
目录

一、 背景分析.....	2
二、 需求分析.....	3
1. 数据分析.....	3
2. 功能分析.....	3
3. 其它要求.....	4
三、 概要设计.....	5
1. 数据字典.....	5
2. E-R 图.....	5
3. 关系数据库设计.....	6
四、 详细设计.....	8
1. 用户界面设计.....	8
2. 程序调用关系.....	9
五、 编码及实现.....	12
1. 工具选择.....	12
2. UML 类图.....	12
3. 重要 SQL 查询语句（部分）.....	17
4. 重要代码（部分）.....	18
5. 调试过程.....	23
六、 作品概览.....	25
1. 功能展示.....	25
2. 不足之处.....	28

暨南大学本科实验报告专用纸

一、背景分析

药品管理系统，是反映药品购进入库、在库管理、销售出库软件系统，是医院和医药销售企业药品流动信息的传递媒介和执行依据。

然而我们在调研中发现，目前已开发出来的药品管理系统虽然在使用过程中提供了良好的药品查询及其它使用功能，但是主要是面向医院、药品生产企业和药品销售公司，并没有一款针对药品代理商的药品管理系统。药品代理商作为连接药品生产企业和医院的中间渠道，在医药流通环节扮演着举足轻重的角色。药品代理商平时需要处理大量的药品进销数据，如果不能为他们设计一款便捷的药品管理系统，将会给药品流通带来很多不便，也为药品安全埋下隐患。

此外，现有的药品管理系统在功能上还有一些缺陷，比如药品供应企业的药品生产许可证或药品经营许可证已到期，系统不能及时警示；已有的药品查询系统中不能正确区别药品的质量类别和层次。这无疑不符合药品的 GSP 管理要求，给药品管理带来不安全隐患。

基于以上原因，我们面向药品代理商，开发一款增加企业证照审核系统和药品质量类别查询的药品管理系统，给医院及医药销售企业的药品管理提供安全保障及操作方便。

暨南大学本科实验报告专用纸(附页)

二、需求分析

在第一章，我们已经分析了用户对药品管理系统的迫切需求。因此，我们需要实现一个可以满足药品代理商基本需求的应用程序。在本章将分析应用程序需要具体实现哪些功能，需要包含那些数据。

1. 数据分析

在和一线药品销售人员交流之后，根据他们日常的进货、销售活动，我认为该应用程序应该包括以下几个方面的内容：

- 药品的基本信息
- 生产厂家的基本信息
- 医院的基本信息
- 进货记录
- 销售记录

药品的基本信息包括：药品的名称（通用名和商品名），批准文号，规格，生产厂家，价格等。

生产厂家的基本信息包括企业的名称，企业的批准文号，企业联系人员的姓名和联系方式，向企业付款的银行帐号信息等。

医院的基本信息包括医院名称，等级，地址，联系人员的姓名和联系方式等。

进货记录包括日期，进货的药品名称、数量、价格等。由于药品不同于其它商品，对有效期有很高的要求，进货的信息应该包括该批次药品的有效期。

销售记录包括日期，销售的药品名称、数量、价格，销售对象（医院名称）和该批次药品的有效期等

2. 功能分析

应用程序应该包括以下基本功能：

- 数据的浏览
- 数据的查询
- 数据的插入、修改、删除
- 简单的统计功能
- 信息提示功能

暨南大学本科实验报告专用纸(附页)

数据的浏览应该可以对第 1 节所提到的信息进行浏览。

数据的查询应当可以根据关键字查询相应的记录。由于药品名称较长，商品名种类繁多，最好能够支持模糊查询和条件组合查询。

数据的插入、修改、删除应该对上述的信息进行添加、修改、删除操作。

简单的统计功能应当可以对销售的库存、销售金额等信息进行统计，为销售人员决策提供数据支持。

信息提示功能是为保障药品质量安全，对用户予以习惯的提示。由于药品销售行业对药品的质量、有效期、年审制度要求很高，在应用程序中应该实现一些同时和警告功能。例如：对库存的过期药品进行警告；在进货时，如有年审过期的药品或药品生产企业年审过期，则不予进库；在销售表中增加药品质量属性。

3. 其它要求

为防止用户误操作，在应用程序增加备份数据库和日志记录的功能。

此外，药品销售从业人员普遍为非计算机专业，所以应用程序应该易于安装和操作，并配有帮助手册以辅助用户。为防止用户操作失误，应用程序还要有适当的提示功能和容错能力。

暨南大学本科实验报告专用纸(附页)

三、 概要设计

在第二章中，已经确定了药品销售管理系统的基本功能和所包含的数据。在本章，将对应用程序中的数据库进行设计。

1. 数据字典¹

对应用程序出现的一些专业术语进行定义：

药品名称：即通用名，是由世界卫生组织编定的在全球范围通用和在药品标准中列入的名称。

商品名：系指经国家药品监督管理局部门批准的特定企业使用的商品名称。

批准文号：生产新药或者已有国家标准的药品的，须经国务院药品监督管理部门批准，并在批准文件上规定该药品的专有编号，此编号称为药品批准文号。批准文号=[国药准字 | 国药试字]+[H|S|J|T|F|B|Z]+8{数字}8

生产批号：用于识别“批”的一组数字或字母加数字，用以追溯和审查批药品的生产历史。

规格：指单位剂量药品中含有药物的量。

包装：指单位包装箱内销售单元数。

药品质量类别：药品质量类别=[原研类|专利类|GMP类|GMP单独定价类]

药品有效期：药品有效期是指该药品被批准的使用期限，表示该药品在规定的贮存条件下能够保证质量的期限。

药品年审：在药品实际销售过程中，由国家工商、药监部门对年审合格药品发放的相关证明。有效期一年，具体数值为年审截止日期，格式为年月日。

企业年审：由国家工商、药监部门对年审合格的药品生产企业发放的相关证明。有效期一年，具体数值为年审截止日期，格式为年月日。

医院等级：依据医院功能、设施、技术力量等对医院资质评定的指标，共分三级十等。

2. E-R 图

根据需求分析中对应用程序所含数据的分析、数据字典的定义和数据之间的关系，可以绘制出 E-R 图（Entity-Relationship Diagram），参见图 3-1。

¹ 本节内容来源于网络

暨南大学本科实验报告专用纸(附页)

E-R 图包含了应用程序数据库中的实体和实体之间的联系。

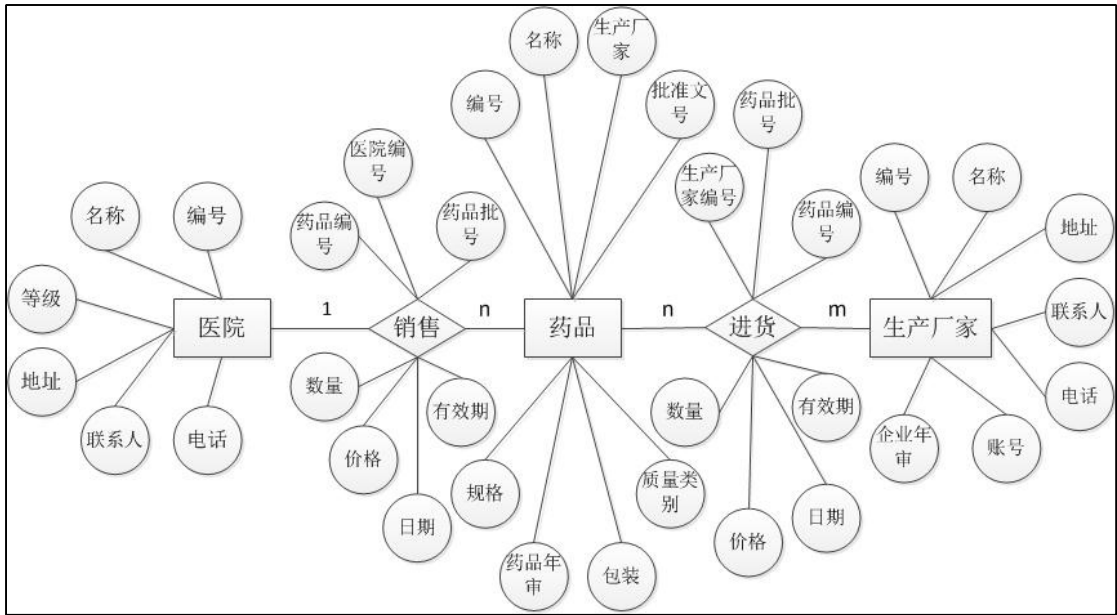


图 3-1 E-R 图

3. 关系数据库设计

根据 E-R 图可以分析出，应用程序中的数据库应该包括药品表、医院表、生产企业表、进货表、销售表五个基本表。每一个基本表中以药品销售活动中必要的信息作为属性。为了减少冗余和便于编程，为每一个表添加一个编号作为码。其具体设计如下：

1) 药品表

药品编号	药品名称	商品名	批准文号	规格	包装	质量类别	企业编号	药品年审
------	------	-----	------	----	----	------	------	------

药品编号=D+3{数字}3

在药品表中，药品编号是码。分析各属性之间的关系可知，没有部分依赖和传递依赖；且只有一个码“药品编号”，所以药品表的关系模式属于 BCNF。

2) 医院表

医院编号	医院名称	等级	地址	联系人	电话
------	------	----	----	-----	----

医院编号=H+3{数字}3

在医院表中，医院编号是码。分析各属性之间的关系可知，没有部分依赖和传递依赖；且只有一个码“医院编号”，所以医院表的关系模式属于 BCNF。

暨南大学本科实验报告专用纸(附页)

3) 生产企业表

企业编号	企业名称	地址	联系人	电话	银行地址	银行账号	企业年审
------	------	----	-----	----	------	------	------

企业编号=F+3{数字}3

在生产企业表中，企业编号是码。分析各属性之间的关系可知，没有部分依赖和传递依赖；且只有一个码“企业编号”，所以生产企业表的关系模式属于 BCNF。

4) 进货表

进货编号	药品编号	生产批号	有效期	数量	单价	金额	日期
------	------	------	-----	----	----	----	----

进货编号=P+数字日期+3{数字}3

数字日期=8{数字}8，（例如：20111123）

进货表的码是“进货编号”。分析表中的各个属性可以发现金额还可以由单价和数量决定，造成了数据冗余。但是为方便后期程序的实现，还是保留了这一个属性。“药品编号”是进货关系的外码。由于存在传递依赖，进货表的关系模式属于 2NF。

5) 销售表

销售编号	药品编号	生产批号	质量	有效期	医院编号	数量	单价	金额	日期
------	------	------	----	-----	------	----	----	----	----

销售编号=S+日期+3{数字}3

“销售编号”是销售表的码。“药品编号”和“医院编号”是销售关系的外码。销售表和进货表类似，销售表也存在传递依赖。所以关系模式属于 2NF。

四、 详细设计

在完成了设计数据库的设计后，则进入了详细设计阶段。在本章，将设计出应用程序的用户界面、程序的基本模块和模块之间的调用关系。

1. 用户界面设计

根据对需求的分析，我认为界面应该有如下特点：

- 直观简洁；
- 方便操作。

1) 主界面

基于以上考虑，在设计用户界面时，设想把基本表的显示部分和控制按钮、查询部分放在一个界面。每一个基本表有一个对应的界面。构想参见图。在 Java Swing 中提供了一种 JTabbedPane 的界面模式，JTabbedPane 可以添加多个 JPanel，能够做出类似于 Office2007 Ribbon 界面的切换卡的界面。采用切换卡界面有如下优势：

- 所有功能有组织地集中存放，不再需要查找级联菜单、工具栏等等；
- 更好地在每个应用程序中组织命令；
- 提供足够显示更多命令的空间；
- 丰富的命令布局可以帮助用户更容易地找到重要的，常用的功能。



图 4-1 主界面设计图

2) 查询结果显示

设置查询条件后，点击按钮可以显示查询结果。查询结果显示在对话框

中的表格里，表格的格式和基本表相同。

3) 新建记录

点击新建记录的按钮，可以弹出对话框进行数据的录入。点击对话框中的按钮就可以完成记录的创建。修改记录的界面与之类似。原型界面参见图。



图 4-2 数据插入对话框设计图

2. 程序调用关系

1) UML 用例图

用例图包含了系统、行为者、用例和用例之间的关系。

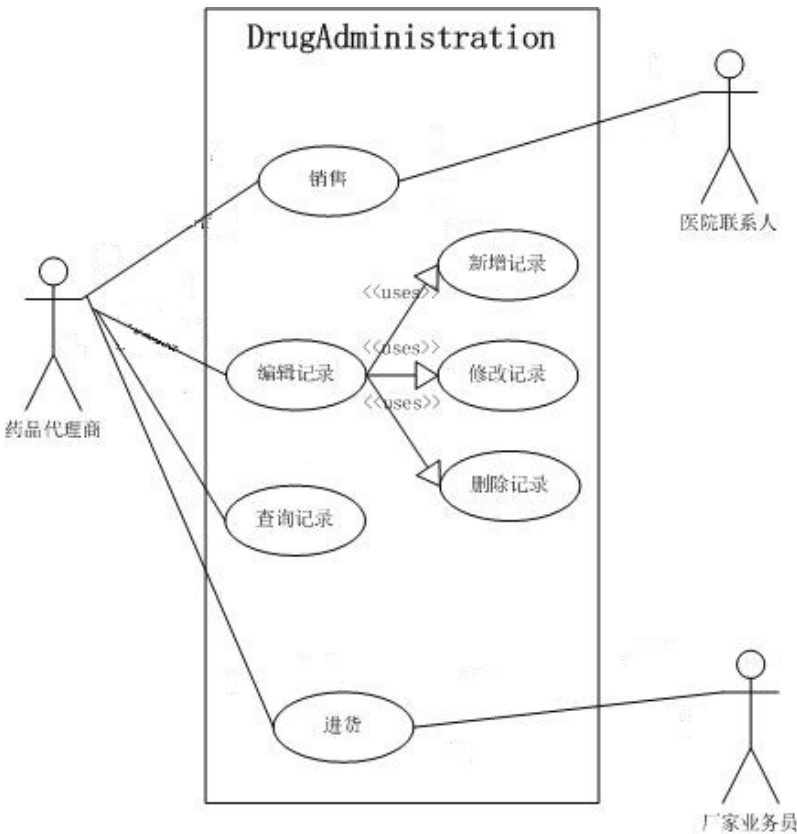


图 4-3 DrugAdministration 用例图

暨南大学本科实验报告专用纸(附页)

如图 4-3，在药品销售管理中，行为者有药品代理商、厂商业务员、医院联系人。药品代理商可以编辑、查询应用程序中的数据记录，编辑记录功能是使用新增记录、修改记录、删除记录三个用例。药品代理商分别通过销售用例、进货用例和医院联系人、厂家业务员联系起来。

2) 数据流图

数据流图 (Data Flow Diagram) 是一种图形化技术，它描绘信息流和数据从输入到输出的过程中所经受的变换。根据数据流图，可以分析出程序所需的模块和模块之间的调用关系。

如图 4-4，在药品销售管理中，在用户界面捕捉用户的操作，接受事务后数据流流向不同的模块。

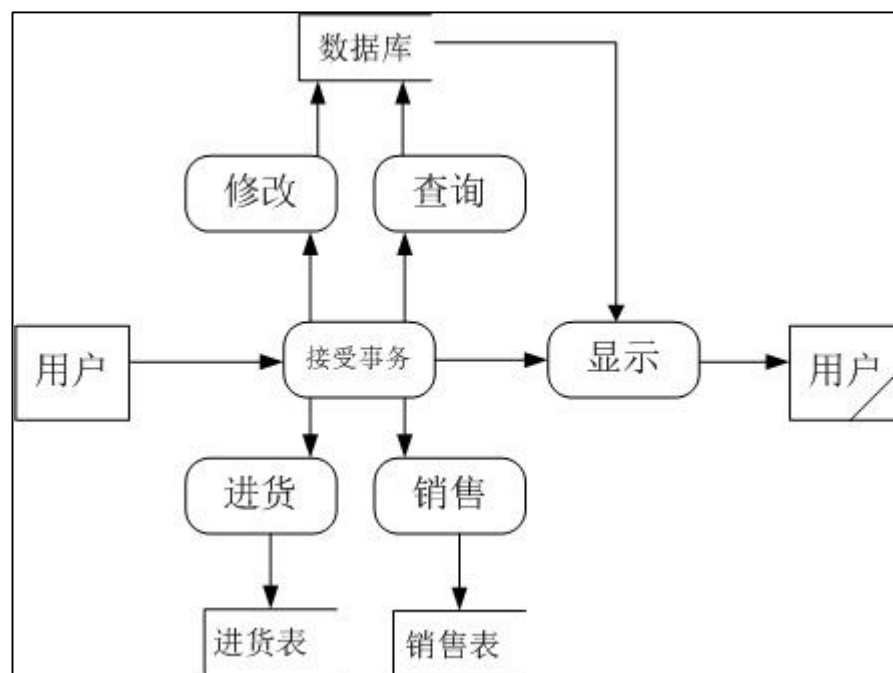


图 4-4 DrugAdministration 数据流图

3) 调用关系图

将数据流图不断细化，可以转化出程序的调用关系图。参见图 4-5。

以药品页面为例，用户点击按钮，应用程序接受事务。根据按钮的不同，调用执行相应程序的模块。如果是修改数据库记录，则要根据用户的选择，调用具体的操作。修改数据记录最终要调用读、写数据库的底层模块。如果点击了关于报表的按钮，则要先读取数据库中的记录，再调用生成报表的模块。在接收了关于统计数据的事务时，需要先读取数据库的记录，在执行相关计算的模块，得出结果。

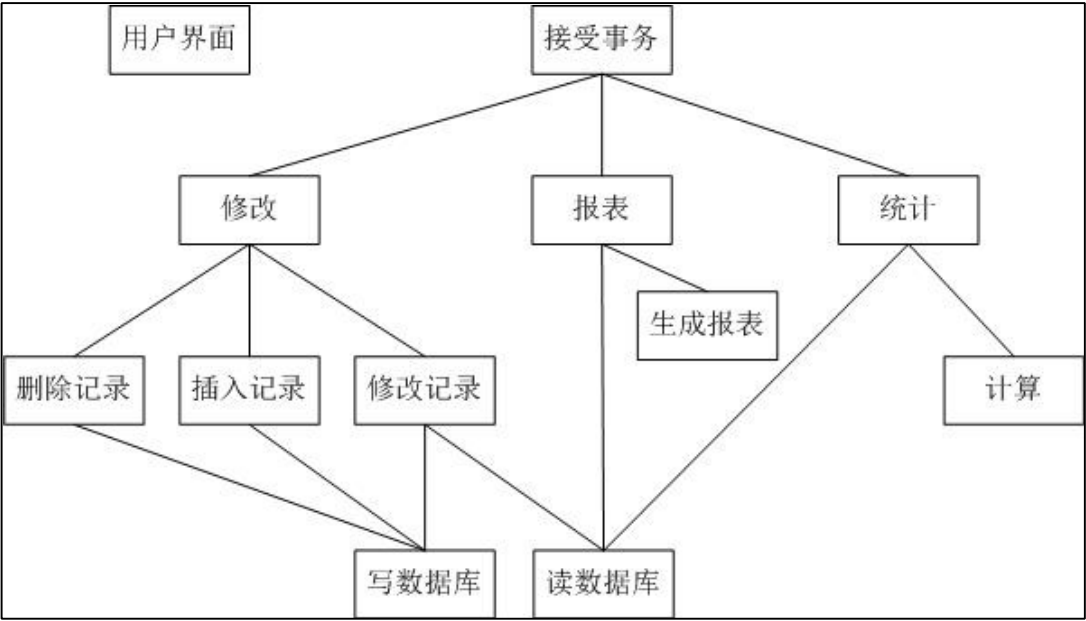


图 4-5 DrugAdministration 程序调用图

五、 编码及实现

在本章，将记录开发应用程序中编码和调试的过程。为清晰的展示源程序的结构，本章将结合程序的 UML 类图对部分代码进行介绍。

1. 工具选择

- 编写语言：JAVA 1.6
- 数据库：Microsoft Office Access
- 报表程序：JasperReport + iReport²

2. UML 类图

1) 实现查询功能的类

共有 7 个相关的类来实现应用程序中的查询功能，UML 类图参见图。BaseSQL 是基类，有受保护的静态变量 url（用来连接 ODBC）和 7 个受保护的方法（用来实现插入、删除、更新、计算记录个数等常用功能）。DrugSQL（药品查询）、HospitalSQL（医院查询）、SupplerSQL（供应商查询）、SaleSQL（销售查询）、PurchaseSQL（进货查询）和 Statistic（统计查询）六个类继承自 BaseSQL，重写了 BaseSQL 里的方法，并根据需要的功能实现了相应的方法。

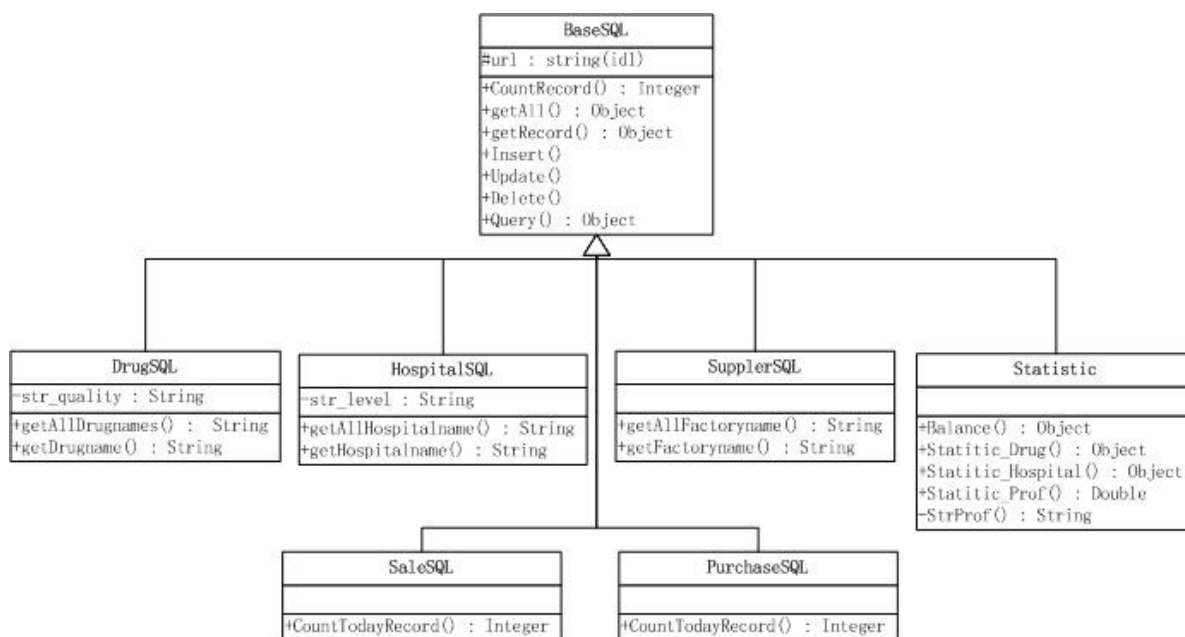


图 5-1 关于查询功能部分类的 UML 类图

² JasperReport 和 iReport 是 JasperForge 开发的 Java 报表程序。

2) 实现报表功能的类

这里实现报表功能主要指连接数据源。本应用程序使用了第三方开源报表工具包 JasperReport 及其设计软件 iReport。在 iReport 中定义报表格式 xml 文件，在源程序中根据 xml 文件，实现 JasperReport 包中相应的方法即可以生成报表。

与实现 SQL 的类相似，共有 7 个相关的类来实现应用程序中的连接报表数据源的功能，UML 类图参见图。BaseReportDataSorce 类继承自 JDataSource 类。JDataSource 是 JasperReport 中提供的类，用以实现连接数据库的功能。BaseReportDataSorce 类有若干个受保护的静态变量和方法。DrugReportDataSorce（连接药品表）、HospitalReportDataSorce（连接医院表）、SupplierReportDataSorce（连接供应商表）、SaleReportDataSorce（连接销售表）、PurchaseReportDataSorce（连接进货表）和 Statistic（连接统计结果）六个类继承自 BaseReportDataSorce，重写了 BaseReportDataSorce 里的方法，并根据需要的功能实现了相应的方法。

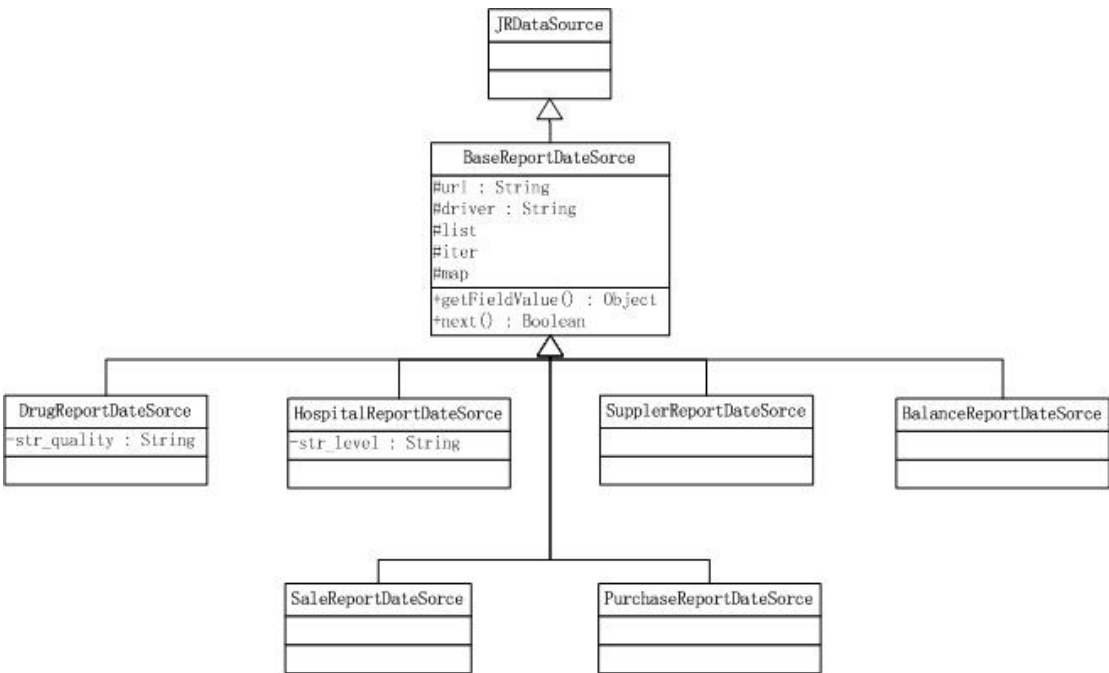


图 5-2 关于报表功能部分类的 UML 类图

3) 实现主界面的类

为实现在设计阶段构想的用户界面，也出于代码重用的目的，设计了以下结构，UML 类图参见图。

MyPanel 类继承自 javax.swing.JPanel 类，有多个受保护的 JButton、

JFrame、JPanel 变量，这些变量是实现每一个子界面所必要的变量，如确认、取消按钮等。MyPanel 类还实现了 java.awt.ActionListener 接口，用来响应按钮事件。DrugPanel（药品子界面）、HospitalPanel（医院子界面）、SupplierPanel（供应商表子界面）、SalePanel（销售表子界面）和 PurchasePanel（进货表子界面）五个类继承自 MyPanel，重写了 MyPanel 里的方法。由于统计界面和帮助界面在外观上不同，并未继承 MyPanel 类，并根据需要的功能实现了相应的方法。

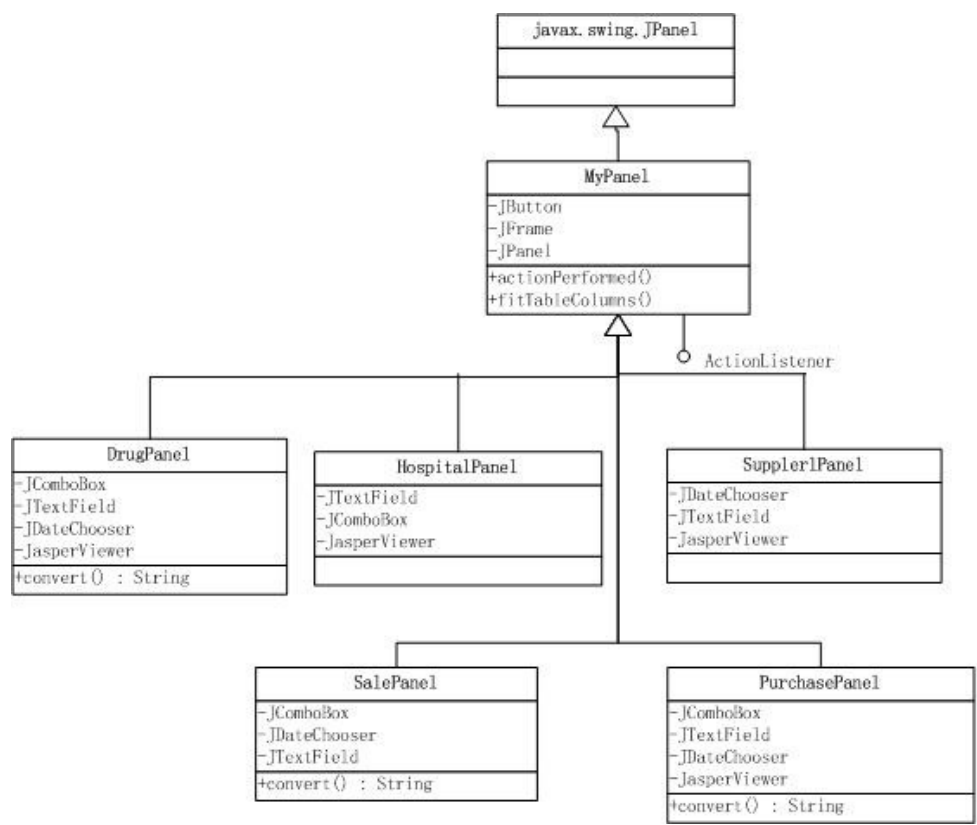


图 5-3 关于主界面部分类的 UML 类图

4) 实现新建记录对话框的类

应用程序要求通过对话框来输入和显示数据，需要重新设计一个 MyDialog。UML 类图参见图。

MyDialog 类继承自 javax.swing.JDialog 类，有多个受保护的 JButton、JPanel 等变量，这些变量是实现每一个子界面所必要的变量，如确认、取消按钮等。MyDialog 类包含了必要的方法，如 Dialog 中读取数据、将数据库中的数据写入 Dialog 里等。MyDialog 类还实现了 java.awt.ActionListener 接口，用来响应按钮事件。NewDrugDialog（药品

子界面)、NewHospitalDialog (医院子界面)、NewSupplierDialog (供应商表子界面)、NewSaleDialog (销售表子界面) 和 NewPurchaseDialog (进货表子界面) 五个类继承自 MyDialog，重写了 MyDialog 里的方法，并根据需要的功能实现了相应的方法。

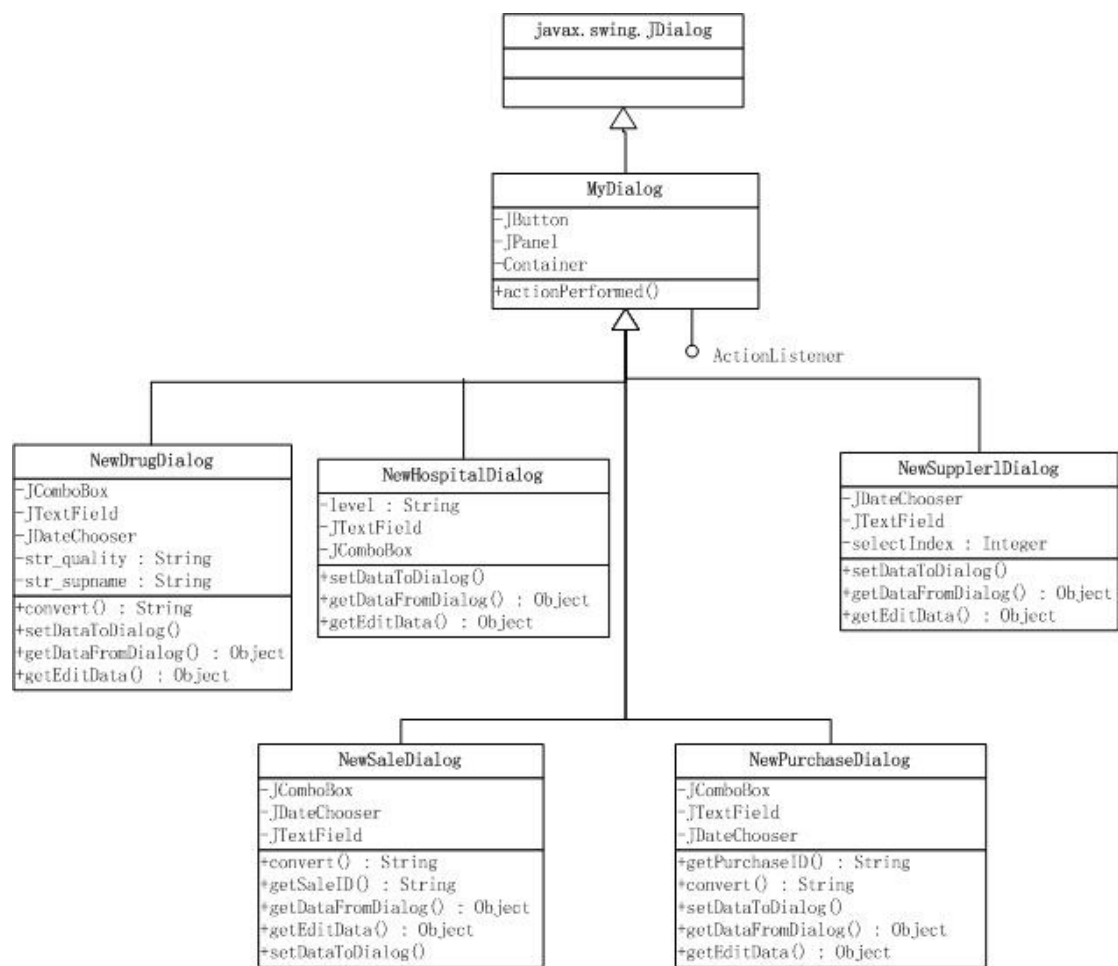


图 5-4 关于数据插入对话框部分类的 UML 类图

5) 自定义 TableModel 类

本应用程序中实验 Java Swing 里提供的 JTable 控件来显示表格。实现 JTable 需要指定 TableModel，用来定义表格的数据模型。由于 Java Swing 中提供的 TableModel 的结构和功能不能满足本应用程序的功能，需要自行定义 MyTableModel 类。UML 类图参见

MyTableModel 类继承自 javax.swing.table.AbstractTableModel。MyTableModel 中定义了两个变量：String 数组 tablehead——用来存放表头和 ArrayList<Object[]> rows——二维表格。MyTableModel 实现了多个方法，包括添加、删除、获取表头数组等。DrugTableModel (药品表)、

HospitalTableModel（医院表）、SupplierTableModel（供应商表）、SaleTableModel（销售表）、PurchaseTableModel（进货表）和BalanceTableModel（库存表）六个类继承自 BaseTableModel，在实现过程中只需变更相应的构造方法即可。

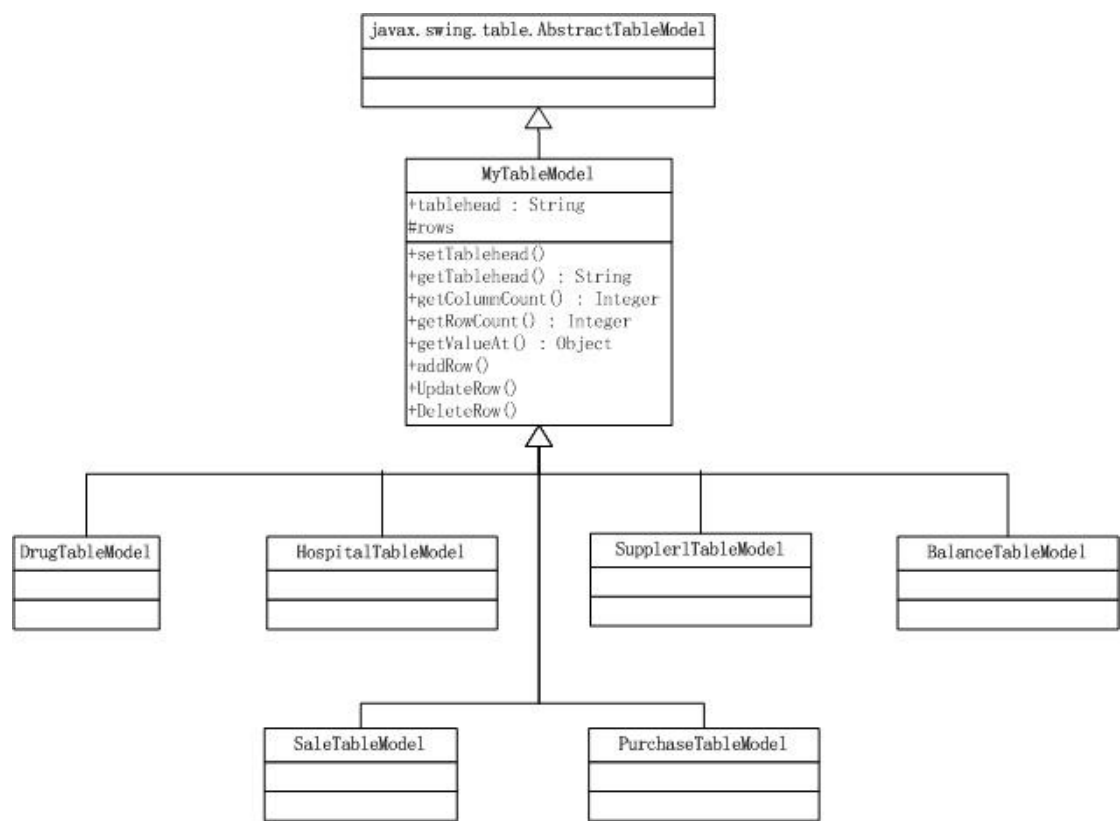


图 5-5 关于 TableModel 部分类的 UML 类图

6) 类之间的组织

在之前的内容，已经说明了各个功能的实现。又是如何将这些功能块组织到一起，具体实现一个模块呢？在此，以实现药品模块为例，说明如何将用户界面、查询、查询结果显示等类组织在一起。

参见图 5-6，DrugAdministration 程序主体由 MyWindow 类组成。MyWindow 通过 JTabbedPane 控件将多个 Panel 组合在一起形成主界面，其中包括 DrugPanel 类的对象 drugPanel。drugPanel 调用 NewDrugDialog 类的对象 dialog 来新建、修改数据记录；drugPanel 调用 DrugTableModel 类的对象 tablemodel 来定义表格；drugPanel 调用 DrugReport 类的对象 viewer 来显示报表；drugPanel 调用 DrugSQL 类的静态方法实现数据查询。

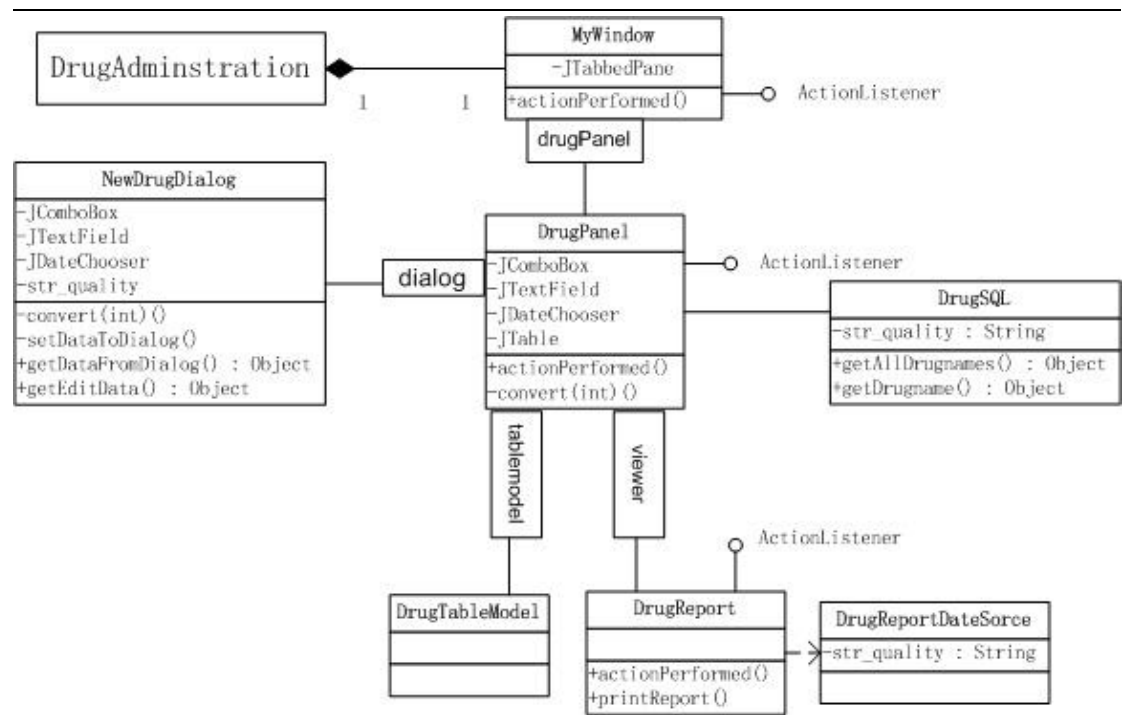


图 5-6 关于 Drug 部分类的 UML 类图

3. 重要 SQL 查询语句（部分）

由于应用程序中不同表的插入、删除、修改的 SQL 语句类似，仅是属性有所不同。所以这里只列出关于药品表的 SQL 语句。

查询药品数量

```
SELECT COUNT(*) FROM Drug
```

查询所有药品的名称，并按升序返回

```
SELECT drug_name FROM Drug ORDER BY drug_id ASC
```

根据药品编号查询药品的名称

```
SELECT drug_name FROM Drug WHERE drug_id = ?
```

查询所有药品的记录

```
SELECT * FROM Drug ORDER BY drug_id ASC
```

根据药品编号查询某一药品记录

```
SELECT * FROM Drug WHERE drug_id = ?
```

根据药品编号删除某一药品记录

```
DELETE FROM Drug WHERE drug_id = ?
```

查询不同药品的销售数量

暨南大学本科实验报告专用纸(附页)

```
SELECT drug_id, SUM(amount) FROM sale GROUP BY drug_id
```

查询不同医院的采购数量

```
SELECT hos_id, SUM(amount) FROM sale GROUP BY hos_id;
```

根据药品、生产批号和有效期将进货进行分组

```
SELECT drug_id, batch_num, vali_date, SUM(amount)
```

```
FROM purchase
```

```
GROUP BY drug_id, batch_num, vali_date;
```

根据药品、生产批号和有效期将销售进行分组

```
SELECT drug_id, batch_num, vali_date, SUM(amount)
```

```
FROM sale
```

```
GROUP BY drug_id, batch_num, vali_date;
```

在求库存的药品时，将会调用这两个 SQL 语句的结果。

4. 重要代码（部分）

在程序中多次使用过convert()方法。该方法可以把JComboBox控件返回的选中项下标转化为相应记录在供应商表中的码。例如输入数字1可以返回字符串格式的码“F001”。

```
private String convert(int num) {  
    char[] c = { 'F', '0', '0', '0' };  
    c[3] += num % 10;  
    num /= 10;  
    c[2] += num % 10;  
    c[1] += num / 10;  
    String s = new String(c);  
    return s;  
}
```

生成查询药品的 SQL 语句。因为查询需要支持多条件复合查询，必须能够根据用户输入的条件生成相应的 SQL 语句。Object 数组 con 存有在查询区内获取的条件，如果条件不为空，则在 SQL 语句中加入这个条件。这样，就可以获得多个条件的 SQL 语句。

```
StringBuffer sb = new StringBuffer("SELECT * FROM Drug WHERE ");  
boolean b = false;  
if (con[0] != "") {
```

暨南大学本科实验报告专用纸(附页)

```
sb.append("drug_name = '" + con[0] + "'");
b = true;
}
if (con[1] != "") {
    if (b)
        sb.append(" AND ");
    sb.append("factory = '" + con[1] + "'");
    b = true;
}
if (con[2] != null) {
    if (b)
        sb.append(" AND ");
    sb.append("quality = " + con[2]);
    b = true;
}
if (con[3] != null) {
    if (b)
        sb.append(" AND ");
    sb.append("verification < #" + con[3] + "#");
}
String sql_query = new String(sb);
```

获取库存药品数量。`purchase_amount` 和 `sale_amount` 是均为视图，分别根据药品、生产批号和有效期对进货和销售进行分组。这两个视图就用到了第 3 节提到的两个 SQL 语句。视图中有四个属性：药品、生产批号、有效期和数量。获取库存药品数量的方法如下：

```
public static Object[][] Balance() {
    Object[][] Pobj = null, Sobj = null;
    String sql_query1 = "SELECT * FROM purchase_amount";
    String sql_query2 = "SELECT * FROM sale_amount";
    try {
        Connection conn = DriverManager.getConnection(url, "",
    "");

        Statement sql = conn.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
        ResultSet rs;
        rs = sql.executeQuery(sql_query1);
        rs.last(); // Move to the last row
        int n1 = rs.getRow(); // Last row's number
        rs.beforeFirst(); // Move back to the first row
```

暨南大学本科实验报告专用纸(附页)

```
Pobj = new Object[n1][4];
int i = 0;
while (rs.next()) {
    Pobj[i][0] = DrugSQL.getDrugname(rs.getString(1));
    Pobj[i][1] = rs.getString(2);
    Pobj[i][2] = rs.getDate(3);
    Pobj[i][3] = rs.getInt(4);
    i++;
}
rs = sql.executeQuery(sql_query2);
rs.last(); // Move to the last row
int n2 = rs.getRow(); // Last row's number
rs.beforeFirst(); // Move back to the first row
Sobj = new Object[n2][4];
int j = 0;
while (rs.next()) {
    Sobj[j][0] = DrugSQL.getDrugname(rs.getString(1));
    Sobj[j][1] = rs.getString(2);
    Sobj[j][2] = rs.getDate(3);
    Sobj[j][3] = rs.getInt(4);
    j++;
}
conn.close();
} catch (SQLException e) {
    e.printStackTrace();
}
for (int i = 0; i < Pobj.length; i++)
    for (int j = 0; j < Sobj.length; j++)
        if (Pobj[i][0].equals(Sobj[j][0])
            && Pobj[i][1].equals(Sobj[j][1]))
            Pobj[i][3] =
Integer.parseInt(Pobj[i][3].toString()) -
Integer.parseInt(Sobj[j][3].toString());
    Sobj = null;
    return Pobj;
}
```

这两个视图的结果分别储存在二维 Object 数组 Pobj 和 Sobj 中。如果 Pobj 和 Sobj 里的药品名称和生产批号相同，只可以判断是同一批的同种药品，两者之差就是库存量。

计算利润。本应用程序中可以显示近半年内的利润。一个难点是限制时

暨南大学本科实验报告专用纸(附页)

间的范围是最近的半年。另外一个难点是根据进货和销售表计算利润。

计算最近六个月的代码如下。其中整数数组 `month` 储存的就是最近六个月月份的数值。

```
int[] month = new int[6];
Calendar cal = Calendar.getInstance();
cal.setTime(new Date());
int i, m = cal.get(Calendar.MONTH) + 1;
for (i = 5; i >= 0; i--, m--)
    month[i] = (m + 12) % 12 != 0 ? (m + 12) % 12 : 12;
```

解决了限制时间范围的问题后,计算利润时先计算每个月销售的总收入,再减去当月进货的总支出,即可以得到当月的利润。代码如下:

```
private static String StrProf(int num, boolean isSale) {
    String str1 = "SELECT SUM(Sale.cost) FROM Sale WHERE
sale_date >= ";
    String str2 = " AND sale_date < ";
    String str3 = "SELECT SUM(Purchase.cost) FROM Purchase
WHERE pur_date >= ";
    String str4 = " AND pur_date < ";
    int[] month = new int[6];
    String[] strmonth = new String[7];
    Calendar c = Calendar.getInstance();
    c.setTime(new Date());
    int i, y = c.get(Calendar.YEAR), m = c.get(Calendar.MONTH)
+ 1;
    for (i = 5; i >= 0; i--, m--)
        month[i] = (m + 12) % 12 != 0 ? (m + 12) % 12 : 12;
    for (i = 0; i < 6; i++) {
        StringBuffer s = new StringBuffer("#--01#");
        if (month[i] > c.get(Calendar.MONTH) + 1) {
            s.insert(2, month[i]);
            s.insert(1, y - 1);
        } else {
            s.insert(2, month[i]);
            s.insert(1, y);
        }
        strmonth[i] = new String(s);
    }
    StringBuffer s = new StringBuffer("#--01#");
    s.insert(2, month[5] + 1);
    if (month[5] == 12)
        s.insert(1, (y + 1));
```

暨南大学本科实验报告专用纸(附页)

```
        else
            s.insert(1, y);
        strmonth[6] = new String(s);
        if (isSale)
            return (str1 + strmonth[num] + str2 + strmonth[num + 1]);
        else
            return (str3 + strmonth[num] + str4 + strmonth[num + 1]);
    }

    public static Double[] Statitic_Prof() {
        Double[] prof = new Double[6];
        String sq;
        try {
            Connection conn = DriverManager.getConnection(url, "",
""");
            Statement sql = conn.createStatement();
            ResultSet rs;
            for (int i = 0; i < 6; i++) {
                sq = StrProf(i, true);
                rs = sql.executeQuery(sq);
                while (rs.next()) {
                    prof[i] = rs.getDouble(1);
                }
                sq = StrProf(i, false);
                rs = sql.executeQuery(sq);
                while (rs.next()) {
                    prof[i] -= rs.getDouble(1);
                }
            }
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return prof;
    }
}
```

为了实现进货和销售时自动编号，即根据当日进货、销售数量自动编号。比如，2012 年 1 月 1 日还没有销售记录，此时打开新建记录对话框，可以自动把销售编号编为 S20120101001。于是使用 `getSaleID` 方法来实现以上功能。

```
private String getSaleID(int num) {
```

暨南大学本科实验报告专用纸(附页)

```
String str = null;
char[] c = { '0', '0', '0' };
num++;
c[2] += num % 10;
num /= 10;
c[1] += num % 10;
c[0] += num / 10;
String s = new String(c);
SimpleDateFormat ft = new SimpleDateFormat("yyyyMMdd");
str = "S" + ft.format(new Date()) + s;
return str;
}
```

打开帮助文件。打开帮助文件使用的是 Java 提供的方法。

`Runtime.getRuntime().exec()`可以执行系统指令。

```
try {
    String[] cmd = new String[4];
    cmd[0] = "cmd.exe";
    cmd[1] = "/C";
    cmd[2] = "start";
    cmd[3] = ".\\Manual.pdf";
    Runtime.getRuntime().exec(cmd);
} catch (IOException e1) {
    JOptionPane.showMessageDialog(frame, "帮助文件打开失败!", "警告", JOptionPane.WARNING_MESSAGE);
}
```

5. 调试过程

在程序编写初期，每一个窗口都是独立的一个类，不仅要写大量重复的代码，还增加了出错的机会。后来，该有继承基类的方法，将每一个类重复出现的部分代码在基类中实现，使代码更加简洁，逻辑关系更加清晰。

在编码过程中，需要计算 SQL 语句执行结果的记录数以便申请合适大小的数组。但是 Java 语言不能够直接得到 `ResultSet` 的大小。在网上搜索到解决方法。通过移动游标，可以间接求得记录的数目，方法十分巧妙。代码如下：

先将 `sql` 设置为可以回滚的类型：

```
Statement sql = conn.createStatement(
```

暨南大学本科实验报告专用纸(附页)

```
ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
```

再有以下语句：

```
rs = sql.executeQuery(sql_query);  
rs.last(); // Move to the last row  
int n = rs.getRow(); // Last row;s number  
rs.beforeFirst(); // Move back to the first row
```

在 Access 数据库上操作有关日期的数据时，遇到一个问题是 SQL 语句错误。后来得知在 Access 上日期是用 ‘#’ 表示，而不是其它数据库常用的单引号。所以 SQL 语句要写成 `SELECT * FROM Drug WHERE verification < # ? #` 而不是 `SELECT * FROM Drug WHERE verification < ' ? '`。此外，在 Java 提供的数据类型中，日期格式有两个类，分别是 `java.util.Date` 和 `java.sql.Date`，两者并不通用。比如一般日期操作用的是 `java.util.Date`，在数据库操作时用的是 `java.sql.Date`，所以在编程时要留心日期格式的转换，避免错误。

此外，在编程过程中还出现了许多小的错误，比如数组越界、数据格式转化错误……，在此不一一赘述。

通过完成这次数据库课程设计，不仅对数据库操作的相关知识有了系统的学习，对自己的编程能力、程序设计能力的提高都有很大的促进作用。

六、 作品概览

1. 功能展示

1) 数据的浏览

程序主界面参见图 6-1，可以看到主界面由多个切换卡构成。每一个页面下有表格区、控制按钮、查询区。表格区可以显示基本表，以供用户对数据进行浏览。控制按钮由多个按钮。点击“打开表格”即可显示基本表。点击“新建记录”按钮可以在弹出的对话框中输入数据创建记录。点击“修改记录”按钮或“删除记录”按钮，可以进行修改或删除。点击“导出报表”按钮可以导出报表。在查询区，则可以输入条件进行查询。



图 6-1 数据浏览

2) 数据的查询

如图 6-2。在查询区中在查询区，有多个查询条件，可以选择填入一个或多个条件。点击“查询”，可以在基本表中查询对应的记录；点击“清空”可以清空输入的条件。查询结果将在对话框中显示。

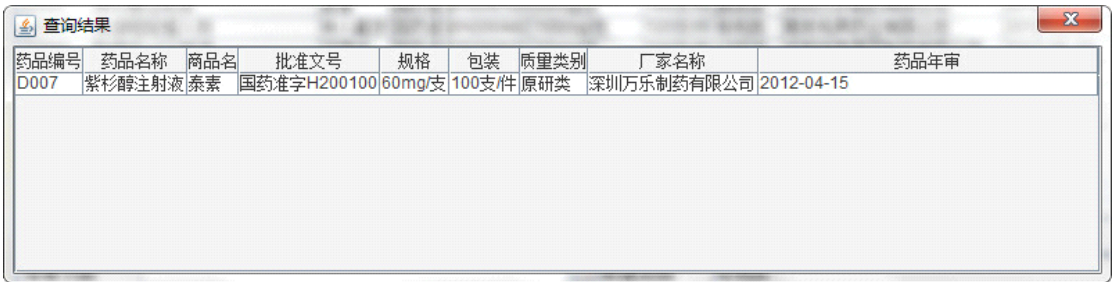


图 6-2 查询结果显示

暨南大学本科实验报告专用纸(附页)

3) 数据的插入、修改、删除

在每一个页面下点击“新建记录”，可以弹出对话框，如图 6-3，可以输入具体的数据。点击对话框内的“确认”按钮，就可以完成记录的插入。

新建药品

药品信息

药品编号

药品名称

商品名称

生产批号

规格

包装

质量类别

生产厂家

企业年审

确认

取消

图 6-3 新建药品记录

在基本表中选择一条数据，再点击“修改记录”，可以弹出对话框，对话框和图 6-3 所示对话框相同，只是已经填有数据库中的数据，可以对数据进行修改。点击对话框内的“确认”按钮，就完成了记录的修改。

在基本表中选择一条数据，再点击“删除记录”，可以弹出一个警告对话框，提醒用户是否确认删除。点击“确认”后，即可以记录的删除。

4) 简单的统计功能

点击“打开表格”，可以显示库存表，包括了药品名称、生产批号、有效期和库存数量在内的信息。点击“导出报表”，可以导出库存报表。



图 6-4 统计页面

点击统计图表中的选项，可以显示对应的统计图。统计图包括：

- 1. 月度利润统计图（折线图）——显示近半年来每月的利润；
- 2. 药品销量统计图（饼图）——显示各种药品销量的比重；

暨南大学本科实验报告专用纸(附页)

3. 医院销量统计图（饼图）——显示各家医院采购量的比重。

5) 报表导出功能

应用程序中可以导出五个基本表和库存报表。报表实现了自动分页的功能。报表如图 6-5。



图 6-5 生成报表

6) 备份和备份的导入

在帮助页面，可以点击“备份”按钮，可以选择一条路径备份 Access 数据库文件。备份就是把数据库文件复制到用户选择的路径下。参见图 6-6。



图 6-6 备份和帮助用户

暨南大学本科实验报告专用纸(附页)

点击“导入备份”，则可以将备份文件导入到应用程序中。

此外，在帮助界面中还可以点击“打开帮助文件”来打开用户手册。

7) 日志记录功能

应用程序中包含一个 Record.log 文件，用来记录应用程序的数据操作日志。记录的内容包括时间、执行操作的 SQL 语句（仅记录添加、修改和删除操作）和操作的结果（成功或失败）。如某一次操作后记录的内容如下：

```
Mon Nov 28 15:55:10 CST 2011
INSERT INTO Drug VALUES
('D999','ABCD','LSD','S0123431','1k/g','1m/l',2,'F007','#2011-11-30#)
Execute Succeed

Mon Nov 28 15:56:24 CST 2011
UPDATE Drug SET drug_name ='ABCD',con_name=LSD',app_num='
S0123431',spec='1k/g ',pack='1m/l',quality=1,factory='F008',verification=#2012-11-30#
WHERE drug_id = 'D999'
Execute Succeed

Mon Nov 28 15:56:27 CST 2011
DELETE FROM Drug WHERE drug_id = 'D999'
Execute Succeed
```

2. 不足之处

由于时间和精力有限，也由于编程能力还有待提高等原因，药品销售管理系统应用程序中还有一些不足。

一个不足是程序中数据结构的封装做的不足。程序中诸如药品、医院等数据可以封装成类，这样更加符合面向对象程序设计的要求，也可以使程序的逻辑结构更加清晰，但出于编程简单，并没有构造这些类。在日后的修改中，可以进一步将这些数据结构封装成类，

另外一个不足是一些功能没有实现。比如用户需要药品进货入库时对年审过期的药品应当予以提示。但在程序中没能够实现此功能。

以上不足之处应当在日后的维护、升级过程中予以改进。