FPT SOFTWARE

FRESHER ACADEMY

BASIC JAVA

Ngattt (FHO.FA)

04/20/2022

[Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.]

Contents

Objective 2
Business needs 2
Working requirements 2
Technologies 2
Project Descriptions 2

FPT Software	**Assignment topic** : Java Basic Lab **Assignment duration** : 30 minutes	**FRESHER ACADEMY**

Objective

- Fresher can apply knowledge about Abstract class, Subclasses to code a simple application.
- Create an Email class with instance variables for sender, recipient, subject, and content.
    - Implement appropriate constructors, getters, and setters for the Email class.
- Create an EmailManagementSystem class that allows users to send emails and display the inbox.
    - Implement a method to display the number of unread emails in the inbox.
- Fresher can create a class, and write some methods; create some instances of class and test all the methods of the class.
- Fresher can using ArrayList: Click here to read about ArrayList

Business needs

- Allow users to send emails to different recipients.
- Display the list of all received emails in the inbox with their details (sender, subject, and content).
- Mark emails as read when they are opened.
- Calculate and display the number of unread emails in the inbox.

Working requirements

- Working environment: Eclipse/IntelliJ.

Technologies

- The product implements Java program language with: Abstract class, Sub-classes

Project Descriptions

1. **Step 1: Create a new Java project** (skip this step if you already have **YourFullName_JavaSE** project)
    - Open your preferred Java IDE (such as Eclipse, IntelliJ, or NetBeans).
    - Create a new Java project by selecting File > New > Java Project.
    - Name the project **YourFullName_JavaSE**, **for example** "**NguyenVanA_JavaSE**" and click Finish.
2. **Step 2: Create a package** (skip this step if you already have **lab4** package)
    - In the project explorer, right-click on the src folder and select New > Package.

- Name the package "**lab4**" and click Finish.

3. **Step 3: Create a Email class**
   - In the Package Explorer panel on the left side of the screen, right-click on the '**lab4**' package to create the class.
   - Select "New" from the context menu, then "Class" from the submenu.
   - In the "New Java Class" dialog box, enter "**Email**" for the class in the "Name" field.
   - Click "Finish" to create the class.
4. **Step 4: Create a EmailManagementSystem class**
   - In the Package Explorer panel on the left side of the screen, right-click on the '**lab4**' package to create the class.
   - Select "New" from the context menu, then "Class" from the submenu.
   - In the "New Java Class" dialog box, enter "**EmailManagementSystem**" for the class in the "Name" field.
   - Click "Finish" to create the class.
5. **Step 5: Create a StudentManagementSystemTest class**
   - In the Package Explorer panel on the left side of the screen, right-click on the '**lab4**' package to create the class.
   - Select "New" from the context menu, then "Class" from the submenu.
   - In the "New Java Class" dialog box, enter "**StudentManagementSystemTest**" for the class in the "Name" field.
   - Click "Finish" to create the class.
6. **Step 6**: Coding for **Email class:**
   - The Email class will have instance variables for sender, recipient, subject, and content.
   - Implement constructors, getters, and setters for the Email class to set and retrieve these attributes.
   - Implement **toString()** method to show information of Email
7. **Step 7**: Coding for **EmailManagementSystem class:**
   - **Step 7.1:** Allow users to send emails:
     - Provide the necessary details for the new email (sender, recipient, subject, and content).
     - The new email object will be created and added to the inbox.
   - **Step 7.2:** Display the list of all received emails in the inbox:
     - Iterate through the inbox and print the details of each email (sender, subject, and content).
   - **Step 7.3:** Mark emails as read when they are opened:

- The email with the matching email ID will be marked as read in the inbox.

- **Step 7.4:** Calculate and display the number of unread emails in the inbox:
  - Iterate through the inbox and count the number of emails that are marked as unread.

8. **Step 8**: Coding for **EmailManagementSystemTest** class:
   - In the main() method, test all the methods of **Email** and **EmailManagementSystem** classes.

The End!