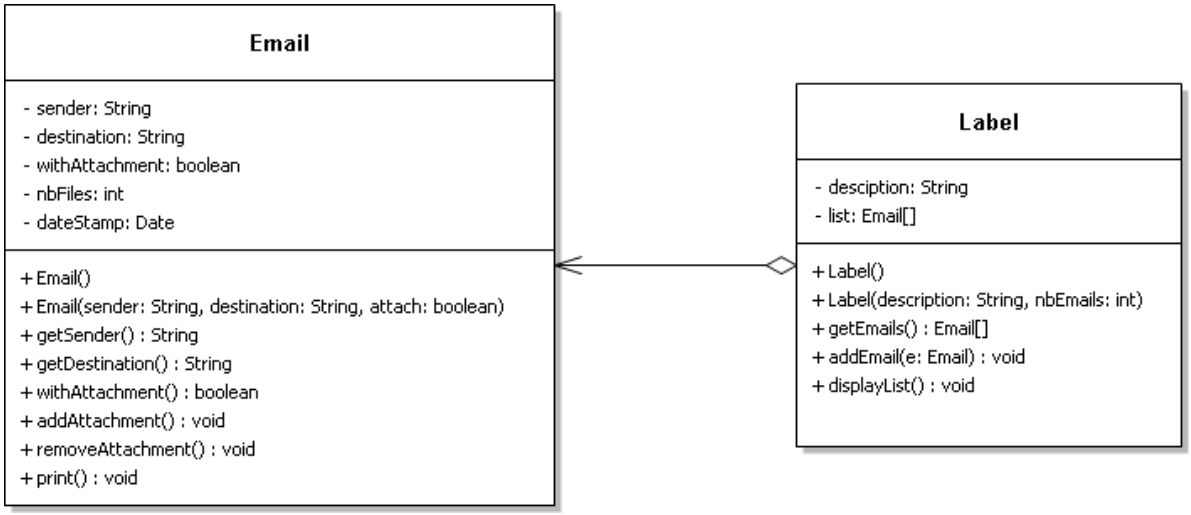
Implement the two classes defined as shown in the below UML diagram:



A. The Email class defines an email sent from a **sender** to a **destination** represented as string data members in the class. An email might be sent with **attachment** (value will be true). Only when this property is set to **true**, it would be possible to add an attachment, therefore use the **addAttachment** method which would increment the **nbFiles** attached by one. A maximum of 5 attachments is allowed. The **removeAttachment** method decrements the nbFiles in case the email will be sent with an attachment. The **dateStamp** has the current date of creation as value.
    1. Implement the Email class.
    2. Define the constructors that initialize data fields.
    3. Getter methods should also be implemented.
    4. Write the body of the addAttachment and removeAttachment methods based on the above description
    5. The print method is used to display the email information. In case of attachment allowed, indicate the number of files.

B. The Label class represents a folder to group emails received or sent by topic.
    - So the **description** data field is used as the label title.
    - Each label might contain a fixed number of emails defined by the user when creating a new label. So this will be the **size** of the **list** array.
    - The no-arg constructor creates an array of size 10 by default.
    - The **getEmails** method returns the list of emails added to the label.
    - The **addEmail** method adds the email sent as parameter to the list, whereas the **removeEmail** removes it from array. Two emails are considered identical if they have the same sender and destination.
    - The displayList method prints all the emails information added to the label.
    - **Implement Label class as described above.**

C. Write a main class in which the user is asked to do the following:

- Enter a label description and the number of emails it can contain.
- Fill the list created with email objects. The user is asked to enter all the email information: sender, destination, withAttachments and whether or not to add attachments.
- You will implement a method called **countNbEmailWithAttachments** returns the number of emails entered in the list that are defined with attachments.
- Display all the emails information available in the list.