

DataGrid

[Документация DataGrid](#)

[Статья на Metanit по DataGrid](#)

DataGrid позволяет отображать и изменять данные из различных источников, например из базы данных SQL, запроса LINQ или любого другого источника данных, доступного для привязки.

В этом задании вам предстоит реализовать приложение, позволяющее контролировать книги в библиотеке. Библиотека имеет в своем распоряжении коллекцию книг, каждая книга входит в эту коллекцию в некотором количестве экземпляров. Библиотека выдает книги читателям, получает их обратно, а также периодически заказывает новые экземпляры. Классы, описывающие библиотеку, вам уже даны, вам необходимо лишь реализовать WPF приложение, которое будет, используя DataGrid, отображать данные о книгах в библиотеке.

Задание 1.

В первую очередь необходимо добавить DataGrid на форму. Дополните содержимое MainWindow.xaml:

Добавьте в `<DataGrid.Columns></DataGrid.Columns>` 8 колонок:

- 1) ISBN книги (`Book.ISBN`)
- 2) Название книги (`Book.Title`)
- 3) Автор книги (`Book.Author`)
- 4) Год издания (`Book.Year`)
- 5) Количество копий, которые есть в библиотеке (`Total`)
- 6) Количество копий, доступных для выдачи (`Available`)
- 7) Маркер того, планирует ли библиотека заказать экземпляры данной книги (`ToBeOrdered`)
- 8) Ссылка на книгу на сайте библиотеки (`Link`)

WPF предлагает разные типы колонок. Выберите наиболее подходящий тип для каждого значения.

В таблице мы будем отображать список `List<LibraryItem> books` объекта класса `Library`. Поэтому при создании колонки необходимо указать «путь» к ее значению. Для этого используйте параметр `Binding`. Для того, чтобы указать путь к «вложенным» свойствам, используйте «.» синтаксис.

В нашей системе есть значения, которые менять можно, и те, которые менять нельзя. Это необходимо также учесть при создании колонок.

Чтобы иметь возможность открывать ссылки (колонка 8), добавьте метод `OnHyperlinkClick()` как обработчик соответствующего события.

Задание 2.

Теперь, когда мы добавили на форму `DataGrid`, мы можем отобразить в таблице данные.

Дополните конструктор `MainWindow()`:

- 1) Используя предоставленную библиотеку, отобразите в таблице коллекцию книг библиотеки (`List<LibraryItem> books`). Для этого установите свойство `ItemsSource` объекта `DataGrid`.
- 2) Для удобства тестирования, используйте метод `Library.CreateRandomLibrary(int)`, чтобы добавить в библиотеку случайные книги.

Как уже было сказано, `DataGrid` позволяет изменять данные из привязанного источника. Можете поэкспериментировать и попробовать поменять данные в колонках, где это возможно. Когда вы будете менять значение в колонке, оно автоматически изменится и в источнике (в нашем случае, в списке книг библиотеки).

Кнопка «Refresh Table» меню переопределяет источник данных таблицы и обновляет ее содержимое. Можете проверить, что после обновления содержимого таблицы, ваши изменения сохраняются.

Задание 3.

Рассмотрим теперь, как мы можем программно менять содержимое таблицы. Реализуйте метод `newBook_Click()`, добавляющий новую книгу в библиотеку.

Метод почти полностью написан. Вам необходимо лишь достать из объекта `NewBookWindow` полученную книгу и добавить ее в таблицу.

Задание 4*.

Табличные данные удобно хранить в [CSV файлах](#). В этом задании вам необходимо реализовать сохранение данных из `DataGrid` в CSV файл. Для этого классе `Library.Utils` реализуйте методы `ReadLibraryFromCSV()` и `WriteLibraryToCSV()`. В классе `MainWindow` реализуйте методы `openCSV_Click()` и `saveCSV_Click()`, использующие `ReadLibraryFromCSV()` и `WriteLibraryToCSV()` соответственно.

Задание 5*.

Реализуйте метод `searchLine_TextChanged()`, который смотрит на вводимый пользователем запрос в поисковой строке и фильтрует таблицу, отображая только соответствующие запросу результаты. «Соответствие» книги запросу можете определять на свое усмотрение.