

Application Development with .NET (32998, 31927)

Lab-8 Questions

Please download the sample code from Canvas and follow the instructions

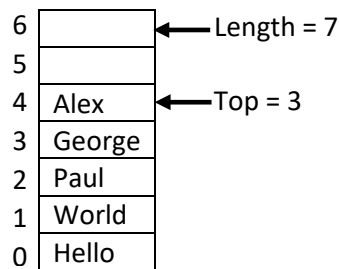
Program 1:

Write a program to create a Generic Stack data structure which can handle multiple data types. Use the following class diagram as a reference:

MyGenericStack<T>
- top : int - stackElements: T[]
+ MyGenericStack (int myGenericStackSize) + Push(T value): void + Pop(): T + Length (Property: returns stackElements length) + NoElements (Property: returns top)

Description:

1. **top**: Keeps track of the number of elements added to the Stack in LIFO (Last In First Out) order, as shown in the example below:



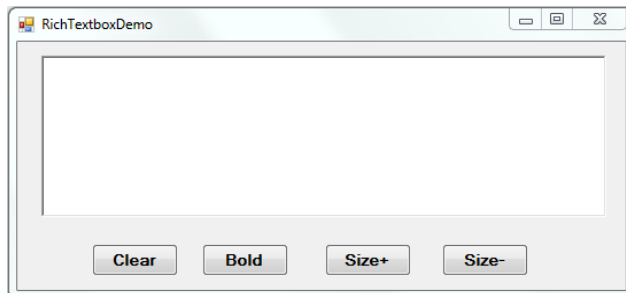
stackElements

2. **stackElements**: Is an array and the type is determined based on the type of 'T'
3. **MyGenericStack()**: Is the constructor which initializes the stack based on the value of myGenericStackSize, which should be greater than zero.
4. **Push()**: The method will add an element in the stack. First the value of **top** is incremented by 1, and checked if **top < Length**. If the value of **top** is less than Length, the element is added, else an exception, `StackOverflowException` is created with the message "Stack is full, cannot push the value" and is thrown. Syntax:
`throw new StackOverflowException("Stack is full, cannot push the value");`
5. **Pop()**: The method returns the current value at the location indicated by **top** and will decrement the value of **top** by 1. It should also check whether the value of **top < 0**, which indicates that the stack is empty.
6. **Length** and **NoElements**: Are accessors which return the size of the stack and the number of elements currently in the stack, respectively.

The class for testing the generic stack and expected output are provided in the sample code.

Program 2:

Create a WindowsForm application using RichTextBox control and Buttons. Your form should look similar to the sample form is given below as reference:



1. Add a RichTextBox control
2. Add 4 button and rename them as shown in the sample form above.
3. Add the following functionalities.

Functionalities required:

1. User should be able to add multiple lines of text in the RichTextBox, and should have both horizontal and vertical scrollbars.
2. **Clear** Button: Delete the contents of the RichTextBox
3. **Bold** Button: Makes the selected text to Bold style
4. **Size+** Button: Increases the font size of the selected text by 2.
5. **Size-** Button: Decreases the font size of the selected text by 2.

Hint:

Useful Properties of RichTextBox:

- **SelectionFont**: Gets or sets the font of the current text selection or insertion point. Useful for font related manipulations.
- **Font**: Gets or sets the font of the text displayed by the control.
- **Text**: Gets or sets the current text in the rich text box.

Useful methods of RichTextBox:

- **Clear()**: Clears all text from the text box control.
- **Copy()**: Copies the current selection in the text box to the **Clipboard**.
- **Cut()**: Moves the current selection in the text box to the **Clipboard**.
- **Paste()**: Replaces the current selection in the text box with the contents of the **Clipboard**.

Useful Reference:

1. <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.richtextbox?view=netframework-4.7.2>
2. https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.richtextbox.selectionfont?view=netframework-4.7.2#System_Windows_Forms_RichTextBox_SelectionFont