folgende GUI. Erstelle als erstes ein Store-Objekt, dessen Daten in dem Fenster verwendet werden können. Zeige beim Starten der Anwendung alle Items des Store-Objekts an.



Neue Einträge erstellen

- **comboBoxCategory** – diese zeigt die Kategorien an, die beim Anlegen eines neuen Buch- bzw. Audiobuch-Objektes ausgewählt werden können.

Fülle die ComboBox für die Kategorie mit Werten des Aufzählungstyps *Category*, indem du dir mittels `Enum.GetNames(typeof(Category))` ein Array von Strings daraus erstellst und diese in der ComboBox anzeigst. Setze die ausgewählte Kategorie auf den ersten Wert aus dem Array.

- **buttonCreate** – Abhängig von der ausgewählten Artikelart soll also ein Book, ein AudioBook oder ein Newspaper mit den eingegebenen Werten erzeugt werden. Achte darauf, dass bei ungültigen Werten die Logikklassen eine *Exception* werfen können, die hier abgefangen werden muss.

Die Property `SelectedDate` des **Date Pickers** ist ein nullable-Typ (`DateTime?`). Du musst hier überprüfen, ob der ausgewählte Wert nicht null ist, bevor du mit der Value Property den eingegebenen Wert auslesen kannst.

Du musst auch noch das ausgewählte Element (einen String) der **ComboBox** in einen Wert vom Typ *Category* umwandeln. Dazu kannst du den folgenden Ausdruck verwenden:

```
Category cat = Enum.Parse<Category>((string)cbCategory.SelectedItem);
```

Natürlich musst du auch hier überprüfen, ob überhaupt etwas ausgewählt wurde.

Teste deinen Code, indem du unterschiedliche Artikel hinzufügst.

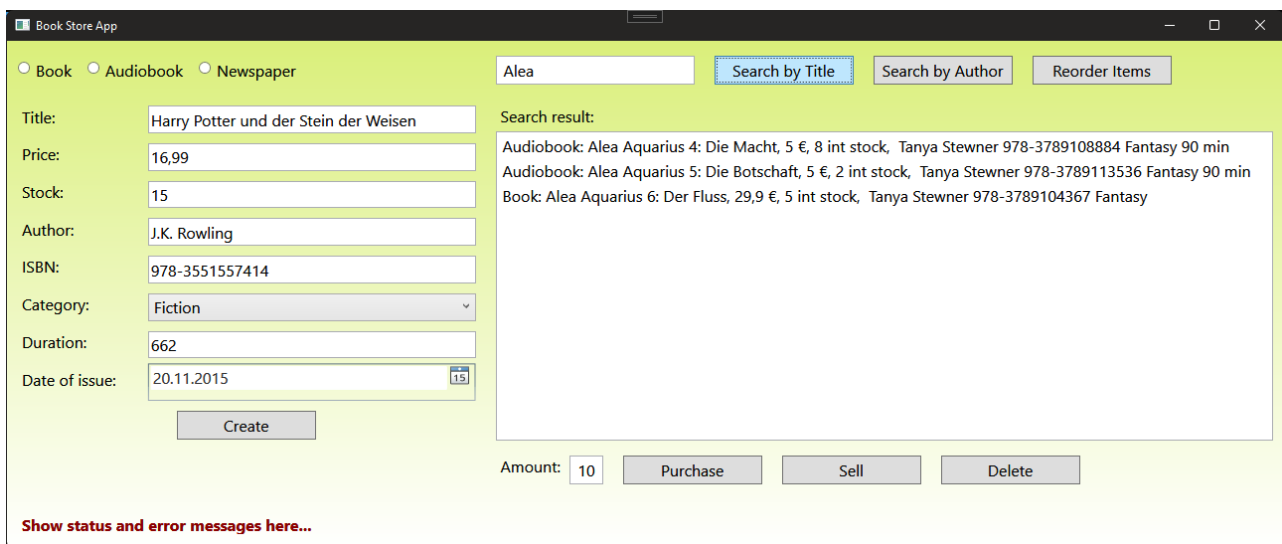
Die Funktionalität für die Purchase, Sell, Delete Buttons

- **buttonDelete** – Überprüfe, ob ein Eintrag in der ListBox ausgewählt ist und – wenn ja – lösche das entsprechende Element aus dem Store.
- **buttonPurchase, buttonSell** - beide überprüfen wieder, ob ein Eintrag ausgewählt wurde und – wenn ja – rufen die entsprechenden Methoden (Purchase/Sell) des Eintrags auf. Wenn nichts ausgewählt wurde oder der Rückgabewert der Methoden false ist, dann hat es nicht geklappt und eine Fehlermeldung soll ausgegeben werden.

Natürlich muss hier vor dem Aufruf auch überprüft werden, ob ein Wert in die Textbox für die Anzahl (engl. amount) eingegeben wurde; sonst gibt es wieder eine Fehlermeldung.

Die Suche und die Nachbestellung von Items

- **buttonSearchByTitle** –ruft die entsprechende Methode des Store-Objekts auf und zeigt das Ergebnis der Suche in der ListBox an.



Book Store App

Book Audiobook Newspaper

Title: Harry Potter und der Stein der Weisen

Price: 16,99

Stock: 15

Author: J.K. Rowling

ISBN: 978-3551557414

Category: Fiction

Duration: 662

Date of issue: 20.11.2015

Create

Search by Title Search by Author Reorder Items

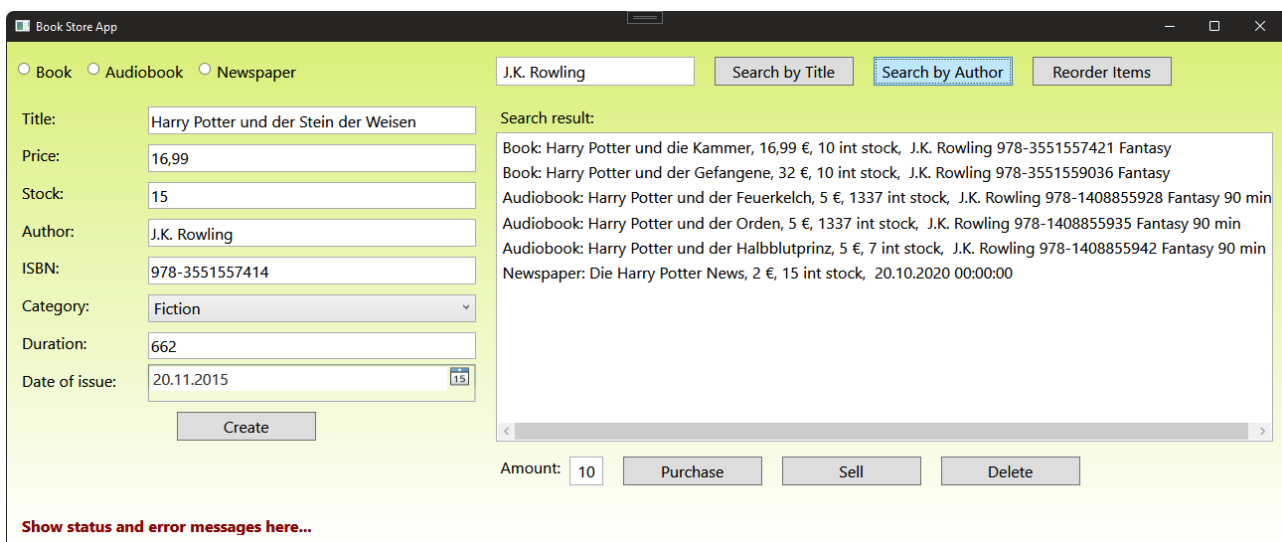
Search result:

Audiobook: Alea Aquarius 4: Die Macht, 5 €, 8 int stock, Tanya Stewner 978-3789108884 Fantasy 90 min
 Audiobook: Alea Aquarius 5: Die Botschaft, 5 €, 2 int stock, Tanya Stewner 978-3789113536 Fantasy 90 min
 Book: Alea Aquarius 6: Der Fluss, 29,9 €, 5 int stock, Tanya Stewner 978-3789104367 Fantasy

Amount: 10 Purchase Sell Delete

Show status and error messages here...

- **buttonSearchByAuthor** - ruft die entsprechende Methode des Store-Objekts auf und zeigt das Ergebnis der Suche in der ListBox an.



Book Store App

Book Audiobook Newspaper

Title: Harry Potter und der Stein der Weisen

Price: 16,99

Stock: 15

Author: J.K. Rowling

ISBN: 978-3551557414

Category: Fiction

Duration: 662

Date of issue: 20.11.2015

Create

J.K. Rowling Search by Title Search by Author Reorder Items

Search result:

Book: Harry Potter und die Kammer, 16,99 €, 10 int stock, J.K. Rowling 978-3551557421 Fantasy
 Book: Harry Potter und der Gefangene, 32 €, 10 int stock, J.K. Rowling 978-3551559036 Fantasy
 Audiobook: Harry Potter und der Feuerkelch, 5 €, 1337 int stock, J.K. Rowling 978-1408855928 Fantasy 90 min
 Audiobook: Harry Potter und der Orden, 5 €, 1337 int stock, J.K. Rowling 978-1408855935 Fantasy 90 min
 Audiobook: Harry Potter und der Halbblutprinz, 5 €, 7 int stock, J.K. Rowling 978-1408855942 Fantasy 90 min
 Newspaper: Die Harry Potter News, 2 €, 15 int stock, 20.10.2020 00:00:00

Amount: 10 Purchase Sell Delete

Show status and error messages here...

- **buttonReorderItemss** - ruft die entsprechende Methode des Store-Objekts auf und aktualisiert die aktuelle Anzeige in der ListBox.

Schritt 5: Zusatzaufgaben

Untersuche die Layout Container des Fensters

Wenn du das Programmfenster unserer Anwendung vergrößerst bzw. verkleinerst wird du feststellen, dass sich die Größe der UI Controls (Steuerelemente) automatisch mitverändert. Das kommt daher, dass das User Interface nicht mittels fixer Positionen und Größen, sondern mittels Layout Container gebaut ist.

Untersuche die verwendeten Layout Container `StackPanel`, `DockPanel` und `Grid` und verstehe ihre Funktionsweise.

Untersuche die Stilangaben für das Fenster

Statt für jedes Steuerelement seine Formatierungseigenschaften (Größe, Ränder, Schriftart, Hintergrund, etc. etc.) anzugeben, kann man in WPF Stilvorlagen definieren, die für alle Steuerelement eines Typs gelten. Schau dir dazu das `<Window.Resources>` Element in XAML an; du kannst diese Fähigkeit in Zukunft auch für deine eigenen Programme nutzen!

Happy Coding!