# 1．引言

## 1.1编写目的

为编码提供方便

## 1.2项目背景

为了方便店员对自己的商品的管理，需要编写一个商品管理的小程序，能够对自己的库存进行管理。

# 2．总体设计

## 2.1需求概述

登录功能，学生信息管理功能，设置功能（能够更改帐号密码）

登录功能，为了保持安全性，店员可以保存自己的帐号信息，通过帐号密码才能够进入信息管理界面；

商品的管理功能，能够对商品进行增加，删除，修改，查询

保密设置，店员可以修改自己的帐号密码，保证信息的安全性

## 2.2软件结构



# 3．程序描述

## 3.1功能

商品的管理功能，能够对商品进行增加，删除，修改，查询

商品管理功能，记录进货、出货与库存。

## 3.2性能

时间特性要求：等待时间<1s

## 3.3输入项目

选择所需操作，输入必要的商品信息，进行相关操作

## 3.4输出项目

前台管理模块：输出商品增加，修改，删除，查找服务，

系统维护模块：根据需要选择下一步的增删改操作进行操作

## 3.5算法

主页面

**package** UI;

**import** java.awt.BorderLayout;

**public** **class** Main **extends** JFrame {

**private** JPanel contentPane;

Main gui = **this**;

**public** **static** List *list*;

**private** Main main;

**private** JTable table;

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

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

javax.swing.UIManager.*setLookAndFeel*("com.jtattoo.plaf.mcwin.McWinLookAndFeel");

LogonUI logon=**new** LogonUI();

logon.setModal(**true**);

logon.setVisible(**true**);

**if**(logon.getLogonResult()==-1)

System.*exit*(0);

Main main=**new** Main();

main.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** Main() {

main=**this**;

setTitle("\u5546\u54C1\u7BA1\u7406\u7CFB\u7EDF");

setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

setBounds(100, 100, 757, 391);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JButton btnNewButton = **new** JButton("\u589E\u52A0");

btnNewButton.setFont(**new** Font("宋体", Font.*PLAIN*, 16));

btnNewButton.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**try** {

AddGoods addGoods=**new** AddGoods(table,main);

addGoods.setVisible(**true**);

} **catch** (Exception e1) {

e1.printStackTrace();

}

}

});

btnNewButton.setBounds(42, 46, 111, 23);

contentPane.add(btnNewButton);

JButton btnNewButton\_1 = **new** JButton("\u66F4\u6539\u4FE1\u606F");

btnNewButton\_1.setFont(**new** Font("宋体", Font.*PLAIN*, 16));

btnNewButton\_1.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**if**(getTable().getSelectedRowCount() == 0)

{

JOptionPane.*showConfirmDialog*(**null**, "请选择要修改的记录!","提示", JOptionPane.*CLOSED\_OPTION*);

}

**else**

{

List<Goods>list1=Control.GoodsControl.*getControl*().findAll();

**for**(**int** i = getTable().getSelectedRow();i < (getTable().getSelectedRow() +

getTable().getSelectedRowCount());i++)

**try** {

ModifyGoods modifyGoods=**new** ModifyGoods(main,(Goods)*list*.get(i));

modifyGoods.setVisible(**true**);

} **catch** (Exception e1) {

e1.printStackTrace();

}

}

}

});

btnNewButton\_1.setBounds(42, 99, 111, 23);

contentPane.add(btnNewButton\_1);

JButton btnNewButton\_2 = **new** JButton("\u5220\u9664");

btnNewButton\_2.setFont(**new** Font("宋体", Font.*PLAIN*, 16));

btnNewButton\_2.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

DefaultTableModel tableModel = (DefaultTableModel) getTable().getModel();

**if**(getTable().getSelectedRowCount() == 0)

{

JOptionPane.*showConfirmDialog*(**null**, "请选择要删除的记录!","提示", JOptionPane.*CLOSED\_OPTION*);

}

**else** **if**(getTable().getSelectedRowCount() > 0)

{

**int** choice = JOptionPane.*showConfirmDialog*(**null**, "选中的记录将被删除!","是否删除:", JOptionPane.*OK\_CANCEL\_OPTION*);

**if**(choice == 0)

{

List<Goods>list1=Control.GoodsControl.*getControl*().findAll();

**for**(**int** i = getTable().getSelectedRow();i < (getTable().getSelectedRow() +

getTable().getSelectedRowCount());i++)

{

tableModel.removeRow(i);

Control.GoodsControl.*getControl*().delete((Goods)*list*.get(i));

}

InitialTable();

}

}

}

});

btnNewButton\_2.setBounds(42, 168, 111, 23);

contentPane.add(btnNewButton\_2);

JScrollPane scrollPane = **new** JScrollPane();

scrollPane.setBounds(189, 41, 536, 273);

contentPane.add(scrollPane);

table = **new** JTable(){

**public** **boolean** isCellEditable(**int** row, **int** column) {

**return** **false**;

}

};

table.setFont(**new** Font("宋体", Font.*PLAIN*, 16));

table.setModel(**new** DefaultTableModel(

**new** Object[][] {

},

**new** String[] {

"编号", "商品名字", "数量","价格"

}

));

**int** selMode = table.getSelectionModel().getSelectionMode();

table.setSelectionMode(ListSelectionModel.*SINGLE\_SELECTION*);

scrollPane.setViewportView(table);

JButton button = **new** JButton("\u6761\u4EF6\u67E5\u627E");

button.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**try** {

Search search=**new** Search(table);

search.setVisible(**true**);

} **catch** (Exception e1) {

e1.printStackTrace();

}

}

});

button.setFont(**new** Font("宋体", Font.*PLAIN*, 16));

button.setBounds(42, 229, 113, 27);

contentPane.add(button);

JButton button\_1 = **new** JButton("\u663E\u793A\u5168\u90E8");

button\_1.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

InitialTable();

}

});

button\_1.setFont(**new** Font("宋体", Font.*PLAIN*, 16));

button\_1.setBounds(42, 287, 113, 27);

contentPane.add(button\_1);

JButton button\_2 = **new** JButton("\u767B\u9646\u7BA1\u7406");

button\_2.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**try** {

SettingUI setting=**new** SettingUI();

setting.setVisible(**true**);

} **catch** (Exception e1) {

e1.printStackTrace();

}

}

});

button\_2.setFont(**new** Font("宋体", Font.*PLAIN*, 16));

button\_2.setBounds(612, 8, 113, 27);

contentPane.add(button\_2);

InitialTable();

}

**public** JTable getTable() {

**return** table;

}

**public** **void** setTable(JTable table) {

**this**.table = table;

}

**public** **void** InitialTable()

{

DefaultTableModel tableModel = (DefaultTableModel) getTable().getModel();

**if**(tableModel.getRowCount() != 0)

{

tableModel.setRowCount(0);

}

*list* = Control.GoodsControl.*getControl*().findAll();

**for**(**int** i = 0;i < *list*.size();i++)

{

Goods goods = (Goods)*list*.get(i);

tableModel.addRow(**new** Object[] {

goods.getId().toString(),goods.getName(),

goods.getCount().toString(),goods.getPrice().toString()});

}

setLocationRelativeTo(**null**);

}

}

**public** JTable getTable() {

**return** table;

}

**public** **void** setTable(JTable table) {

**this**.table = table;

}

## 3.6程序逻辑



#### 时序图：

#### （1）.商品管理时序图



（2）.商品消费时序图：



## 3.8存储分配

以上各个模块都需要数据库中的表，表中存储有关数据

goods表存储商品相关信息，att表存储商品库存信息

## 3.9限制条件

使用人员必须输入正确的数据且按照提示进行操作

## 3.10测试要点

重点检查各种值为空时的情况，检查数字类型数据的情况。