**HaNoi University of science and technology**

**2020-2021**

**Object Oriented Programming Report**

***Exercise 18: Shipping System***

***(Java swing + MySql)***

Subject: **Object Oriented Programming**

Teacher:

Class:

**Group**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Name | Student ID | Email | Work assignment | Completion | Note |
| 01 | Nguyễn Văn A | 01200210 |  | - Overview OOP in Java + Mysql  - Coding the first function Model  - Write Report  - Discuss and divide work for all members  - Research for connecting MySql | - 100% | IDE:NetBean  CSDL:MySql |
| 02 | Nguyễn văn B | 354603134 |  | - Research for java swing,  - Coding the second function Controller  - Coding the third function View  - Draw UML | -100% | Java swing |

**TABLE OF CONTENTS**

[**I.** **Knowledge Base 3**](#_Toc18293)

[**II. Shipping System**](#_Toc18294) **6**

[**1. System Introduce**](#_Toc18295) **6**

[**2. Model**](#_Toc18296) **8**

[**3. Controller**](#_Toc18298) **10**

[**4. View**](#_Toc18300) **11**

[**III.** **Result**](#_Toc18306) **14**

[**IV. Summary**](#_Toc18318) **15**

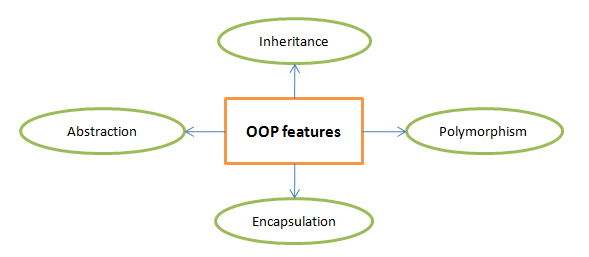
[**V. References**](#_Toc18319) **15**

**I.** **Knowledge Base**

**1. Java OOP:**

Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

Object-oriented programming is based on the following principles:



### **Graphical User Interface – GUI**

A user interface (GUI for short) is a window containing controls such as buttons, text boxes, combo boxes, etc. The user interacts with the interface using a mouse, pointer, or keyboard.

Java allows us to create user interfaces using one of two packages, AWT or Swing. AWT (Asbtract Window Toolkit) is a subset of Swing, so Swing is mainly used, but in Java programs, both Swing and AWT packages must be imported.

## MySQL :

MySQL is an open source database management system (Relational Database Management System, abbreviated as RDBMS) that operates under the client-server model. RDBMS is a software or service used to create and manage databases (Database) in the form of managing relationships between them.

# Connect java to mysql using JDBC

JDBC (Java Database Connectivity) is an API standard (Application Program Interface) that allows connecting programs written in Java with database management systems (MySQL, MS SQL, Postgre SQL, Oracle, DB2 ...)

JDBC is just a set of interfaces, definitions, error messages, and specifications, not a library. For each database management system, we will have a separate installation of JDBC for it, for example JDBC for MySQL, JDBC for MS SQL ...

**II.** **Shipping System**

**1. System introduce**

**Link source code:** <https://github.com/nguyenhv01012000/IT3100Q_121723_Project_Group01_PN04>

***Exercise 18: Shipping System***

Managing a shipping company: A shipping company receives goods for customers. Have Two types of shipping are documents and parcels. Shipping bill information stored includes person information sender, receiver, send time, receive time. Shipping charges for documents are calculated according to sending and receiving distance in km \* 2,000 VND/km + 12,000 VND. Shipping charges for parcels are charged according to the distance of sending and receiving in km \* 2,000 VND/km + parcel weight in kg \* 10,000 VND. Construction

Develop a management program that includes the following functions:

- Add, edit, delete a shipping order.

- Search for shipping orders by recipient's address.

- Statistics of shipping orders sent to the destination during the input period (from

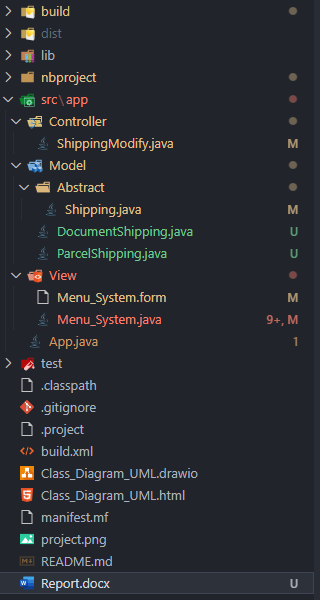
day/month/year to day/month/year).

- Update shipping costs for 1 km, for 1kg, additional costs for documents when required by the company

- Calculate the shipping company's revenue on the amount received from shipping orders in one

input period (from day/month/year to day/month/year).

**Files of project:**



**The system is made according to the MVC .**

MVC stands for 3 words: Model - View - Controller, is a design pattern that aims to separate the interface and code for easy management, development and maintenance.

The MVC pattern divides an application into three interoperable parts, each with a separate and independent task from the other components.

### • Model: data management and processing

### • View: interface, display data for users

### • Controller: controls the interaction between Model and View

### **Technologies**

* **JAVA SWING**:
* **MYSQL**:

### **Install**

* IDE: Netbean,
* CSDL: MySql

### **Runs**

1. Create name's database in MySql : mydatabase
2. if user = root, password = 123456 in MySql : (ShippingModify.java)
3. connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase", "root", "123456");
4. else edit respectively ( user = “ …..” , password = “ …” )
5. connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase", "....", "......");
6. Run project: run App.java

**--> App will run**

**App.java**

public class App {

*/\*\**

*\* @param args the command line arguments*

*\*/*

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

*// TODO code application logic here*

        Menu\_System view = new Menu\_System();

        view.show();

    }

}

**2. Model**

• Responsible for data management of the application

• Contains all business logic, data description object

• Notify the view to show the results back to the user

File:

* Shipping.java
* DocumentShipping.java
* ParcelShipping.java

**3. Controller**

• Define the system's operations and processes

• Collating user actions from the View. Also interact with the Model to call the View and display the corresponding information to the user.

File:

* ShippingModify.java

1. **View**

• Contains the interface, interacts with the user, uses the Model to display the results to the user

• Output results from Controller layer

• Receive user activities and requests and pass them to the Controller layer for processing

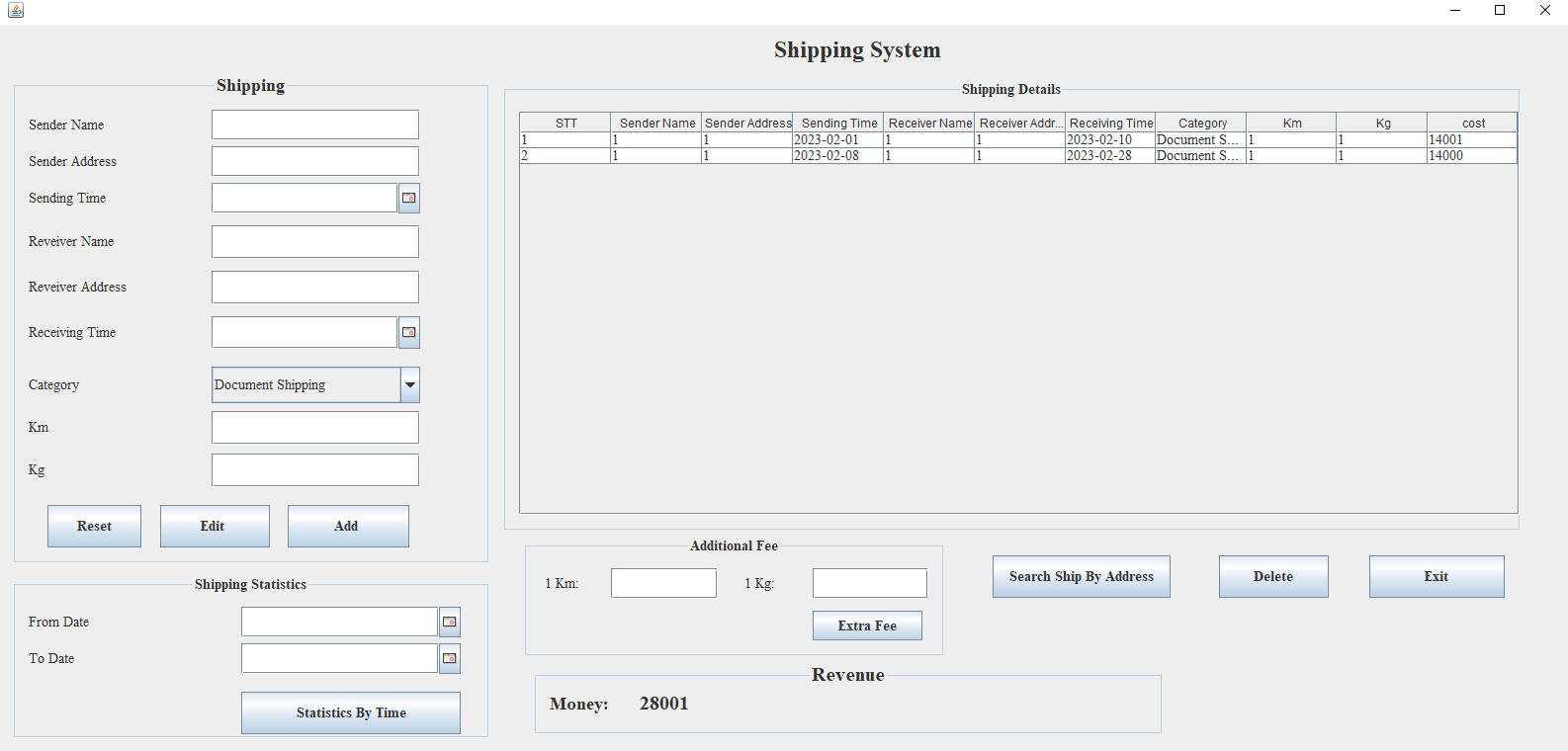
File:

* Menu\_System.java.java

# 

**III. Result**

**When running the program will display:**

****

- Using aggregation/composition for your own class

public class Shipping {

    protected User Sender;

    protected User Receiver;

- Using method overloading for your own class

public void setCost(String *cost*) {

        this.cost = *cost*;

    }

public void setCost(String *km*, String *kg*){};

- Using inheritance for your own class

public class DocumentShipping extends Shipping

public class ParcelShipping extends Shipping

- Using method overriding for your class

public class DocumentShipping extends Shipping{

    public DocumentShipping(int *id*, String *senderName*, String *senderAddress*, String *sendingTime*, String *receiverName*, String *receiverAddress*, String *receivingTime*, String *category*, String *km*, String *kg*, String *cost*) {

        super(*id*, *senderName*, *senderAddress*, *sendingTime*, *receiverName*, *receiverAddress*, *receivingTime*, *category*, *km*, *kg*, *cost*);

    }

    @Override

    public void setCost(String *km*, String *kg*) {

        this.cost = String.valueOf(12000 + 2000\*Integer.parseInt(*km*));

    }

}

public class ParcelShipping extends Shipping{

    public ParcelShipping(int *id*, String *senderName*, String *senderAddress*, String *sendingTime*, String *receiverName*, String *receiverAddress*, String *receivingTime*, String *category*, String *km*, String *kg*, String *cost*) {

        super(*id*, *senderName*, *senderAddress*, *sendingTime*, *receiverName*, *receiverAddress*, *receivingTime*, *category*, *km*, *kg*, *cost*);

    }

    @Override

    public void setCost(String *km*, String *kg*) {

        this.cost = String.valueOf(12000 + 2000\*Integer.parseInt(*km*));

    }

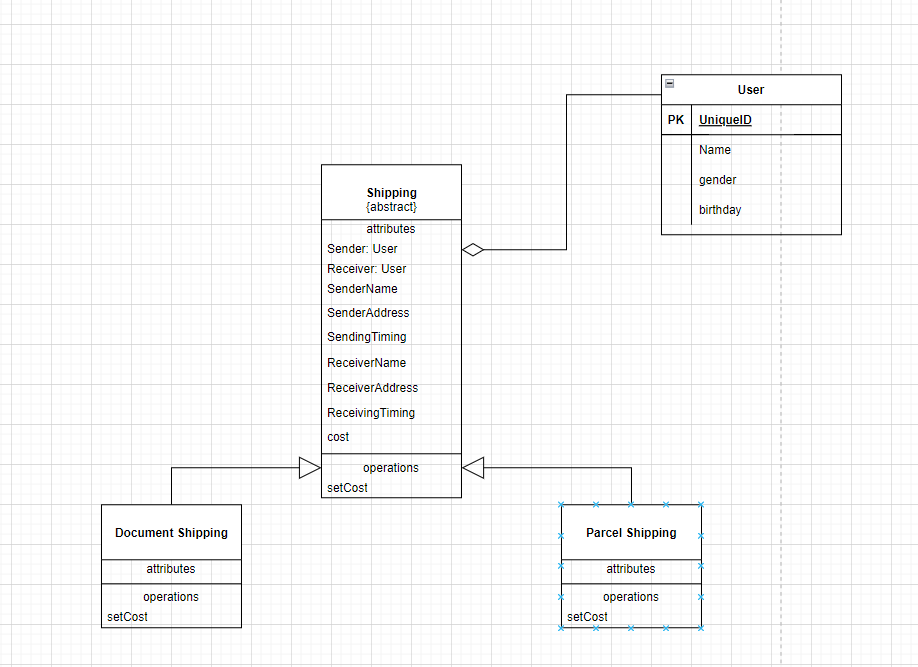
}

- Using up casting/down casting in your own class

    Shipping shipping1 = (Shipping) new DocumentShipping();

Shipping shipping2 = (Shipping) new ParcelShipping();

**Class\_Diagram\_UML**

****

**IV. Summary**

With an extremely limited amount of knowledge and not much time for in-depth research, this report does not hope to bring anything new, but simply summarizes, translates and revises articles. writing is available on the internet, printed books and a number of other materials. Errors in content as well as defects in presentation are definitely inevitable, looking forward to receiving your understanding and help to make the report more complete!

**V. References**

* [The JavaTM Tutorials](http://java.sun.com/docs/books/tutorial/index.html)
* [Sun Developer Network](http://java.sun.com/reference/docs/)
* <https://teamvietdev.com/chia-se-bo-tai-lieu-lap-trinh-java-swing/>
* https://www.mysqltutorial.org/
* …