# 03. MailClient



*You will have to create a simple mail client application by implementing the classes needed and the programming logic in the methods.*

**Preparation**

Download the provided skeleton in Judge. **Do not** change the **StartUp** class or its **namespace**.

**Ensure that the project is named MailClient, and all classes, fields, and methods should be named exactly as presented in the document. Maintain the project structure as described below.**

**Problem Description**

Your task is to **create a mail repository** by implementing the following classes.

### Mail

You are given a class **Mail** with the following properties:

* **Sender – string**
* **Receiver - string**
* **Body - string**

The class **constructor** should receive **sender**, **receiver**, and **body**.

* Override the **ToString()** method in the following format:  
  **"From: {sender} / To: {receiver}**

**Message: {body}"**

### MailBox

**Next**, you have a **MailBox** class that contains **Inbox** and **Archive** (collections for storing mails). All entries inside the repository have the **same properties**. The **MailBox** class should have the following **properties**:

* **Capacity - int**
* **Inbox – List<Mail>**
* **Archive – List<Mail>**

The class **constructor** should receive **capacity** and initialize the **Inbox** and **Archive** with new instances of the collections.

Implement the following features:

* **Method IncomingMail(Mail mail)** – **adds** an **entry** to the **Inbox** **collection**, **if** the **Capacity** **allows it**.
* **Method DeleteMail(string sender)** – Finds and **removes** the first **mail from the Inbox by** a **given sender,** if such **exists**, and **returns boolean** (**true** if it is removed, otherwise – **false**)
* **Method ArchiveInboxMessages()** – The method moves all inbox mails to the **Archive**. **Returns** the number of **mails moved.**
* **Method GetLongestMessage()** – **returns** the **ToString()** method of the **Mail** with the longest **Body.**
* **Method InboxView()** – **returns** a **string** in the following **format**:
  + **"Inbox:  
    {Mail1}  
    {Mail2}  
    {…}**

**{Mailn}"**

### Constraints

* You will always have **mails added** **before** receiving commands **manipulating** the collections in the MailBox.
* The **Capacity** property is related to the **Inbox** only.

### Examples

This is an example of how the **MailBox** class is **intended to be used**.

|  |
| --- |
| **Sample code usage** |
| //Initialize new repository (MailBox)  MailBox mailbox = new(5);  //Initialize entities (Mail)  Mail mail1 = new("John", "Alice", "Hello Alice, How are you?");  Mail mail2 = new("Alice", "Bob", "Hi Bob, Here's the document you requested.");  Mail mail3 = new("Bob", "Charlie", "Hey Charlie, Let's meet for lunch.");  Mail mail4 = new("Charlie", "David", "Hi David, Can you help me with the project?");  Mail mail5 = new("David", "Eva", "Hello Eva, Don't forget our meeting tomorrow.");  Mail mail6 = new("Eva", "Frank", "Hi Frank, I found an interesting article for you.");  Mail mail7 = new("Frank", "Grace", "Hey Grace, How's your day going?");  Mail mail8 = new("Grace", "Henry", "Hi Henry, Please review the proposal.");  Mail mail9 = new("Henry", "Isabella", "Hello Isabella, Let's schedule a call.");  Mail mail10 = new("Isabella", "John", "Hi John, I received your message. Thanks!");  //Send mails to Inbox  mailbox.IncomingMail(mail1);  mailbox.IncomingMail(mail2);  mailbox.IncomingMail(mail3);  mailbox.IncomingMail(mail4);  mailbox.IncomingMail(mail5);  //Send when Capacity is full  mailbox.IncomingMail(mail6);  //Delete existing Mail  Console.WriteLine(mailbox.DeleteMail("David")); //True  //Try to delete not existing Mail  Console.WriteLine(mailbox.DeleteMail("Eva")); //False  //Try to send once again, if there is enough Capacity  mailbox.IncomingMail(mail6);  //mail6 should exist and be available to delete  Console.WriteLine(mailbox.DeleteMail("Eva")); //True  //Archive mails and print count  Console.WriteLine(mailbox.ArchiveInboxMessages()); //4  //Send the rest of the mails  mailbox.IncomingMail(mail7);  mailbox.IncomingMail(mail8);  mailbox.IncomingMail(mail9);  mailbox.IncomingMail(mail10);  //Print the mail with the longest message  Console.WriteLine(mailbox.GetLongestMessage());  //From: Isabella / To: John  //Message: Hi John, I received your message.Thanks!  //Display all mails currently in Inbox  Console.WriteLine(mailbox.InboxView());  //Inbox:  //From: Frank / To: Grace  //Message: Hey Grace, How's your day going?  //From: Grace / To: Henry  //Message: Hi Henry, Please review the proposal.  //From: Henry / To: Isabella  //Message: Hello Isabella, Let's schedule a call.  //From: Isabella / To: John  //Message: Hi John, I received your message.Thanks! |

**Submission**

Zip all the files in the project folder except **bin** and **obj** folders.