Commencé le	jeudi 5 octobre 2017, 11:22
État	Terminé
Terminé le	jeudi 5 octobre 2017, 12:10
Temps mis	47 min 30 s
En retard	47 min 29 s
Points	2,90/3,00
Note	**19,33** sur 20,00 (**97**%)

Description

You're going to develop a small system for sending and reading mail. There is a mail server; clients can send mail to the server for other clients, and can read mail sent to them from the server. The system has two classes which have already been developed by your company:

- `MailItem` represents an item of mail to be exchanged between clients and server;
- `MailServer` is the server that handles mail items between clients.

Download these from here (chapter03_mailserver.jar) into your project.

There is a third class which is missing because your company didn't have anyone with the skills to write it. And that's why you've been hired, to develop the `MailClient` class. See the next question for details.

The general rules for an exercise apply: Do the questions below in order. You should finish each question correctly before moving on to the next question. Your answers are evaluated as you go along. You may enter successive code answers to each question until your answer passes all tests. But...you will lose points depending on the number of solutions submitted. One failing answer loses zero points (you're welcome!). Each succeeding failing answer loses 10% of the question's mark. Within each question, some tests may be hidden; they will only be shown when all previous tests have passed.

And don't forget to add yourself as an author of the code - you own it now.

Question 1
Correct
Note de 0,90 sur 1,00

A skeleton structure for the `MailClient` class is given below; you have to implement the methods:

```
package mailsystem;

/**
 * A class to model a simple email client. The client is run by a particular
 * user, and sends and retrieves mail via a particular server.
 *
 * @author David J. Barnes and Michael Kolling
 */
class MailClient {
    // The server used for sending and receiving.

    // The user running this client.

    /**
     * Create a mail client run by user and attached to the given server.
     */
    MailClient(MailServer server, String user)

    /**
     * Return the next mail item (if any) for this user.
     */
    private MailItem getNextMailItem()

    /**
     * Print the next mail item (if any) for this user to the text terminal.
     */
    void printNextMailItem()

    /**
     * Send the given message to the given recipient via the attached mail
     * server.
     *
     * @param to
     *           The intended recipient.
     * @param message
     *           The text of the message to be sent.
     * @return true if the item was sent; false otherwise.
     */
    boolean sendMailItem(String to, String message)
}
```

Once you have finished developing `MailClient`, copy all the code for this class only, paste it into the Answer box and Check the answer.

Don't change any of the other classes (`MailServer` and `MailItem`); you don't have to submit either of these classes.

Note: You may find that the compiler complains that one of the declared method access-levels above is too strict. Change it (not more than necessary) for the code to compile.

For example:

Test	Result

`String JOHN = "John";`
`MailServer server= new MailServer();`
`MailClient client = new MailClient(server, JOHN);`
`System.out.println(client == null);`
	`false`

Réponse:

```
        }

        /**
         * Send the given message to the given recipient via the attached mail
         * server.
         *
         * @param to
```

```
    *           The intended recipient.
    * @param message
    *           The text of the message to be sent.
    * @return true if the item was sent; false otherwise.
    */
   boolean sendMailItem(String to, String message){
       MailItem mail = new MailItem(nom,to,message);
       if(server.post(mail))return true;
           else return false;
   }
}
```

Vérifier

	Test	Expected	Got

✓	`String JOHN = "John";`
	`MailServer server= new MailServer();`
	`MailClient client = new MailClient(server, JOHN);`
	`System.out.println(client == null);`
		`false`	`fal`

✓	`String JOHN = "John";`
	`String PAUL = "Paul";`
	`MailServer server= new MailServer();`
	`MailClient client = new MailClient(server, JOHN);`
	`System.out.println(new MailClient(server, PAUL) != null);`
		`true`	`tru`

✓	`String JOHN = "John";`
	`String PAUL = "Paul";`
	`String MSG = "Yo! Ni hao! Tervist!";`
	`MailServer server= new MailServer();`
	`MailClient client = new MailClient(server, JOHN);`
	`System.out.println(client.sendMailItem(PAUL, MSG));`
		`true`	`tru`

✓	`String JOHN = "John";`
	`String PAUL = "Paul";`
	`String MSG = "Yo! Ni hao! Tervist!";`
	`MailServer server= new MailServer();`
	`MailClient from = new MailClient(server, JOHN);`
	`MailClient to = new MailClient(server, PAUL);`
	`from.sendMailItem(PAUL, MSG);`
	`MailItem item = to.getNextMailItem();`
	`System.out.println(item != null);`
	`System.out.println(MSG.equals(item.getMessage()));`
	`System.out.println(JOHN.equals(item.getFrom()));`
	`System.out.println(PAUL.equals(item.getTo()));`
		`true`
		`true`
		`true`
		`true`
			`tru`
			`tru`
			`tru`
			`tru`

✓	`String JOHN = "John";`
	`String PAUL = "Paul";`
	`String MSG = "Yo! Ni hao! Tervist!";`
	`MailServer server= new MailServer();`
	`MailClient from = new MailClient(server, JOHN);`
	`MailClient to = new MailClient(server, PAUL);`
	`from.sendMailItem(PAUL, MSG);`
	`from.sendMailItem(PAUL, MSG);`
	`to.printNextMailItem();`
	`to.printNextMailItem();`
	`to.printNextMailItem();`
		`From: John`
		`To: Paul`
		`Message: Yo! Ni hao! Tervist!`
		`From: John`
		`To: Paul`
		`Message: Yo! Ni hao! Tervist!`
		`No new mail.`
			`Fro`
			`To:`
			`Mes`
			`Fro`
			`To:`
			`Mes`
			`No`

Passed all tests! ✓

Correct

Points pour cet envoi : 1,00/1,00. En tenant compte des tentatives précédentes, cela donne **0,90/1,00**.

Question 2

Correct

Note de 1,00 sur 1,00

Submit your `MailClient` again. Just checking that you own it.

Réponse:

```
package mailsystem;

/**
 * A class to model a simple email client. The client is run by a particular
 * user, and sends and retrieves mail via a particular server.
 *
 * @author David J. Barnes and Michael Kolling
 * @author Florian Salord
 */
class MailClient {
    // The server used for sending and receiving.
    private MailServer server;
    // The user running this client.
    private final String nom;

    /**
     * Create a mail client run by user and attached to the given server.
     */
```

Vérifier

	Test	Expected	Got	
✓	""	Additional author added	Additional author added	✓

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 3

Correct

Note de 1,00 sur 1,00

Add a subject line for an email to mail items in the mailsystem package. Make sure printing messages also prints the subject line, eg,

```
From: Wilma
To: Fred
Subject: supper
Message: Did you remember to pick up the mammoth for supper? Better make it a large one if you c
an - we're having the whole tribe over!
```

You'll probably have to modify the mail client accordingly.

Submit *both* classes, `MailItem` and `MailClient` into the Answer box and Check your code.

For example:

Test

```
String FRED = "Fred";
String WILMA = "Wilma";
String MSG = "Did you remember to pick up the mammoth for supper? Better make it a large one
MailServer server = new MailServer();
MailClient from = new MailClient(server, "Wilma");
MailClient to = new MailClient(server, "Fred");
from.sendMailItem(FRED, "supper", MSG);
System.out.println("Fred, you have " + server.howManyMailItems(FRED) + " mails...");
to.printNextMailItem();
to.printNextMailItem();
```

Réponse:

```
package mailsystem;

/**
```

```
 * A class to model a simple mail item. The item has sender
 * and recipient addresses and a message string.
 *
 * @author David J. Barnes and Michael Kolling
 * @author Florian Salord
 * @version 2016.02.29
 */
class MailItem {
    // The sender of the item.
    private final String from;
    // The intended recipient.
    private final String to;
    // The text of the message.
    private final String message;

```

Vérifier

	Test

✓	`String FRED = "Fred";`
	`String WILMA = "Wilma";`
	`String MSG = "Did you remember to pick up the mammoth for supper? Better make it a large one`
	`MailServer server = new MailServer();`
	`MailClient from = new MailClient(server, "Wilma");`
	`MailClient to = new MailClient(server, "Fred");`
	`from.sendMailItem(FRED, "supper", MSG);`
	`System.out.println("Fred, you have " + server.howManyMailItems(FRED) + " mails...");`
	`to.printNextMailItem();`
	`to.printNextMailItem();`

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.