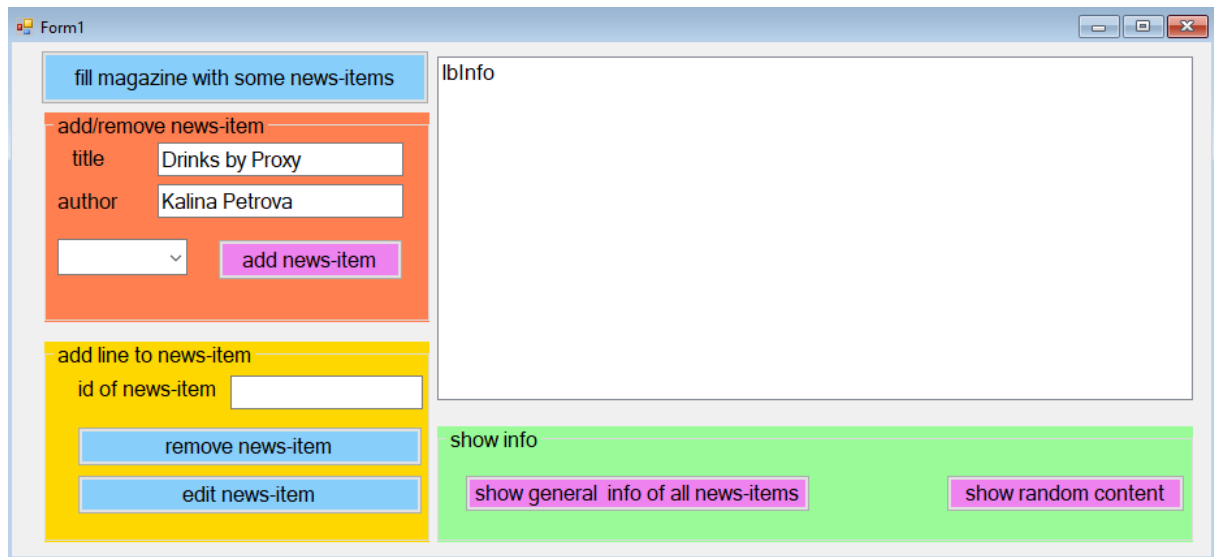


The Magazine app

Introduction: An app for keeping track of news-items of a magazine.

With this app it should be possible to keep track of news-items of a magazine. Authors can write news-items for a magazine. It should be possible to add news-items, to update news-items, to remove news-items and some more functionality.

For this app the user interface has already been made (see figure below).



Screenshot 1

The start-up project contains some code, but it does not compile yet! You have to implement some code to make it work. More specific instructions will be provided in the assignments.

The class diagram for the desired solution is:

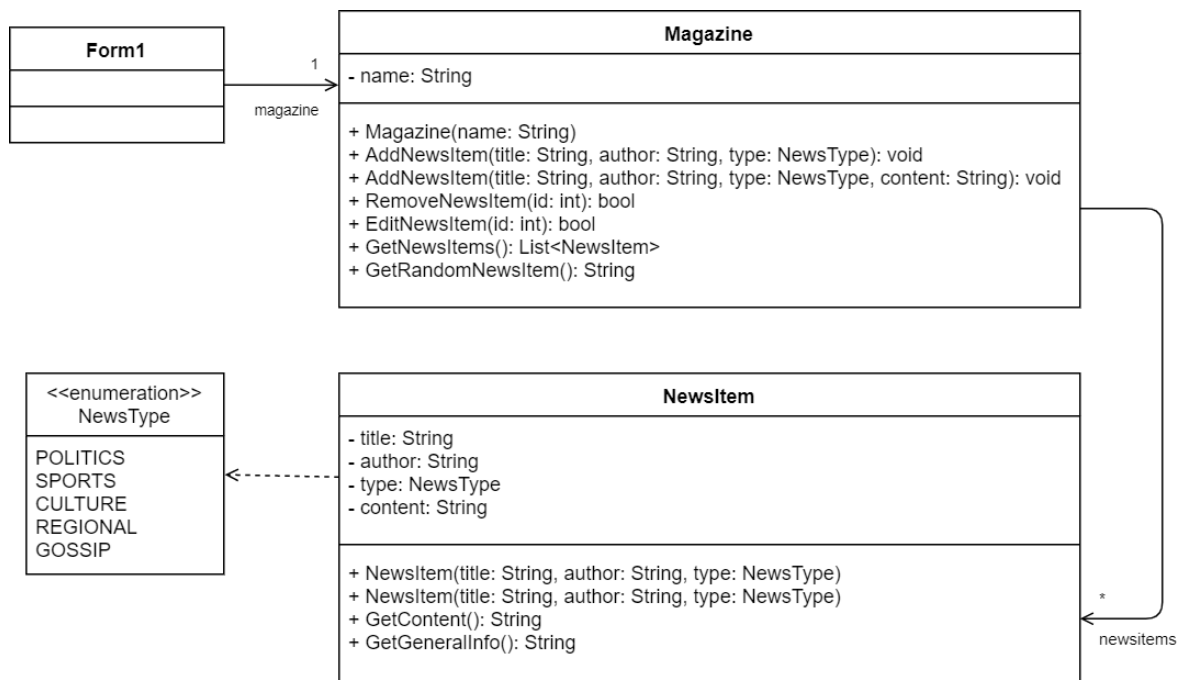


Diagram 1

The classes `Magazine` and `NewsItem` are empty.
The enum `Newstype` is missing.

1. About the enum `Newstype`.

Add the enum `Newstype` to the project according to the above class diagram.

2. About the class `NewsItem`.

Implement the class `NewsItem` according to the above class diagram. Some explanation and additional requirements:

- a) Every news-item should have a title (for instance "Elton John is back in town"), is written by an author (for instance "Georgiana Manolache"), has an unique identity-number, a string for content and a news type.
- b) The class `NewsItem` must have 2 constructors:
 - one constructor has parameters for the title, the author and the newstype. This constructor creates a `NewsItem`-object with title, author and newstype as in the parameters;
 - another contractor has parameters for the title, the author, the newstype and content. This constructor creates a `NewsItem`-object with title, author, newstype and content as in the parameters.

The news-item that will be created first should get 100 as identity-number; the next news-item that will be created should get 101 as identity-number; the next gets 102; the next gets 103, etc.
- c) There should be a method `GetContent` to get the news-item's content.
- d) There must be a method `GetGeneralInfo` that returns a string holding information about the id, title, author and the number of lines.

3. About the class `Magazine`.

Implement the class `Magazine` according to the above class diagram. Some explanation and additional requirements:

- a) A magazine has a name and a collection of news-items.
- b) The constructor should create a `Magazine`-object with name as in the parameter and an empty collection of news-items.
- c) The methods `AddNewsItem` should add a news-item to the magazine as in the parameters.
- d) The method `EditNewsItem` should, if possible, edit the news-item with that id from the magazine. In case it is successfully edited, this method should return true, otherwise it returns false.
- e) The method `RemoveNewsItem` should, if possible, remove the news-item with that id from the magazine. In case it is successfully removed, this method should return true, otherwise it returns false.
- f) The method `GetNewsItems` should return a collection of all news-items of the magazine.
- g) The method `GetRandomNewsItem` should return a random news-item's content of the magazine.

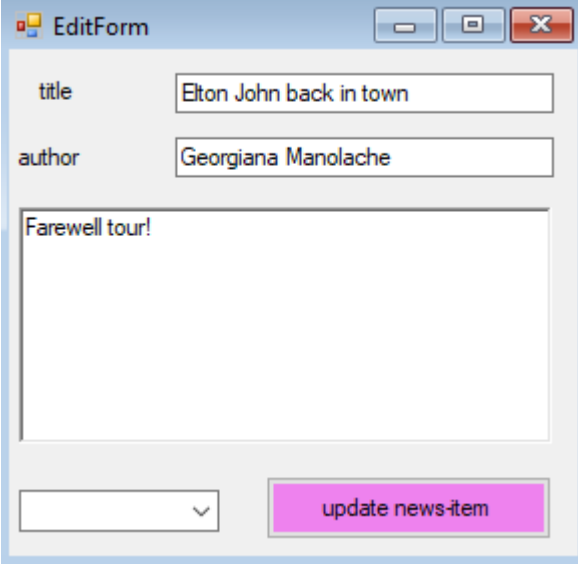
4. The `Form1`-class.

Button `btnFillMagazineWithSomeNewsitems` is already implemented, uncomment the lines of code and do not remove it. It will be your task to implement the rest of the buttons.

- a) Clicking the button with text "add news-item" should add a news-item as specified in the orange groupbox.
Note that the combobox is already populated with all the possible news-item types.

- b) Clicking the button with text "show general info of all news-items" should show general information of all news-items in the listBox on the right-top. Make it work.
- c) Clicking the button with text "show random news-items" should show in a MessageBox with the content a randomly selected news-item. Make it work.
- d) Clicking the button with text "remove news-item" should check if there is a news-item with identity-number as specified in the textbox tbIdNewsItem. If so, should remove it, else show an appropriate message.
- e) Clicking the button with text "edit news-item" should check if there is a news-item with identity-number as specified in the textbox tbIdNewsItem. If so, it should show another window (EditForm), else show an appropriate message.

The user interface for EditForm is already implemented for you.

A screenshot of a Windows-style window titled "EditForm". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there are three input fields: a "title" field containing "Elton John back in town", an "author" field containing "Georgiana Manolache", and a larger rich text box containing "Farewell tour!". Below these fields is a dropdown menu with a downward arrow. To the right of the dropdown is a pink button labeled "update news-item".

Screenshot 2

Make sure to update the controls (textboxes and combobox) with the values from the selected news-item to be edited. Apart from the title, author and type, there is a richTextBox control, where the content of the news-item can be updated. Pressing button "update news-item" should update the item accordingly and should close the window (EditForm).