

TRƯỜNG ĐẠI HỌC VĂN LANG

KHOA: CNTT

Mã học phần: 72ITSE40803

Mã nhóm lớp HP: 231\_72ITSE40803\_01

Thời gian làm bài: 75 (phút)

Hình thức thi: **Thực hành**

ĐÁP ÁN ĐỀ THI KẾT THÚC HỌC PHẦN

Học kỳ: 1 Năm học: 2023 - 2024

Tên học phần: Lập trình Java nâng cao

**Câu 1 (2 điểm):** Tạo cơ sở dữ liệu trên MySQL theo cấu trúc dữ liệu trong tập tin đính kèm.

```
Create database mycompany
go
CREATE TABLE mycompany.product(
  idproduct VARCHAR(12) NOT NULL,
  name VARCHAR(45) NULL,
  price INT NULL,
  PRIMARY KEY (idproduct));
INSERT INTO mycompany.product values('1111','Pepsi',12035);
INSERT INTO mycompany.product values('2222','CocaCola',21035);
INSERT INTO mycompany.product values('3333','Fanta',11035);
INSERT INTO mycompany.product values('4444','Sprite',32035);
INSERT INTO mycompany.product values('5555','Orange',52035);
GO
CREATE TABLE mycompany.staff(
  idStaff VARCHAR(12) NOT NULL,
  name VARCHAR(45) NULL,
  PRIMARY KEY (idStaff));
INSERT INTO mycompany.staff VALUES('nv001','Nguyen Hoang Nam');
INSERT INTO mycompany.staff VALUES('nv002','Hoang Dung');
INSERT INTO mycompany.staff VALUES('nv003','Quach Tinh');
INSERT INTO mycompany.staff VALUES('nv004','Chau Ba Thong');
INSERT INTO mycompany.staff VALUES('nv005','Hong That Cong');
GO
CREATE TABLE mycompany.customer(
  idCustomer VARCHAR(12) NOT NULL,
  name VARCHAR(45) NULL,
  PRIMARY KEY (idCustomer));
INSERT INTO mycompany.customer VALUES('kh001','Muong Thanh');
INSERT INTO mycompany.customer VALUES('kh002','WinFast');
INSERT INTO mycompany.customer VALUES('kh003','Toyota');
INSERT INTO mycompany.customer VALUES('kh004','Sony');
INSERT INTO mycompany.customer VALUES('kh005','Honda');
GO
CREATE TABLE mycompany.order(
  idOrder VARCHAR(12) NOT NULL,
  idStaff VARCHAR(12) NULL,
  idCustomer VARCHAR(12) NULL,
  orderDate VARCHAR(12) NULL,
  PRIMARY KEY (idOrder),
  CONSTRAINT FK_Staf_Order FOREIGN KEY (idStaff) REFERENCES mycompany.staff(idStaff) ON DELETE
NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT FK_Customer_Order FOREIGN KEY (idCustomer) REFERENCES
mycompany.customer(idCustomer) ON DELETE NO ACTION ON UPDATE NO ACTION);

CREATE TABLE mycompany.orderdetail(
  idOrder VARCHAR(12) NOT NULL,
  idProduct VARCHAR(12) NOT NULL,
```

price INT NULL,  
 quantity INT NULL,  
 PRIMARY KEY (idOrder, idProduct),  
 CONSTRAINT FK\_Detail\_Order FOREIGN KEY (idOrder) REFERENCES mycompany.order(idOrder) ON DELETE NO ACTION ON UPDATE NO ACTION,  
 CONSTRAINT FK\_Detail\_Product FOREIGN KEY (idProduct) REFERENCES mycompany.product(idproduct) ON DELETE NO ACTION ON UPDATE NO ACTION);

**Câu 2 (1 điểm):** Thiết kế giao diện nhập hóa đơn theo mẫu như sau:

The screenshot shows a window titled "Add new Order" with the following components:

- INPUT ORDER** header.
- Sales Staff:** A dropdown menu and a text field containing "Nguyen Hoang Nam".
- Customer:** A dropdown menu and a text field containing "Muong Thanh".
- Code of order:** A text field containing "hd003".
- Date of order:** A date picker showing "Nov 14, 2023".
- List of Product:** An empty list box.
- Product details:** Fields for "Id code:" (4444), "Name:" (Sprite), "Price:" (32035), and "Quantity:" (17). An "Add" button is next to the quantity field.
- Table:** A table with columns: Code of product, Name of product, Price, Quantity, Total.
- Buttons:** "Close", "Save", and "Sum:" (with a value field).

**Câu 3 (7 điểm):** Thực hiện các chức năng xử lý như sau:

The screenshot shows the same window as above, but with the following changes:

- List of Product:** A list containing "[1111] Pepsi", "[2222] CocaCola", "[4444] Sprite" (highlighted), and "[5555] Orange".
- Table:** The table now contains three rows of data:
 

Code of product	Name of product	Price	Quantity	Total
5555	Orange	52035	9	468315
2222	CocaCola	21035	12	252420
4444	Sprite	32035	17	544595
- Sum:** The "Sum:" field now displays "1265330".

1. Truy vấn cơ sở dữ liệu lấy danh sách nhân viên đưa vào JComboBox (Sales Staff) và khi chọn nhân viên thì họ tên sẽ hiển thị trong JTextField bên cạnh. (1 điểm).

```
public class ClassProduct {
    String idCode;
    String name;
    int price;
```

```

public ClassProduct(String idCode, String name, int price) {
    this.idCode = idCode;
    this.name = name;
    this.price = price;
}

public ClassProduct() {
    this.idCode = null;
    this.name = null;
    this.price = 0;
}

public String getIdCode() {
    return idCode;
}

public void setIdCode(String idCode) {
    this.idCode = idCode;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getPrice() {
    return price;
}

public void setPrice(int price) {
    this.price = price;
}

}

public ArrayList<ClassProduct> getAllProductFromDB() {
    ResultSet result = null;
    String sql = "Select * from mycompany.Product";
    try{
        Connection conn = connectDB();
        Statement st = conn.createStatement();
        result = st.executeQuery(sql);
        while(result.next()) {
            ClassProduct pro = new ClassProduct();
            pro.setIdCode(result.getString("idproduct"));
            pro.setName(result.getString("name"));
            pro.setPrice(Integer.parseInt(result.getString("price")));
            this.list_Pro.add(pro);
        }
    } catch(Exception ex){
        JOptionPane.showMessageDialog(null,ex,"Notification", JOptionPane.INFORMATION_MESSAGE);
    }
    return this.list_Pro;
}

comboBox_Empl = new JComboBox();
comboBox_Empl.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
    }
}

```

```

        int index = comboBox_Empl.getSelectedIndex();
        textField_Empl.setText(staList.get(index).getName());
    }
});
comboBox_Empl.setBackground(new Color(255, 255, 255));
comboBox_Empl.setBounds(10, 87, 78, 22);
contentPane.add(comboBox_Empl);

textField_Empl = new JTextField();
textField_Empl.setBackground(new Color(255, 255, 255));
textField_Empl.setFont(new Font("Tahoma", Font.PLAIN, 12));
textField_Empl.setEnabled(false);
textField_Empl.setBounds(91, 87, 169, 23);
contentPane.add(textField_Empl);
textField_Empl.setColumns(10);

```

2. Truy vấn cơ sở dữ liệu lấy danh sách khách hàng đưa vào JComboBox (Customer) và khi chọn khách hàng thì tên sẽ hiển thị trong JTextField bên cạnh. (1 điểm).

```

import java.util.Date;

public class ClassEmployee {
    String idEmployee;
    String name;
    String gender;
    Date birthDay;
    String address;
    String phoneNumber;

    public ClassEmployee(String idEmployee, String name, String gender,
        Date birthDay, String address, String phoneNumber) {
        this.idEmployee = idEmployee;
        this.name = name;
        this.gender = gender;
        this.birthDay = birthDay;
        this.address = address;
        this.phoneNumber = phoneNumber;
    }

    public ClassEmployee() {
        this.idEmployee = null;
        this.name = null;
        this.gender = null;
        this.birthDay = new Date();
        this.address = null;
        this.phoneNumber = null;
    }

    public String getIdEmployee() {
        return idEmployee;
    }

    public void setIdEmployee(String idEmployee) {
        this.idEmployee = idEmployee;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {

```

```

        this.name = name;
    }

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public Date getBirthDay() {
        return birthDay;
    }

    public void setBirthDay(Date birthDay) {
        this.birthDay = birthDay;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }
}

public ArrayList<ClassStaff> getAllStaffFromDB() {
    ResultSet result = null;
    String sql = "Select * from mycompany.Staff";
    try{
        Connection conn = connectDB();
        Statement st = conn.createStatement();
        result = st.executeQuery(sql);
        while(result.next()) {
            ClassStaff sta = new ClassStaff();
            sta.setIdStaff(result.getNString("idStaff"));
            sta.setName(result.getString("name"));
            this.list_Sta.add(sta);
        }
    } catch(Exception ex){
        JOptionPane.showMessageDialog(null,ex,"Notification", JOptionPane.INFORMATION_MESSAGE);
    }
    return this.list_Sta;
}

comboBox_Cust = new JComboBox();
comboBox_Cust.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        int index = comboBox_Cust.getSelectedIndex();
        textField_Cust.setText(cusList.get(index).getName());
    }
}

```

```

});
comboBox_Cust.setBounds(10, 156, 78, 22);
contentPane.add(comboBox_Cust);

textField_Cust = new JTextField();
textField_Cust.setBackground(new Color(255, 255, 255));
textField_Cust.setEnabled(false);
textField_Cust.setColumns(10);
textField_Cust.setBounds(91, 155, 169, 23);
contentPane.add(textField_Cust);

```

3. Truy vấn cơ sở dữ liệu lấy danh sách sản phẩm đưa vào JList (List of Product) và khi chọn sản phẩm thì thông tin liên quan sẽ hiển thị trong các JTextField tương ứng bên cạnh. (1 điểm).

```

public class ClassCustomer {
    String idCustomer;
    String name;
    String address;
    String phoneNumber;
    public ClassCustomer(String idCustomer, String name, String address, String phoneNumber) {
        this.idCustomer = idCustomer;
        this.name = name;
        this.address = address;
        this.phoneNumber = phoneNumber;
    }

    public ClassCustomer() {
        this.idCustomer = null;
        this.name = null;
        this.address = null;
        this.phoneNumber = null;
    }

    public String getIdCustomer() {
        return idCustomer;
    }

    public void setIdCustomer(String idCustomer) {
        this.idCustomer = idCustomer;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }
}

```

```

    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }
}

public ArrayList<ClassCustomer> getAllCustomerFromDB() {
    ResultSet result = null;
    String sql = "Select * from mycompany.Customer";
    try{
        Connection conn = connectDB();
        Statement st = conn.createStatement();
        result = st.executeQuery(sql);
        while(result.next()) {
            ClassCustomer cus = new ClassCustomer();
            cus.setIdCustomer(result.getNString("idCustomer"));
            cus.setName(result.getString("name"));
            this.list_Cus.add(cus);
        }
    } catch(Exception ex){
        JOptionPane.showMessageDialog(null,ex,"Notification", JOptionPane.INFORMATION_MESSAGE);
    }
    return this.list_Cus;
}

model_Pro = new DefaultListModel<>();
list_Pro = new JList(model_Pro);
list_Pro.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        int index = list_Pro.getSelectedIndex();
        textField_IdProduct.setText(proList.get(index).getIdCode());
        textField_NameProduct.setText(proList.get(index).getName());
        textField_Price.setText(proList.get(index).getPrice() + "");
    }
});
list_Pro.setBounds(10, 16, 164, 164);
panel.add(list_Pro);

```

#### 4. Khi bấm nút <Add> lấy các thông tin tương ứng đưa vào JTable (Detail of order) và tính giá trị cho Total = Price x Quantity. (1 điểm)

```

table_Order_Detail = new JTable();
table_Order_Detail.setBounds(413, 142, 592, 163);
scrollPane.setViewportView(table_Order_Detail);
table_Order_Detail.setModel(new DefaultTableModel(
    new Object[][] {
    },
    new String[] {
        "Code of product", "Name of product", "Price", "Quantity", "Total"
    }
));

JButton btnLeft = new JButton("Add");
btnLeft.setForeground(Color.BLUE);
btnLeft.setFont(new Font("Tahoma", Font.BOLD, 12));
btnLeft.setBackground(new Color(250, 128, 114));
btnLeft.setBounds(641, 195, 68, 23);
btnLeft.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        ClassOrderDetail detail = new ClassOrderDetail();
    }
});

```

```

detail.setIdOrder(textField_IdOrder.getText().toString());
detail.setIdProduct(textField_IdProduct.getText().toString());
detail.setName(textField_NameProduct.getText().toString());
detail.setPrice(Integer.parseInt(textField_Price.getText().toString()));
detail.setQuantity(Integer.parseInt(textField_Quantity.getText().toString()));
detailList.add(detail);
tbl_Model = (DefaultTableModel)table_Order_Detail.getModel();
tbl_Model.getDataVector().removeAllElements();
int sum = 0;
for(ClassOrderDetail det:detailList) {
    tbl_Model.addRow(new Object[] {
        det.getIdProduct(),
        det.getName(),
        det.getPrice(),
        det.getQuantity(),
        det.getPrice()*det.getQuantity()
    });
    sum += det.getTotal();
}
textField_Sum.setText(sum + "");
});
contentPane.add(btnLeft);

```

5. Viết xử lý cho nút <Save> sẽ lưu thông tin hóa đơn vào bảng Order và thông tin chi tiết hóa đơn vào bảng OrderDetail. (2 điểm)

```

public int addNewOrder(ClassOrder order) {
    int result = 0;
    String sql = "INSERT INTO mycompany.order VALUES(" +
        order.getIdOrder() + "," +
        order.getIdEmployee() + "," +
        order.getIdCustomer() + "," +
        order.getDateOrder() + ")";
    JOptionPane.showMessageDialog(null,sql,"Notification", JOptionPane.INFORMATION_MESSAGE);

    try{
        Connection conn = connectDB();
        Statement st = conn.createStatement();
        result = st.executeUpdate(sql);
        st.close();
        conn.close();
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null,ex,"Notification", JOptionPane.INFORMATION_MESSAGE);
    }

    return result;
}

public int addNewOrderDetail(ArrayList<ClassOrderDetail> detailList) {
    int result = 0;

    try{
        Connection conn = connectDB();
        Statement st = conn.createStatement();
        for(ClassOrderDetail det:detailList) {
            String sql = "INSERT INTO mycompany.orderdetail VALUES(" +
                det.getIdOrder() + "," +
                det.getIdProduct() + "," +
                det.getPrice() + "," +
                det.getQuantity() + ")";
            result = st.executeUpdate(sql);
        }
    }
}

```



```

        st.close();
        conn.close();
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex, "Notification", JOptionPane.INFORMATION_MESSAGE);
    }

    return result;
}
JButton btnSaveOrder = new JButton("Save");
btnSaveOrder.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        ClassOrder order = new ClassOrder();
        order.setIdOrder(textField_IdOrder.getText().toString());
        order.setIdEmployee(staList.get(comboBox_Empl.getSelectedIndex()).getIdStaff());
        order.setIdCustomer(cusList.get(comboBox_Cust.getSelectedIndex()).getIdCustomer());
        DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.SHORT, Locale.FRENCH);
        order.setDateOrder(dateFormat.format(dateChooser_Order.getDate()));
        util.addNewOrder(order);
        util.addNewOrderDetail(detailList);
        JOptionPane.showMessageDialog(null, "Save successfully!", "Notification",
            JOptionPane.INFORMATION_MESSAGE);
    }
});
btnSaveOrder.setForeground(Color.BLUE);
btnSaveOrder.setFont(new Font("Tahoma", Font.BOLD, 12));
btnSaveOrder.setBackground(new Color(250, 128, 114));
btnSaveOrder.setBounds(101, 366, 78, 23);
contentPane.add(btnSaveOrder);

```

## 6. Viết xử lý nút <Close> đóng form và tính ô Sum là tổng cột Total. (1 điểm)

```

JButton btnClose = new JButton("Close");
btnClose.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        dispose();
    }
});

```

Tính sum đúng: 0.5 điểm

*Ngày biên soạn: 19/11/2023*

**Giảng viên biên soạn đề thi: TS. Huỳnh Hữu Nghĩa**

*Ngày kiểm duyệt: 23/11/2023*

**Trưởng (Phó) Khoa/Bộ môn kiểm duyệt đề thi: TS. Bùi Minh Phụng**