МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем

та мереж

**Звіт**  до лабораторної роботи №5

з дисципліни “Екстремальне програмування ”

**Програмний функціонал**

Виконала: студентка гр. КН-311 Боштан А.О.

Прийняв:

Щербак С.С.

Львів – 2020

Теоретичні відомості

**Об'єктно-орієнтоване програмування**— одна з [парадигм](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [програмування](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка розглядає програму як множину [«об'єктів»](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), що взаємодіють між собою. Основу ООП складають чотири основні концепції: [інкапсуляція](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D0%BA%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D1%8F%D1%86%D1%96%D1%8F), [успадкування](https://uk.wikipedia.org/wiki/%D0%A3%D1%81%D0%BF%D0%B0%D0%B4%D0%BA%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), [поліморфізм](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D0%BC%D0%BE%D1%80%D1%84%D1%96%D0%B7%D0%BC_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) та абстракція. Одною з переваг ООП є краща [модульність](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) програмного забезпечення (тисячу функцій процедурної мови, в ООП можна замінити кількома десятками класів із своїми методами). Попри те, що ця парадигма з'явилась в [1960-тих](https://uk.wikipedia.org/wiki/1960-%D1%82%D1%96) роках, вона не мала широкого застосування до [1990-тих](https://uk.wikipedia.org/wiki/1990-%D1%82%D1%96), коли розвиток комп'ютерів та комп'ютерних мереж дав змогу писати надзвичайно об'ємне і складне програмне забезпечення, що змусило переглянути підходи до написання програм. Сьогодні багато [мов програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) або підтримують ООП ([PHP](https://uk.wikipedia.org/wiki/PHP), [Lua](https://uk.wikipedia.org/wiki/Lua)) або ж є цілком [об'єктно-орієнтованими](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) (зокрема, [Java](https://uk.wikipedia.org/wiki/Java_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), [C#](https://uk.wikipedia.org/wiki/C_Sharp_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), [C++](https://uk.wikipedia.org/wiki/C%2B%2B), [Python](https://uk.wikipedia.org/wiki/Python), [Ruby](https://uk.wikipedia.org/wiki/Ruby) і [Objective-C](https://uk.wikipedia.org/wiki/Objective-C), [ActionScript 3](https://uk.wikipedia.org/wiki/ActionScript), [Swift](https://uk.wikipedia.org/wiki/Swift_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), [Vala](https://uk.wikipedia.org/wiki/Vala)).

На відміну від традиційних поглядів, коли програму розглядали як набір [підпрограм](https://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%B4%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0), або як перелік [інструкцій](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D1%96%D1%8F) комп'ютеру, ООП-програми можна вважати сукупністю об'єктів. Відповідно до парадигми об'єктно-орієнтованого програмування, кожен об'єкт здатний отримувати [повідомлення](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BC%D1%96%D0%BD_%D0%BF%D0%BE%D0%B2%D1%96%D0%B4%D0%BE%D0%BC%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F%D0%BC%D0%B8), обробляти дані, та надсилати повідомлення іншим об'єктам. Кожен об'єкт — своєрідний незалежний автомат з окремим призначенням та відповідальністю.

**Java Mail**

JavaMail складається з допомоги для синтаксичного аналізу стану доставки повідомлень. Є деякі стратегії та евристики для обробки цієї проблеми. одна з стратегій, Variable Envelope повернутися Paths. Ви можете встановити зворотний шлях в enveloper як доведено в прикладі нижче. тобто адреса, де лист відмов направляються. вам може знадобитися, щоб встановити це в загальний адреса, відмінний ніж From: заголовка, так що ви можете метод дистанційного відскакує. Це робиться за допомогою настройки mail.smtp.from активів в JavaMail.

Частина коду Mail.java

**package** service;  
  
**import** model.Order;  
  
**import** javax.mail.\*;  
**import** javax.mail.internet.InternetAddress;  
**import** javax.mail.internet.MimeMessage;  
**import** java.io.UnsupportedEncodingException;  
**import** java.util.Properties;  
  
**public class** Mail {  
 **public static void** sendMail(Order order) {  
 **try** {  
 Properties props = **new** Properties();  
 props.put(**"mail.smtp.host"**, **"smtp.gmail.com"**);  
 props.put(**"mail.smtp.auth"**, **"true"**);  
 props.put(**"mail.debug"**, **"false"**);  
 props.put(**"mail.smtp.ssl.enable"**, **"true"**);  
 props.put(**"mail.smtp.socketFactory.class"**, **"javax.net.ssl.SSLSocketFactory"**);  
  
 Session session = Session.*getInstance*(props, **new** EmailAuth());  
 Message msg = **new** MimeMessage(session);  
 InternetAddress from = **new** InternetAddress(**"ostap.tuz.kn.2017@lpnu.ua"**, **"DSNS"**);  
 msg.setFrom(from);  
 InternetAddress toAddress = **new** InternetAddress(**"ostap.tuz.kn.2017@lpnu.ua"**);  
 msg.setRecipient(Message.RecipientType.***TO***, toAddress);  
 msg.setSubject(**"Reservation notification"**);  
  
 String specialOccasion = (order.isSpecialOccasion()) ? **"Yes"** : **"No"**;  
 String separateRoom = (order.isSeparateRoom()) ? **"Yes"** : **"No"**;  
 msg.setContent(**"<html>\n"** +  
 **"<body>\n"** +  
 **" <div style=\"width: 400px; background-color: whitesmoke; border: 1px solid black; border-radius: 5px;\">\n"** +  
 **" <h3 style=\"text-align: center\">Reservation notification</h3>\n"** +  
 **" <div style=\"margin-left: 20px\">\n"** +  
 **" <p>Name: "** + order.getUser().getName() + **"</p>\n"** +  
 **" <p>Surname: "** + order.getUser().getSurname() + **"</p>\n"** +  
 **" <p>Email: "** + order.getUser().getEmail() + **"</p>\n"** +  
 **" <p>Phone: "** + order.getUser().getPhone() + **"</p>\n"** +  
 **" <p>Date of reservation: "** + order.getDate() + **"</p>\n"** +  
 **" <p>Amount of people: "** + order.getPersonAmount() + **"</p>\n"** +  
 **" <p>Special occasion: "** + specialOccasion + **"</p>\n"** +  
 **" <p>Separate room: "** + separateRoom + **"</p>\n"** +  
 **" <p>Email: "** + order.getHall().name() + **"</p>\n"** +  
 **" <p>Addition advise: "** + order.getAdditionalAdvise() + **"</p>\n"** +  
 **" <p>-------------------------------------------------</p>\n"** +  
 **" <p>Price: "** + order.calculatePrice() + **"$</p>\n"** +  
 **" </div>\n"** +  
 **" </div>\n"** +  
 **"</body>\n"** +  
 **"</html>"**, **"text/html"**);  
 Transport.*send*(msg);  
 } **catch** (UnsupportedEncodingException ex) {  
 ex.printStackTrace();  
 } **catch** (MessagingException ex) {  
 ex.printStackTrace();  
 }  
 }

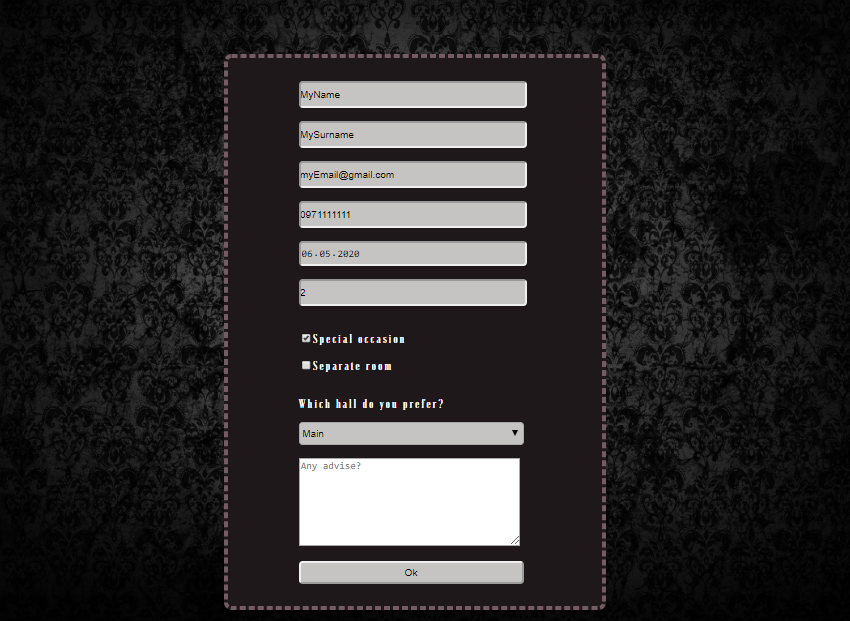
****

Рис.1 Форма з інформацією для відправлення



Рис.2 Дата резервації зайнята

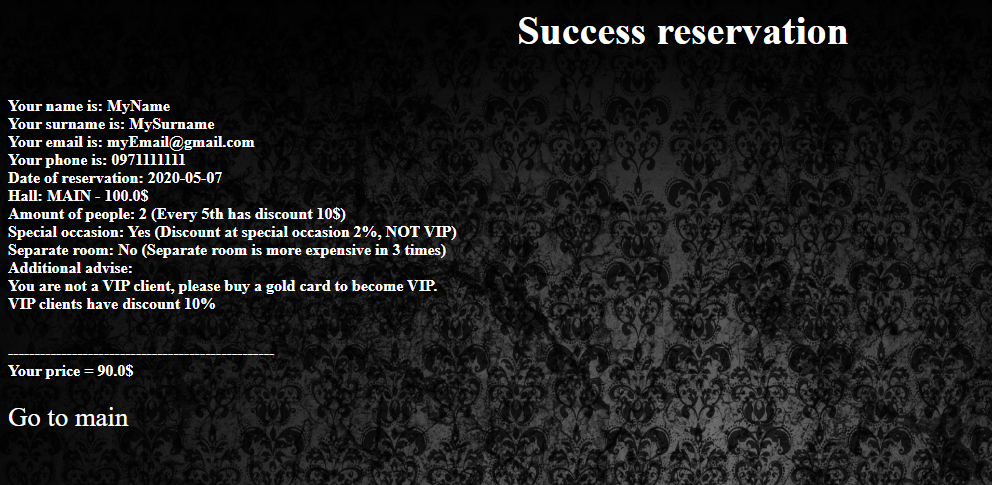


Рис.3 Резервація успішна

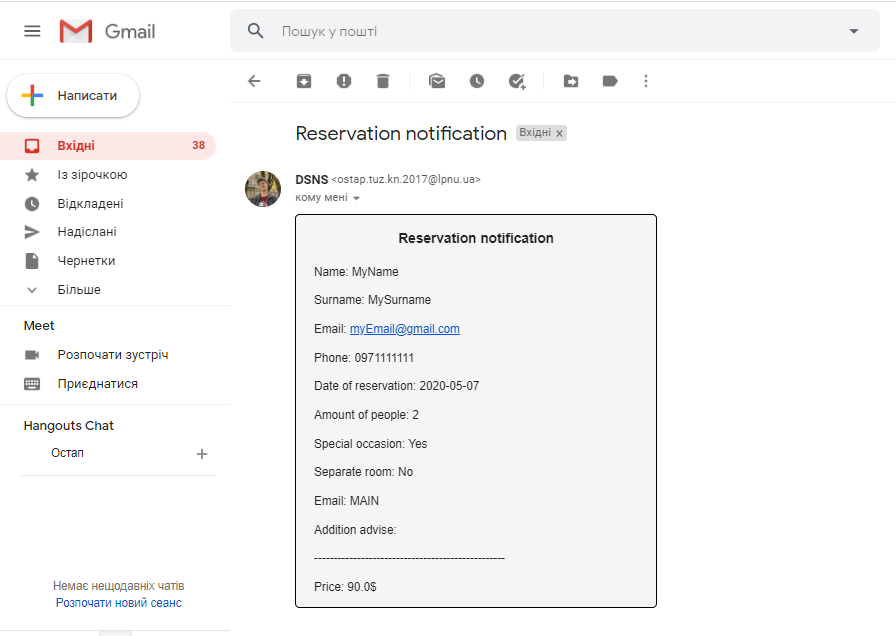


Рис.4 Форму успішно відравлено на пошту

Для відправлення повідомлення використано JavaMail API. Також для зручної роботи із даними і бізнес логікою використано об’єктно-орієнтоване програмування. Створено такі об’єкти як User, Order та інші.

Щодо надсилання повідомлення то було створено сесію, куди передано відповідні параметри, наприклад smtp сервер. В якості SMTP сервера в даній лабораторній роботі використано smtp.gmail.com. Після цього формується лист і вказується отримувач. Коли об’єкт листа сформовано її відсилають за допомогою функції send.

Висновок: у даній лабораторній роботі було ознайомлено з базовими поняттями JavaMail, а також реалізовано значну частину бекенду веб-сайту, зокрема відправлення форми резервації на електронну пошту.