**package** email\_Client;

**public** **abstract** **class** Recipient {

**private** String type;

**private** String name;

**private** String email;

**private** **static** **int** objectCount;

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** **static** **int** getObjectCount() {

**return** objectCount;

}

**public** **static** **void** setObjectCount(**int** objectCount) {

Recipient.objectCount = objectCount;

}

**public** Recipient(String name, String email) {

**this**.name = name;

**this**.email = email;

objectCount++;

}

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

**public** **interface** BirthdayGreeting {

**public** String getBirthdayWish();

**public** String getBirthday();

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

**public** **class** OfficeRecipient **extends** Recipient {

**private** String designation;

**public** String getDesgnation() {

**return** designation;

}

**public** **void** setDesgnation(String desgnation) {

**this**.designation = desgnation;

}

**public** OfficeRecipient(String name, String email,String designation) {

**super**(name, email);

**this**.designation = designation;

}

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

**public** **class** Office\_friend **extends** OfficeRecipient **implements** BirthdayGreeting{

**private** String birthday;

**public** Office\_friend(String name, String email, String designation,String birthday) {

**super**(name, email, designation);

**this**.birthday = birthday;

}

@Override

**public** String getBirthday() {

**return** birthday;

}

@Override

**public** String getBirthdayWish() {

**return** "Wish you a Happy Birthday. " + getName();

}

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

**public** **class** PersonalRecipient **extends** Recipient **implements** BirthdayGreeting{

**private** String nickname;

**private** String birthday;

**public** String getNickname() {

**return** nickname;

}

**public** **void** setNickname(String nickname) {

**this**.nickname = nickname;

}

@Override

**public** String getBirthday() {

**return** birthday;

}

**public** **void** setBirthday(String birthday) {

**this**.birthday = birthday;

}

**public** PersonalRecipient(String name, String nickname,String email,String birthday){

**super**(name, email);

**this**.nickname = nickname;

**this**.birthday = birthday;

}

@Override

**public** String getBirthdayWish() {

**return** "Wish you a Happy Birthday and hugs and love on your birthday. " + getName();

}

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

**import** javax.mail.\*;

**import** javax.mail.internet.InternetAddress;

**import** javax.mail.internet.MimeMessage;

**import** java.util.Properties;

**public** **class** SendMail {

**public** **static** **void** sendMail(String email, String subject, String content) {

**final** String username = "assignmentoop530@gmail.com";

**final** String password = "jejwfqlkooiudjdj";

Properties prop = **new** Properties();

prop.put("mail.smtp.host", "smtp.gmail.com");

prop.put("mail.smtp.port", "587");

prop.put("mail.smtp.auth", "true");

prop.put("mail.smtp.starttls.enable", "true"); //TLS

Session session = Session.getInstance(prop,

**new** javax.mail.Authenticator() {

**protected** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(username, password);

}

});

**try** {

Message message = **new** MimeMessage(session);

message.setFrom(**new** InternetAddress("assignmentoop530@gmail.com"));

message.setRecipients(

Message.RecipientType.TO,

InternetAddress.parse(email)

);

message.setSubject(subject);

message.setText("Hello,"

+ "\n\n" + content);

Transport.send(message);

//System.out.println("Email sent by sendMail");

} **catch** (MessagingException e) {

e.printStackTrace();

}

}

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**public** **class** UpdateSentMailList {

**public** **static** **void** updateSentMail(String subject, String emailAddress, String name,String date, String filePath) {

**try** {

//Create an object of PrintWriter

PrintWriter print\_writer = **new** PrintWriter(**new** FileWriter(filePath, **true**));

//Write details to text file

print\_writer.write("\n" + subject + "," + emailAddress + "," + name + "," + date);

//Close PrintWriter Object

print\_writer.close();

}

// Catch if exceptions occurs

**catch** (IOException e) {

// Print the exception

// using getMessage() method

System.out.print(e.getMessage());

}

}

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**import** java.util.Date;

**public** **class** RequireEmailClient {

**private** ArrayList<Recipient> clientList = **new** ArrayList<>();

**private** ArrayList<BirthdayGreeting> BirthdayGreetingList = **new** ArrayList<>();

**private** String currentDate;

//create recipient objects

**private** Recipient createRecipientObjects (String data) {

String[] clientList = data.split(":",2);

String[] dataList = clientList[1].split(",");

Recipient recipientObj;

**switch**(clientList[0]) {

**case** "Official":

recipientObj = **new** OfficeRecipient(dataList[0],dataList[1],dataList[2]);

//System.out.println("Object Created");

**return** recipientObj;

**case** "Office\_friend":

recipientObj = **new** Office\_friend(dataList[0],dataList[1],dataList[2],dataList[3]);

BirthdayGreetingList.add((BirthdayGreeting)recipientObj);

//System.out.println("Object Created");

**return** recipientObj;

**case** "Personal":

recipientObj = **new** PersonalRecipient(dataList[0],dataList[1],dataList[2],dataList[3]);

BirthdayGreetingList.add((BirthdayGreeting)recipientObj);

//System.out.println("Object Created");

**return** recipientObj;

**default**:

System.out.println("Invalid entry found");

**return** **null**;

}

}

**public** **void** readClientFile(String file\_name) {

**try** {

File clientFile = **new** File(file\_name);

FileReader fileReader = **new** FileReader(clientFile);

BufferedReader bufferedReader = **new** BufferedReader(fileReader);

String data;

**while** ((data = bufferedReader.readLine()) != **null**) {

**if** (!data.equals("")) {

//System.out.println(data);

clientList.add(createRecipientObjects(data));

}

}

bufferedReader.close();

} **catch** (IOException e) {

}

}

//method to send birthday greeting on the correct day for recipient who has birthday on current day

**private** **void** sendGreetingEmail() {

**for**(BirthdayGreeting birthdayObj: BirthdayGreetingList ) {

SimpleDateFormat formatDate = **new** SimpleDateFormat("yyyy/MM/dd");

currentDate = formatDate.format(**new** Date());

//System.out.println(curentDate.substring(5));

**if** (birthdayObj.getBirthday().substring(5).equals(currentDate.substring(5))) {

Recipient recip = (Recipient)birthdayObj;

SendMail.sendMail(recip.getEmail(), ("Happy Birthday"), birthdayObj.getBirthdayWish());

UpdateSentMailList.updateSentMail("Happy Birthday", recip.getEmail(), recip.getName(), currentDate, "D:\\UoM\\CSE\\Semester02\\CS1040 Program Construction\\Email\_Client\_Assignment\\sentEmailsList.txt");

}

}

System.out.println("Birthday wishes are sent who has birthday Today!\n");

}

**public** **void** startEmailClient() {

System.out.println("Email Client is Running..!!\n"

+ "Please wait...");

**try** {

sendGreetingEmail();

}**catch**(NullPointerException e) {

}

}

//print who has birthday on entered day

**public** **void** findWhoHaveBday(String Enter\_day) {

**boolean** haveBirthday = **false**;

**for**(BirthdayGreeting birthdayObj: BirthdayGreetingList ) {

**if** (birthdayObj.getBirthday().substring(5).equals(Enter\_day.substring(5))) {

Recipient recipient1 = (Recipient)birthdayObj;

System.out.println(recipient1.getName());

haveBirthday = **true**;

}

}

**if** (!haveBirthday) {

System.out.println("No birthdays on "+ Enter\_day + "\n");

}

}

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**public** **class** UpdateClientList {

**public** **void** updateClientList(String new\_recipient\_details, String filePath) {

//System.out.println(new\_recipient\_details);

//System.out.println(filePath);

**try** {

//Create an object of PrintWriter

PrintWriter print\_writer = **new** PrintWriter(**new** FileWriter(filePath, **true**));

//Write details to text file

print\_writer.write("\n" + new\_recipient\_details);

//Close PrintWriter Object

print\_writer.close();

}

// Catch if exceptions occurs

**catch** (IOException e) {

// Print the exception

// using getMessage() method

System.out.print(e.getMessage());

}

}

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** java.io.ObjectInputStream;

**import** java.io.ObjectOutputStream;

**import** java.util.ArrayList;

**public** **class** FindSentMail {

**private** **static** ArrayList<String> sentMailClientList = **new** ArrayList<>();

//Finding is there any mails have sent on entered day by reading sentMail file

**public** **static** **void** findSendMail(String date, String file\_name) {

**try** {

File clientFile = **new** File(file\_name);

FileReader fileReader = **new** FileReader(clientFile);

BufferedReader bufferedReader = **new** BufferedReader(fileReader);

String data;

**boolean** foundSentMail = **false**;

**while** ((data = bufferedReader.readLine()) != **null**) {

**if** (!data.equals("")) {

//System.out.println(data);

sentMailClientList.add(data);

String[] clientList = data.split(",",0);

**if** ((clientList[clientList.length-1]).equals(date)) {

foundSentMail = **true**;

System.out.println(clientList[0]+ "," + clientList[1]+ "," + clientList[2]);

}

}

}

**if** (!foundSentMail) {

System.out.println("No Emails Found on given date");

}

bufferedReader.close();

} **catch** (IOException e) {

}

}

//serialization sentMailClientList

**public** **void** saveSentMailListToHardDisk() {

**try** {

FileOutputStream fileOutputStream = **new** FileOutputStream("sentEmailListFile.ser");

ObjectOutputStream objectOutputStream = **new** ObjectOutputStream(fileOutputStream);

objectOutputStream.writeObject(sentMailClientList);

objectOutputStream.close();

fileOutputStream.close();

}**catch** (IOException e){

e.printStackTrace();

}

}

// deserialization sentMailClientList

@SuppressWarnings({ "unchecked", "unused" })

**private** **void** retrieveSentEmailList() {

**try** {

FileInputStream fileInputStream = **new** FileInputStream("sentEmailListFile.ser");

ObjectInputStream objectInputStream = **new** ObjectInputStream(fileInputStream);

sentMailClientList = (ArrayList<String>) objectInputStream.readObject();

objectInputStream.close();

}**catch**(IOException | ClassNotFoundException e) {

e.printStackTrace();

}

}

}

///////////////////////////////////////////////////////////////

**package** email\_Client;

//index number - 200530D

//import libraries

**import** java.util.Scanner;

**import** java.text.SimpleDateFormat;

**import** java.util.Arrays;

**import** java.util.Date;

**import** java.util.List;

**public** **class** Email\_Client {

**public** **static** **void** main(String[] args) {

RequireEmailClient requireEmailClientObj = **new** RequireEmailClient();

requireEmailClientObj.readClientFile("clientList.txt");

// call method to send birthday greeting on the correct day for recipient who has birthday on current day

//requireEmailClientObj.startEmailClient();

String sentMailPath = "D:\\UoM\\CSE\\Semester02\\CS1040 Program Construction\\Email\_Client\_Assignment\\sentEmailsList.txt";

String clientPath= "D:\\UoM\\CSE\\Semester02\\CS1040 Program Construction\\Email\_Client\_Assignment\\clientList.txt";

**try** (Scanner scanner = **new** Scanner(System.in)) {

System.out.println("Enter option type: \n"

+ "1 - Adding a new recipient\n"

+ "2 - Sending an email\n"

+ "3 - Printing out all the recipients who have birthdays\n"

+ "4 - Printing out details of all the emails sent\n"

+ "5 - Printing out the number of recipient objects in the application");

**int** option = scanner.nextInt();

**switch**(option){

**case** 1:

// input format - Official: nimal,nimal@gmail.com,ceo

// Use a single input to get all the details of a recipient

// code to add a new recipient

// store details in clientList.txt file

// Hint: use methods for reading and writing files

System.out.println("Input Format :\n"

+ "Official:<name>,<email>,<designation>\n"

+ "Office\_friend:<name>,<email>,<designation>,<birthday>\n"

+ "Personal:<name>,<nickname>,<email>,<birthday>\n"

+ "Example inputs: Official: nimal,nimal@gmail.com,ceo\n"

+ " Office\_friend: kamal,kamal@gmail.com,clerk,2000/12/12\n"

+ " Personal: sunil,<nick-name>,sunil@gmail.com,2000/10/10\n\n"

+ "Enter details of the new recipient ....: ");

UpdateClientList new\_recipient = **new** UpdateClientList();

Scanner input1 = **new** Scanner(System.in);

String new\_recipient\_details = input1.nextLine();

new\_recipient.updateClientList(new\_recipient\_details, clientPath );

**break**;

**case** 2:

// input format - email, subject, content

// code to send an email

System.out.println("Input Format : "

+ "<email>,<subject>,<content>\n"

+ "Enter Email details....:\n");

Scanner input2 = **new** Scanner(System.in);

String email\_details = input2.nextLine();

// split input and store in list

List<String> emailDataList = Arrays.asList(email\_details.split(","));

System.out.println("Email is sending...Please Wait...");

//call sendMail method in SendMail class

SendMail.sendMail(emailDataList.get(0),emailDataList.get(1),emailDataList.get(2));

System.out.println("Email sent\n");

//save sent mail details to text file

SimpleDateFormat formatDate = **new** SimpleDateFormat("yyyy/MM/dd");

String curentDate = formatDate.format(**new** Date());

UpdateSentMailList.updateSentMail(emailDataList.get(1), emailDataList.get(0), "No name entered", curentDate, sentMailPath);

**break**;

**case** 3:

// input format - yyyy/MM/dd (ex: 2018/09/17)

// code to print recipients who have birthdays on the given date

System.out.println("Input Format : "

+ "yyyy/MM/dd (ex: 2018/09/17)\n"

+ "Enter Date....:\n");

Scanner input3 = **new** Scanner(System.in);

String input\_date = input3.nextLine();

**while**(!input\_date.matches("\\d\\d\\d\\d\\/\\d\\d\\/\\d\\d")) {

System.out.println("Wrong Date Format..!!! Enter Again:");

input\_date = input3.nextLine();

}

requireEmailClientObj.findWhoHaveBday(input\_date);

**break**;

**case** 4:

// input format - yyyy/MM/dd (ex: 2018/09/17)

// code to print the details of all the emails sent on the input date

System.out.println("Input Format : "

+ "yyyy/MM/dd (ex: 2018/09/17)\n"

+ "Enter Date....:\n");

Scanner input4 = **new** Scanner(System.in);

String date\_email = input4.nextLine();

**while**(!date\_email.matches("\\d\\d\\d\\d\\/\\d\\d\\/\\d\\d")) {

System.out.println("Wrong Date Format..!!! Enter Again:");

date\_email = input4.nextLine();

}

FindSentMail.findSendMail(date\_email, sentMailPath);

**break**;

**case** 5:

// code to print the number of recipient objects in the application

System.out.println("Number of recipient objects in the application : " + Recipient.getObjectCount()+"\n");

**break**;

**default**:

System.out.println("Invalid Input.!!!\n");

}

}

}

// start email client

// code to create objects for each recipient in clientList.txt

// use necessary variables, methods and classes

}

//create more classes needed for the implementation (remove the public access modifier from classes when you submit your code)