

Java程序设计



第11章 网络、多媒体和数据库编程



第11章 网络、多媒体和数据库编程

- 11.1 网络编程
- 11.2 多媒体编程
- 11.3 数据库编程

11.1 网络编程





网络编程





网络信息获取

- 使用 java.net.URL 进行网络信息获取

- URL : 地址 (uniform resource locator)

- 读取网页文件内容的步骤如下

- (1) 创建一个URL类型的对象。
 - `URL url = new URL("http://www.dstang.com");`
 - (2) 利用URL类的`openStream()` , 获得对应的InputStream类的对象。
 - `InputStream stream = url.openStream();`
 - (3) 通过 `InputStream` 或 `InputStreamReader` 来读取内容。

- 示例 : `URLGetFile.java`



更复杂的网络信息获取

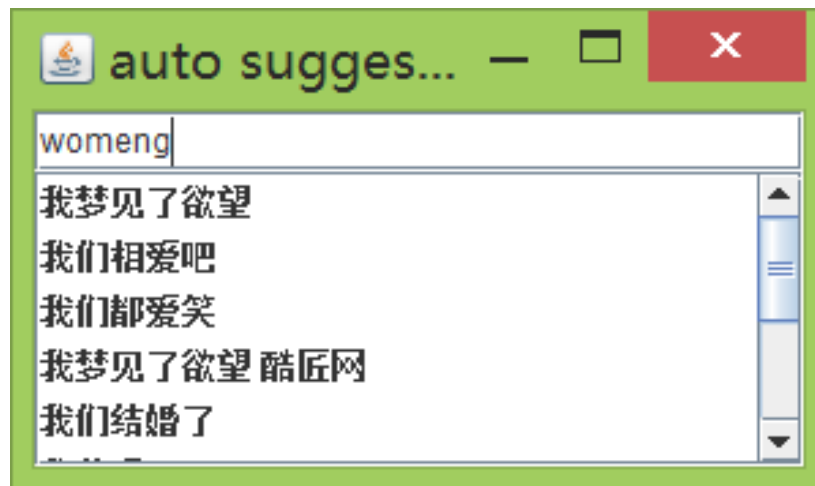
- 可以使用第三方库，参见lib子目录
- 如Apache的httpclient
 - <http://hc.apache.org/>
- 在Eclipse中引入库
 - 项目右击—Build Path—Add External Archives...
 - 类似地NetBeans中，项目—库—点右键—添加jar文件
- 示例 HttpClientStock.java
- `str = Request.Get("http://www.dstang.com")`
- `.execute().returnContent().asString();`
- 稍早的一篇文章 <http://blog.chinaunix.net/uid-7591044-id-1742965.html>



- 获取股票信息
- HttpClientStock.java



- 获取建议词 (suggestion)
- HttpSuggestion.java
- 要点
 - 网络信息获取
 - 字符串处理及正则表达式
 - 界面、布局、事件
 - 线程
- 小技巧：在浏览器中按**F12**可以查看网络通信的过程





实现底层网络通信

- Socket类

- ▣ 客户端要与服务端相连，则客户端需要建立Socket对象。

- Socket s = new Socket("机器名或Ip地址", 端口号)
 - 然后使用流来进行通信

- ServerSocket类

- ▣ 需要调用accept方法接受客户呼叫

- 得到的又是一个Socket, 它可以与客户进行通信

- 示例：TestClient.java TestServer.java

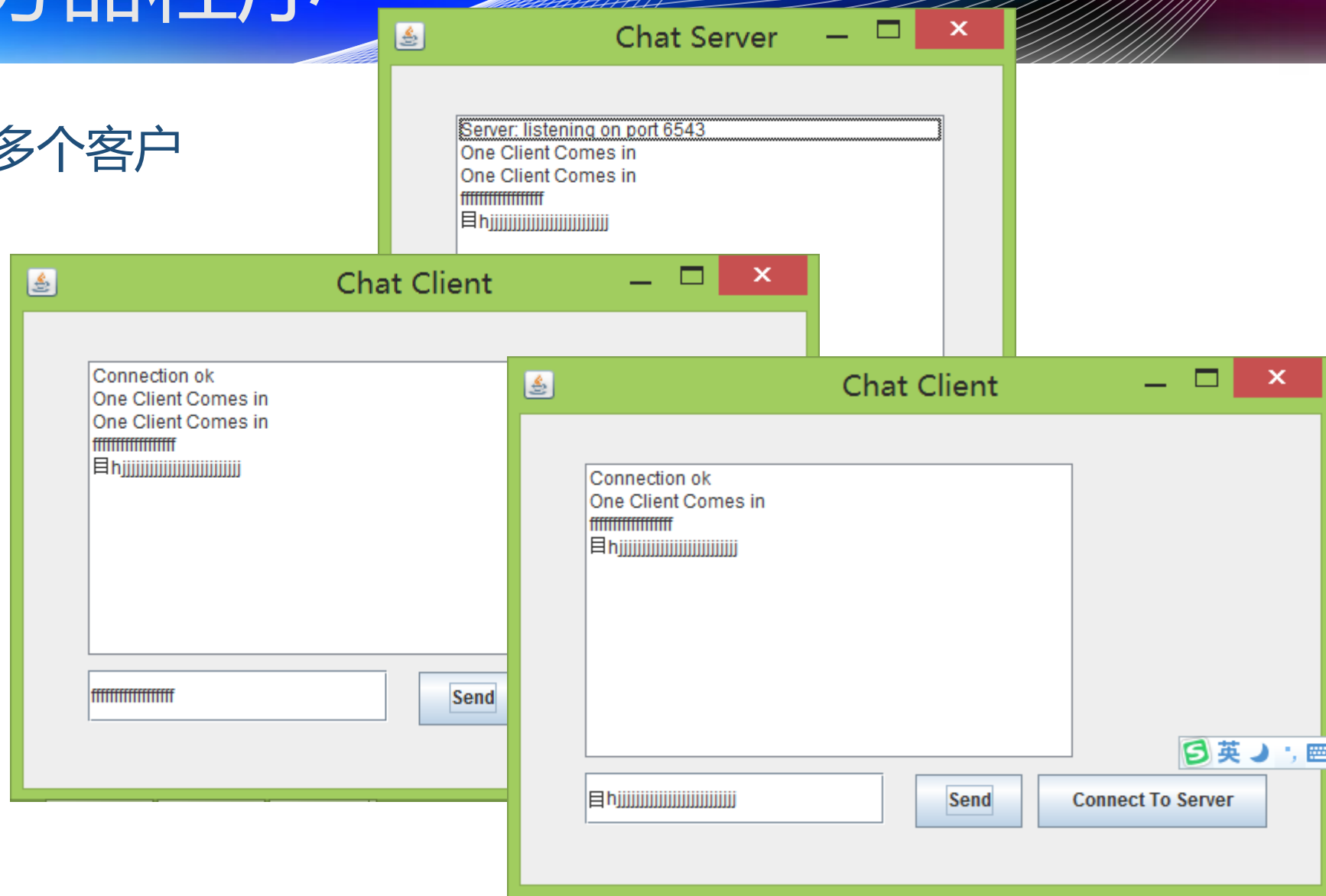
实现多线程服务器程序



□ 经常需用线程来处理多个客户

□ 示例：chat

- chat子目录下的
 - [ChatServer.java](#)
 - [ChatClient.java](#)





用Socket来实现Server及Client

- 见SocketDemo目录
- HTTPServer.java 简单的HTTP服务器
- HTTPClient.java 简单的HTTP客户端



11.2 多媒体编程





多媒体编程





- 绘制图形

- Graphics类 及其子类 Graphics2D

- 它相当于组件的绘图环境，利用它可以进行各种绘图操作。

- 获得Graphics对象常有两种方法

- 一是使用组件的 `getGraphics()`方法来
 - 常用Canvas及JComponent对象来进行绘图。
 - 可以Override Cavans的 `paint(Graphics g)`
 - 或JPanel的 `paintComponent(Graphics g)` 方法
 - 它们会带一个Graphics参数

Graphics的绘图方法



- 画三维矩形 `draw3DRect(int x, int y, int width, int height, boolean raised)`
- 画弧 `drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)`
- 画文字 `drawBytes(byte[] data, int offset, int length, int x, int y)`
- 画文字 `drawChars(char[] data, int offset, int length, int x, int y)`
- 画直线 `drawLine(int x1, int y1, int x2, int y2)`
- 画椭圆 `drawOval(int x, int y, int width, int height)`
- 画多边形 `drawPolygon(int[] xPoints, int[] yPoints, int nPoints)`
- 画多边形 `drawPolygon(Polygon p)`
- 画折线 `drawPolyline(int[] xPoints, int[] yPoints, int nPoints)`
- 画矩形 `drawRect(int x, int y, int width, int height)`
- 画圆角矩形 `drawRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)`
- 画文字 `drawString(AttributedCharacterIterator iterator, int x, int y)`
- 画文字 `drawString(String str, int x, int y)`
- 画填充三维矩形 `fill3DRect(int x, int y, int width, int height, boolean raised)`
- 画填充弧 `fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)`
- 画填充椭圆 `fillOval(int x, int y, int width, int height)`
- 画填充多边形 `fillPolygon(int[] xPoints, int[] yPoints, int nPoints)`
- 画填充多边形 `fillPolygon(Polygon p)`
- 画填充矩形 `fillRect(int x, int y, int width, int height)`
- 画填充圆角矩形 `fillRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)`



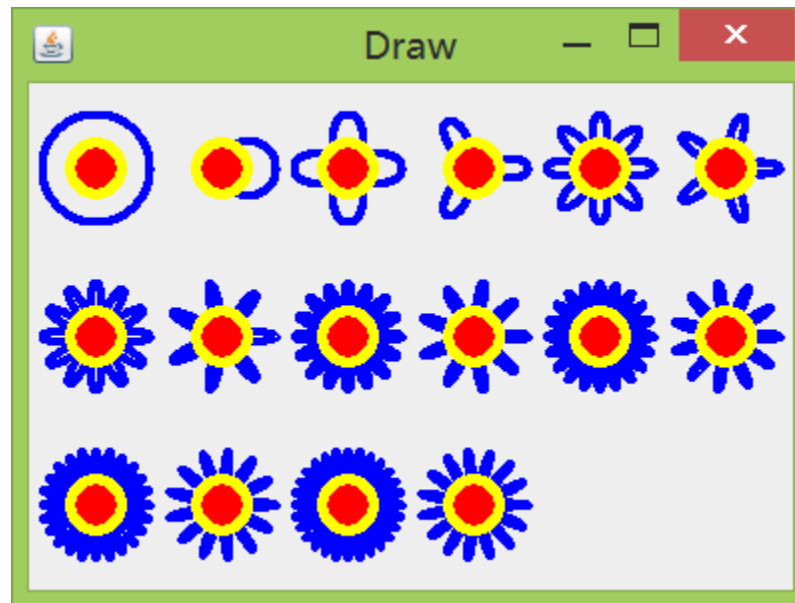
几个辅助类

- Point 表示一个位置点
- Dimension 表示宽和高
- Rectangle 表示一个矩形
- Polygon 表示一个多边形
- Color 表示颜色
- Font 表示字体
 - `myFont = new Font("Times New Roman", Font.BOLD, 12);`

示例：绘图



- 示例：[DrawFlowers.java](#)
- 要点
 - 继承JPanel, 重写 paintComponent()
 - `setDoubleBuffered(true);`



- 利用Graphics类的drawImage()方法显示图像

- 示例：显示图像实现动画

□ DrawImageAnimator.java

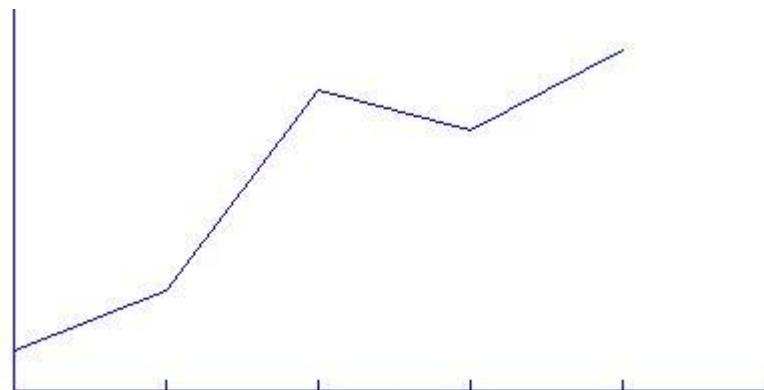
□ 要点：

- getClass().getResource(".");
- ImageIO.read()
- g.drawImage(img, 0, 0, this);
- repaint()





- BufferedImage 类是 Image的子类
- BufferedImage 对象的getGraphics() 可以得到一个绘图对象
- ImageIO类的read及write方法
- 例：JpegCreate.java
 - 创建一个 BufferedImage 对象
 - 使用该对象的getGraphics()得到绘图对象
 - 使用Graphics的drawLine方法来画图
 - 保存为文件，或供其他对象绘制



Java Media API



- <http://www.oracle.com/technetwork/java/javase/tech/media-141984.html>
- 包括
 - Java Advanced Imaging (JAI)
 - Java 3D
 - Java Media Framework (JMF)
 - 下载
<http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-java-client-419417.html>



- JMF : Java Media Framework
- 下载
 - <http://www.oracle.com/technetwork/java/javase/download-142937.html>
 - 安装后，注意path及classpath的变化
 - 如果是用IDE,则要将jar加到Lib中



- 示例

- ▣ jmfDemo目录下

- PlayerMusic.java 播放mp3
 - URL url = file.toURI().toURL();
 - player = Manager.createRealizedPlayer(playFile.toURI().toURL());
 - player.addControllerListener(this);
 - player.prefetch();



可使用第三方库

- 例如：javalayer 的mp3播放库
- <http://www.javazoom.net/javalayer/sources.html>
- `import javazoom.jl.player.Player;`
 - `BufferedInputStream buffer = new BufferedInputStream(
new FileInputStream(filename))`
 - `player = new Player(buffer);`
 - `player.play();`

- 示例

- JMFMediaPlayer.java 播放视频及音频

- jmfDemo目录下



11.3 数据库编程





数据库编程





- 数据库(Database)
- 数据库管理系统(DBMS : relational database management system)
 - MySql
 - Oracle
 - Microsoft SQL Server
 - DB2
 - 等等
- 数据库系统的优点：
 - 数据的共享性、独立性、完整性和冗余数据少
- 管理功能
 - 数据定义/操纵/完整/安全/并发

直观了解数据库



- 数据库中各种表
- 还有其他对象



- 关系型数据库 (relational database)
- 数据表(Table)、记录(Record)、字段(Field)
- 字段类型
 - ▣ char int smallint bit float datetime image
- 主键: primary key
- 示例 :
 - ▣ 书目表: 编号(ID), 书名, 作者, 出版社, 关键字
 - ▣ 读者表: id, 姓名, 性别, 系别, email
 - ▣ 借阅表: 人号, 书号, 日期,
 - ▣ 选课表: 人, 课, 成绩
 - 2501, A03, 80
 - 2501, A04, 83
 - 2502, A04, 87



- 用可视化的界面
 - 如：使用SQL Server企业管理器
- 通过SQL语句
 - 如：使用SQL Server查询分析器



数据库查看工具

- 使用一些第三方工具，如
 - 一个很好用的工具：Navicat（可管理多种数据库）
- Netbeans中查看数据库
 - 窗口--服务，打开服务窗口，其中，数据库—右键—新建连接或注册MySQL服务器
- Eclipse中可以安装一些插件
 - 如 DB Viewer
 - http://www.ne.jp/asahi/zigen/home/plugin/dbviewer/about_en.html
 - 如 Quantum DB
 - <http://quantum.sourceforge.net/>



- 适合于学习的
- JSqlTool （ 推荐下载并阅读 ）
 - ▣ <http://sourceforge.net/projects/jsqltool/>
- SqlAdmin
 - ▣ <http://sourceforge.net/projects/sqladmin/>



- SQL (Structured Query Language) , 即结构化查询语言
- SQL包含数据定义、数据查询、数据操纵和数据控制等多种功能
- 是数据库的标准操作语言



- SELECT * FROM [publishers]
- SELECT age, sex, salary + bonus
 - FROM employee
 - WHERE depart= '销售部' and title= '经理'
 - 有时还可以包含
 - [GROUP BY <列名> [HAVING<条件>]]
 - [ORDER BY <列名> [ASC , DESC]]
- SELECT avg(salary) FROM employee



SQL语句（续）

- 加入数据

- ▣ INSERT INTO [employee] (name, age) VALUES ('李明', 18)

- 更新数据

- ▣ UPDATE [employee] SET salary = salary + 500

- 删除数据

- ▣ DELETE FROM [employee] WHERE age > 80

- 创建及删除数据表

- ▣ CREATE TABLE [employee]

- ▣ (id integer, name char(10), age integer)

- ▣ DROP TABLE [employee]



- Java访问数据库的基本技术
- JDBC (Java Database Connectivity)

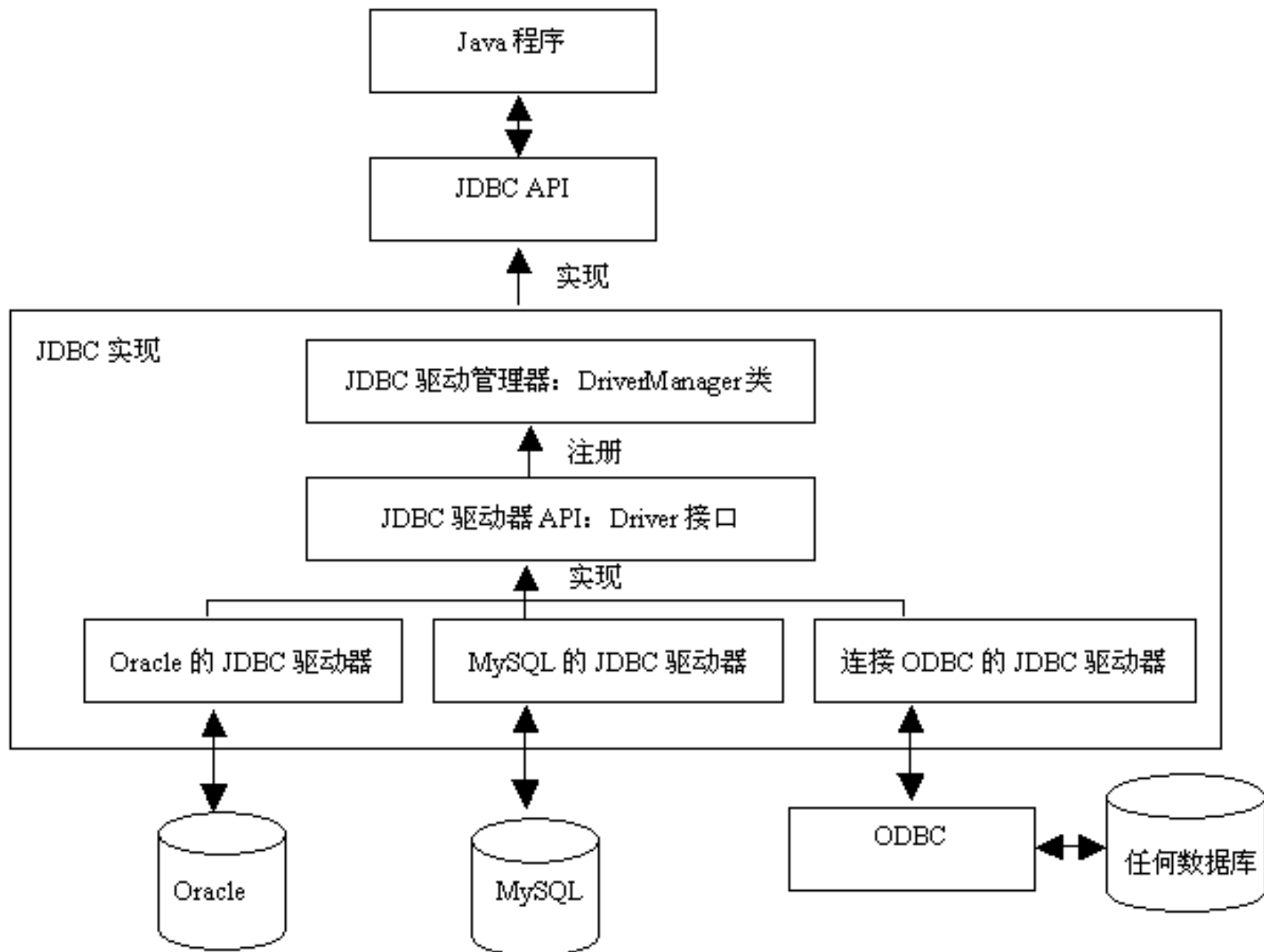


图 12-3 JDBC 的实现



□ Mysql Connector/J

- <http://www.mysql.com/downloads/mysql/>
- <http://dev.mysql.com/downloads/connector/j/>

□ Microsoft SQL Server JDBC Driver

- <http://www.microsoft.com/zh-cn/download/details.aspx?id=11774>

□ Oracle Database JDBC Driver

- <http://www.oracle.com/technetwork/database/features/jdbc/index.html>





□ Sqlite JDBC

- Sqlite 是嵌入式的，Android中就用它
- <https://github.com/crawshaw/sqlitejdbc>
- 参考文章 <http://ttitfly.iteye.com/blog/143934>
- <http://blog.csdn.net/liuhe688/article/details/6712782>
- 或 <http://www.ch-werner.de/javasqlite/>

□ JavaDB

- 这是JDK自带的，是derby的定制版本，纯Java开发
- 有嵌入的，有网络的，NetBeans中直接可用
- 驱动程序在 `jdk\db\lib` 下



□在运行时，要将相应的jar加到classpath中

- classpath 可以使用系统环境变量，也可以在命令行上使用，如 `java -cp ".;c:\Program Files\java1.7\jre\lib\rt.jar" xxxxx`
- 如果是用JDBC-ODBC桥，则加上 `java1.7\jre\lib\rt.jar`
- 遗憾的是Java8不再支持ODBC

□如果是IDE则要到Libraries(库)中

- Eclipse中，选中工程名——点击右键——选择 “Properties”——“Java Build Path”——“Libraries”——“Add External JARs”
- NetBeans中，项目—库—点右键—添加jar文件



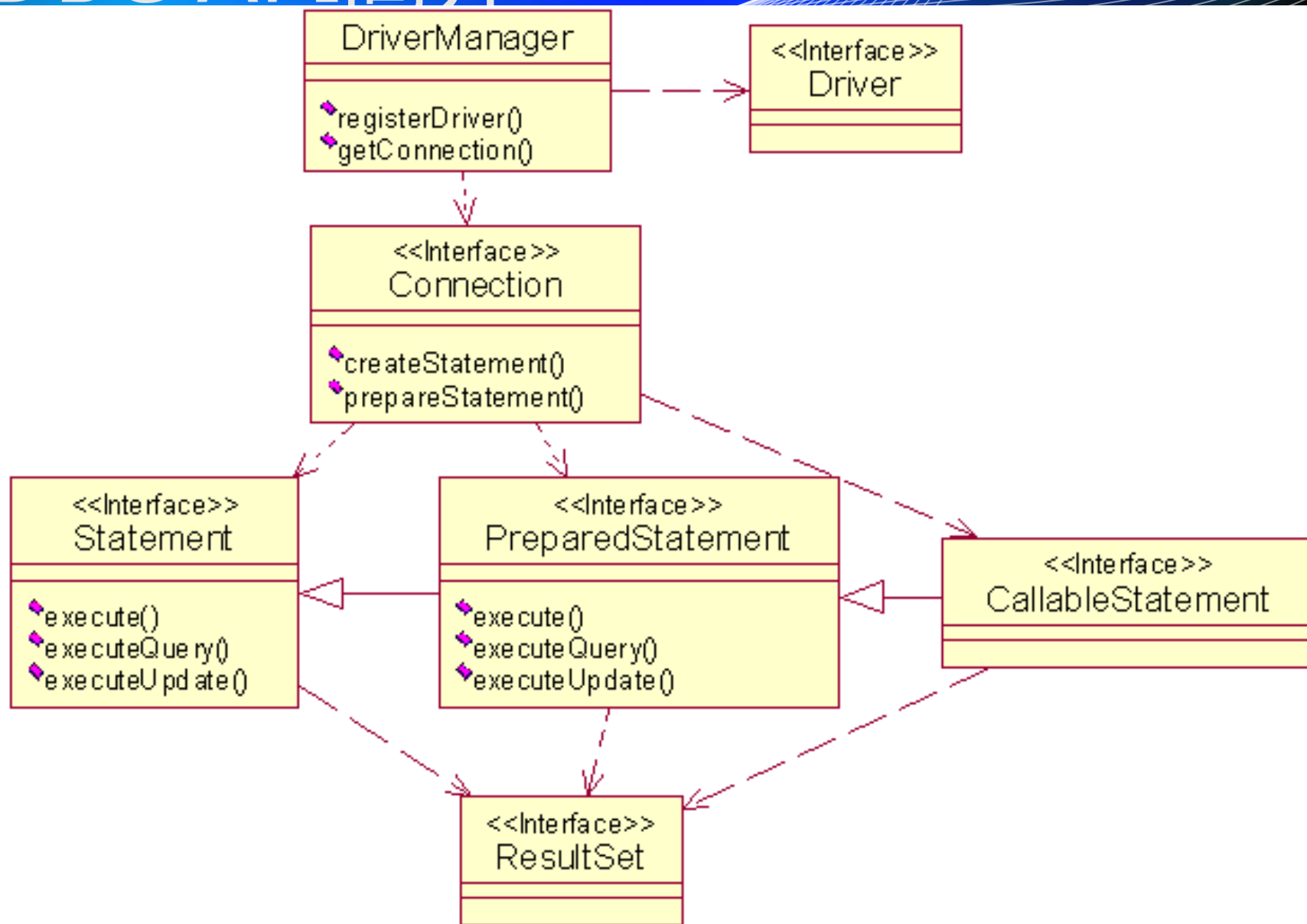
关键的类或接口

- Connection 连接
- Statement 语句
 - ▣ 或用 PreparedStatement
- ResultSet 结果集
 - ▣ 可以用 next() 方法来遍历所有的记录



12.3 JDBC API简介

- java.sql





加载驱动及连接数据库

- // 加载驱动程序
- `Class.forName("org.sqlite.JDBC");`
- // 得到与数据库的连接
- `String connString = "jdbc:sqlite:d:/test3.db"; // 连接串"`
- `Connection conn =
DriverManager.getConnection(connString);`



执行语句

- // 得到一个Statement对象
- `stmt = con.createStatement();`
- // 执行非查询
- `stmt.executeUpdate("delete from DemoTable");`
- // 查询数据库得到记录集
- `rs = stmt.executeQuery(`
• `"SELECT * from DemoTable ORDER BY test_id");`



- // 显示表DemoTable中的所有记录
- System.out.println("Display all results:");
- while(rs.**next**())
- {
- int theInt= rs.**getInt**("test_id");
- String str = rs.**getString**("test_val");
- System.out.println("\ttest_id= " + theInt + "\tstr = " + str);
- }

几个示例



- 请见 JDBC Demo 目录下
 - ▣ JDBC4Sqlite.java 连mysql
 - ▣ JDBC4Mysql.java 连mysql
 - ▣ JDBC4JavaDB.java 连JavaDB
 - ▣ JDBC4SqlServer.java 连sql server



编写基于数据库的应用

- Client/Server模式 (客户/服务模式)
 - 数据库提供服务，称为Server
 - 程序是客户端，称为Client
 - 客户端通过SQL命令来向服务端发出请求
- 编程的任务
 - 通过用户界面获取参数，并形成SQL语句
 - 建立与数据库的连接
 - 向数据库管理系统发出命令 (SQL)
 - 从数据库得到数据并显示到用户界面



使用PreparedStatement

- PreparedStatement 表示经过编译的语句
 - ▣ 如果多次使用同样的语句，执行速度略快一些
 - ▣ 可以方便地加上参数
- 例如：
 - `String sql2 = "select count(*) from people where name like ? and age> ?";`
 - `PreparedStatement pstmt = connection.prepareStatement(sql2);`
 - `pstmt.setString(1, "%r%");`
 - `pstmt.setInt(2, 18);`
 - `ResultSet rs2 = pstmt.executeQuery();`
 - `if(rs2.next()) System.out.println(rs2.getInt(1));`



安全方面的考虑

- PreparedStatement
- 还有安全方面的原因：避免sql注入攻击
- Select count(*) from User
 - where name='aaa' and pwd='bbb'
 - 如果有人将口令输入 ' or 1=1; delete from table ccc; --
 - 则上面的语句变为：
 - where name='aaa' and pwd=' ' or 1=1; delete from table ccc; --'



使用JTable显示数据表

- 使用JTable显示数据表

- 可以直接用二维数组或Vector

- JTable(Object[][] rowData, Object[] columnNames))
 - JTable(Vector rowData, Vector columnNames)

- 使用数据模型

- JTable(**TableModel** tm)
 - 可继承自 AbstractTableModel 或 DefaultTableModel
 - 其中有 getColumnCount() getRowCount() getValueAt(r,c)
 - **示例** JDBCJTable.java

- 别忘了使用JScrollPane



- 简单通讯录 MyContacts.java
- 关键点：
 - 界面的参数转为sql语句
 - 数据访问层
 - DAO (Data Access Object)
 - 对象实体 (Bean 或叫 Entity 或 Object , POJO)
 - ORMapping (对象关系映射)