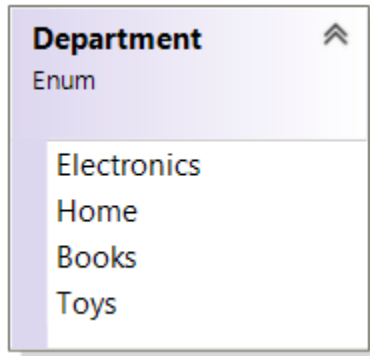


Assignment 4

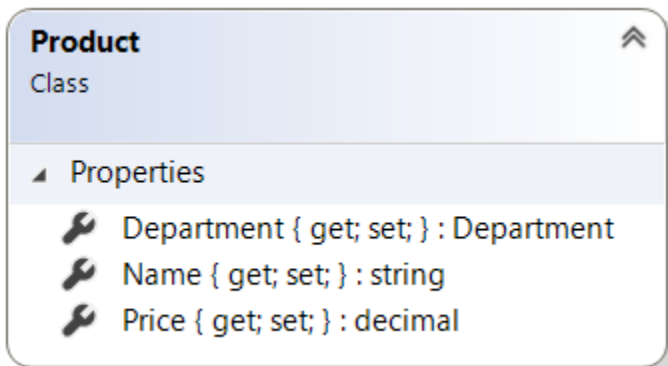
In this assignment you will be managing a catalog of products using a windows forms application. The application will be able to load and save the entire catalog from a text file (you may choose any format you like).

Make sure your program is created to be user friendly. This will require you to ensure your program does not throw any unhandled exceptions and also don't forget about tab order.

Here is a brief description of the required types:



- An enum containing the different departments that a product could belong to



- A class describing a products basic properties

Catalog

Class

Fields

`_list : List<Product>`

Methods

`Add(Product product) : void`

`GetAllProducts() : IEnumerable<Product>`

`GetProducts(string filter) : IEnumerable<Product>`

- A class containing a list of products
 - New products can be added
 - A full list of products can be retrieved
 - A filter list of products can be retrieved

DataStore

Static Class

Methods

`Load(string filePath) : Catalog`

`Save(Catalog catalog, string filePath) : void`

- A static class that handles persistent storage of a catalog
 - An entire catalog can be loaded from a file path
 - An entire catalog can be saved to a file path

ProductEventArgs

Class

→ EventArgs

Properties

`NewProduct { get; } : Product`

Methods

`ProductEventArgs(Product product)`

- A class to handle product based event arguments

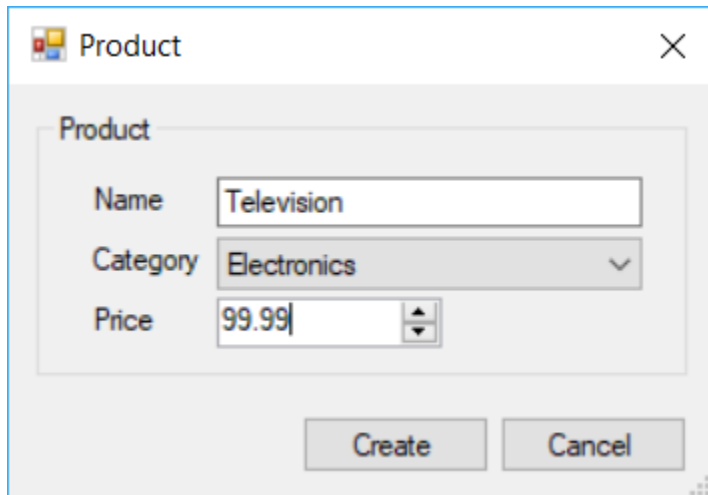
Here is an example of what the forms should look like:

The screenshot shows a Windows application window titled "Catalog". It has a standard Windows interface with minimize, maximize, and close buttons. The window is divided into several sections. At the top, there's a "Data Store" section containing a "File Path" text box with the value "Catalog.txt", a "Load" button, and a checkbox labeled "Auto Save Every" followed by a numeric control set to "500". Below this, there's a "Create Product" button and a "Filter" text box. The main area of the window is occupied by a datagridview. The datagridview has three columns: "Name", "Department", and "Price". It contains six rows of data: "Pillow" (Home, 30), "Bookcase" (Home, 74), "Table" (Home, 245), "Smart TV" (Electronics, 500), "Nintendo Switch" (Electronics, 399), and "Nintendo 3DS" (Electronics, 100). The first row, "Pillow", is highlighted in blue.

Name	Department	Price
Pillow	Home	30
Bookcase	Home	74
Table	Home	245
Smart TV	Electronics	500
Nintendo Switch	Electronics	399
Nintendo 3DS	Electronics	100

- Loading a Catalog
 - The load button should load a catalog of products from the file path specified in the textbox and display the results in the datagridview
- Saving a Catalog
 - If the auto save is checked, then saving the catalog to the file path will happen automatically
 - The frequency of the save can be configured using the numeric control
- Creating a New Product
 - When the create product button is clicked a new form will appear for creating a new product
- Displaying Products
 - When a catalog is loaded then it will be displayed in the datagridview
 - When a product is added to the catalog then it will be displayed in the datagridview

- If there is something typed in the filter textbox then only products the contain the value in their name will be shown



The image shows a 'Product' dialog box with a close button in the top right corner. The dialog contains a 'Product' section with three input fields: 'Name' (text box containing 'Television'), 'Category' (dropdown menu showing 'Electronics'), and 'Price' (numeric spinner set to '99.99'). At the bottom of the dialog are two buttons: 'Create' and 'Cancel'.

- Enter the product name in a textbox
- Select the category in a drop down
- Enter the price in a numeric control
- When the create button is clicked, the form should raise an event that other components can listen to and the close the form
- When the cancel button is clicked, the form should close

Unit Tests

- Write enough tests to fully cover the Catalog and Product classes