第10天 综合案例

**【学习目标】理解、了解、应用、记忆**

通过今天的学习，巩固之前的知识点

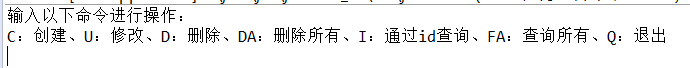
1. **【应用】综合案例**
2. 【理解】阐述综合案例的需求是什么
3. 【应用】独立完成项目环境的搭建工作
4. 【应用】独立编写一个JDBC获取连接的的工具类
5. 【应用】使用键盘录入及switch语句完成初期项目逻辑部分代码
6. 【应用】独立编写代码完成查询所有操作
7. 【应用】独立编写代码完成通过id查询商品详情
8. 【应用】独立编写代码完成添加商品功能
9. 【应用】独立编写代码完成商品修改功能
10. 【应用】独立编写代码根据id删除商品
11. 【应用】独立编写代码完成商品的批量删除操作

# 综合案例

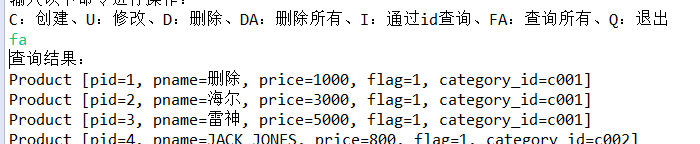
## 需求分析

### 需求

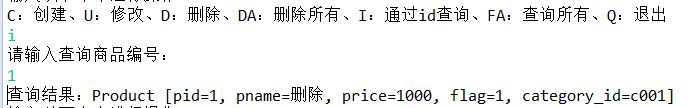
* 运行



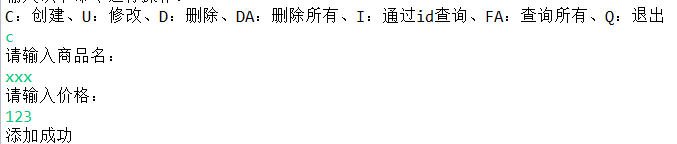
* 查询所有



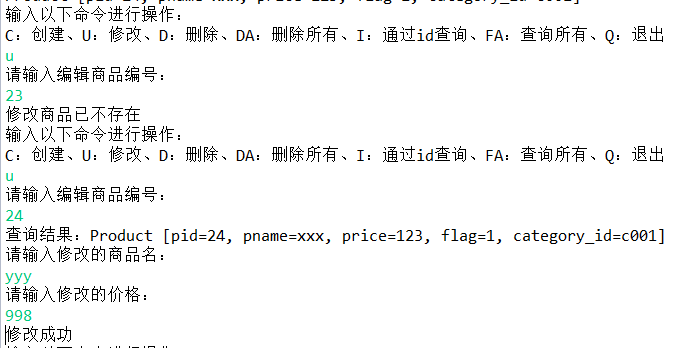
* 通过id查询详情



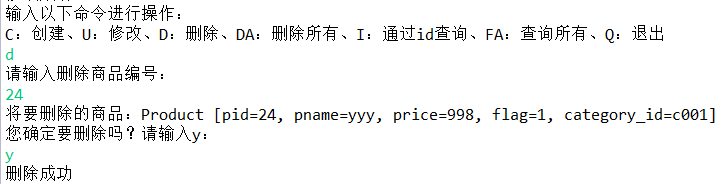
* 添加



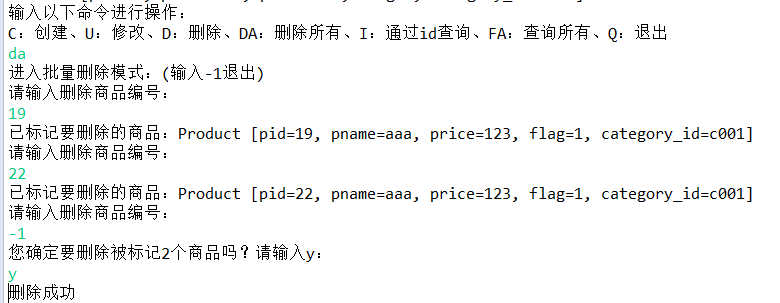
* 修改



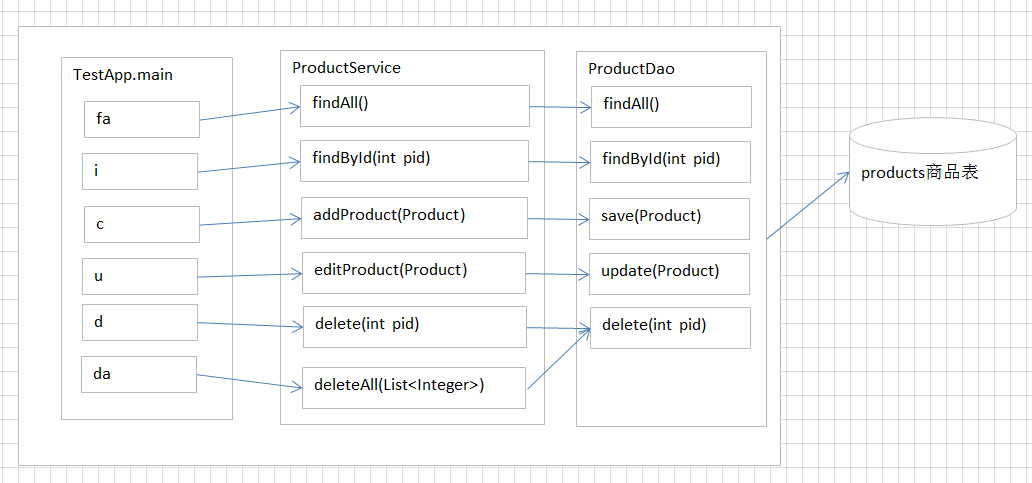
* 通过id删除



* 删除所有



### 案例分析



* 程序将划分层次
  + com.igeek.domain javaBean
  + com.igeek.utils 工具类
  + com.igeek.dao dao层
  + com.igeek.service service层（业务层，一般情况内容非常少，直接调用dao，只有与业务挂钩时才能体现出来的）

## 准备工作

### 准备数据

|  |
| --- |
| **create** **database** igeek;  **use** igeek;  **create** **table** product(  pid **int** **primary** **key** auto\_increment,  pname **varchar**(30),  price **int**,  flag **VARCHAR**(2), # 是否上架标记：1 表示上架 0 表示下架  category\_id **varchar**(32)  );  **INSERT** **INTO** product(pname,price,flag,category\_id) **VALUES**  ('IBM',5500,1,'c001'),  ('长城',3200,1,'c001'),  ('惠普',5000,1,'c001'),  ('华为',3800,1,'c002'),  ('小米',2000,1,'c002'),  ('OPPO',3400,1,'c002'),  ('NOKIA',2000,1,'c002'),  ('佰草集',800,1,'c003'),  ('玉兰油',200,1,'c003'),  ('大宝',5,1,'c003') |

### 控制台输入

|  |
| --- |
| **try** {  // 提供处理流，接收键盘输入  BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(System.***in***));  // 程序无限执行，处理不同命令  **while** (**true**) {  System.***out***.println("输入一下命令进行操作：");  System.***out***.println("C：创建、U：修改、D：删除、DA：批量删除、I：通过id查询、FA：查询所有、Q：退出");  // 接收命令  String line = reader.readLine();  // 根据不同的命令做出不同处理  **switch** (line.toUpperCase()) {  **case** "C":  // 添加  *create*(reader);  **break**;  **case** "U":  // 修改  *update*(reader);  **break**;  **case** "D":  // 删除  *delete*(reader);  **break**;  **case** "DA":  // 批量删除  *deleteAll*(reader);  **break**;  **case** "I":  // 查询详情  *findById*(reader);  **break**;  **case** "FA":  // 查询所有  *findAll*(reader);  **break**;  **case** "Q":  System.***out***.println("欢迎下次使用");  System.*exit*(-1);  **break**;  **default**:  **break**;  }  }  } **catch** (Exception e) {  e.printStackTrace();  System.***out***.println("本次操作异常，请稍后重试");  } |

## 查询功能

### 查询所有

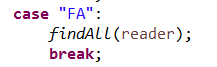
#### dao层

|  |
| --- |
| /\*\*  \* 查询所有  \* **@return**  \* **@throws** SQLException  \*/  **public** List<Product> findAll() **throws** SQLException{    QueryRunner queryRunner = **new** QueryRunner(JdbcUtils.*getDataSource*());  String sql = "select \* from products";  Object[] params = {};    **return** queryRunner.query(sql, **new** BeanListHandler<Product>(Product.**class**), params);  } |

#### service层

|  |
| --- |
| /\*\*  \* 查询所有  \*  \* **@return**  \* **@throws** SQLException  \*/  **public** List<Product> findAll() **throws** SQLException {  // ProductDao productDao = new ProductDao();  **return** productDao.findAll();  } |

#### 入口



|  |
| --- |
| /\*\*  \*  \* **@Title**: findAll  \* **@Description**: 查询所有  \* **@param** reader  \*/  **private** **static** **void** findAll(BufferedReader reader) {  **try** {  ProductService productService = **new** ProductService();  List<Product> list = productService.findAll();  System.***out***.println("查询结果：");  **if** (list != **null**) {  **for** (Product product : list) {  System.***out***.println(product);  }  System.***out***.println("共" + list.size() + "条");  } **else** {  System.***out***.println("没有查询结果");  }  } **catch** (Exception e) {  System.***out***.println("本次查询失败，请稍后重试");  }  } |

### 通过id查询详情

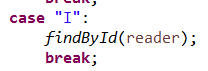
#### dao层

|  |
| --- |
| /\*\*  \* 通过id查询详情  \* **@param** pid  \* **@return**  \* **@throws** SQLException  \*/  **public** Product findById(Integer pid) **throws** SQLException{  QueryRunner queryRunner = **new** QueryRunner(JdbcUtils.*getDataSource*());  String sql = "select \* from products where pid = ?";  Object[] params = {pid};  **return** queryRunner.query(sql, **new** BeanHandler<Product>(Product.**class**), params);  } |

#### service层

|  |
| --- |
| /\*\*  \* 通过id查询详情  \*  \* **@param** pid  \* **@return**  \* **@throws** SQLException  \*/  **public** Product findById(Integer pid) **throws** SQLException {  // ProductDao productDao = new ProductDao();  **return** productDao.findById(pid);  } |

#### 入口



|  |
| --- |
| /\*\*  \*  \* **@Title**: findById  \* **@Description**: 根据Id进行查询  \* **@param** reader  \*/  **private** **static** **void** findById(BufferedReader reader) {  **try** {  // 获得数据  System.***out***.println("请输入查询的商品编号：");  String pidStr = reader.readLine();  Integer pid = Integer.*parseInt*(pidStr);  // 查询  ProductService productService = **new** ProductService();  Product product = productService.findById(pid);  // 显示  **if** (product != **null**) {  System.***out***.println("查询结果：" + product);  } **else** {  System.***out***.println("没有查询结果");  }  } **catch** (Exception e) {  System.***out***.println("查询异常，请稍后重试");  }  } |

## 增加功能

#### dao层

|  |
| --- |
| /\*\*  \* 保存  \* **@param** product  \* **@throws** SQLException  \*/  **public** **void** save(Product product) **throws** SQLException {  QueryRunner queryRunner = **new** QueryRunner(JdbcUtils.*getDataSource*());  String sql = "INSERT INTO products(pname,price,flag,category\_id) VALUES(?,?,?,?);";  Object[] params = {product.getPname(),product.getPrice(),product.getFlag(),product.getCategory\_id()};  queryRunner.update(sql, params);  } |

#### service层

|  |
| --- |
| /\*\*  \* 添加商品  \*  \* **@param** product  \* **@throws** SQLException  \*/  **public** **void** addProduct(Product product) **throws** SQLException {  productDao.save(product);  } |

#### 入口



|  |
| --- |
| /\*\*  \*  \* **@Title**: create  \* **@Description**: 添加商品  \* **@param** reader  \*/  **private** **static** **void** add(BufferedReader reader) {  **try** {  // 获得数据 ，并封装  System.***out***.println("请输入商品名称：");  String pname = reader.readLine();  System.***out***.println("请输入商品价格：");  String priceStr = reader.readLine();  Integer price = Integer.*parseInt*(priceStr);  Product product = **new** Product(pname, price, "1", "c001");  // 添加  ProductService productService = **new** ProductService();  productService.addProduct(product);  System.***out***.println("添加成功");  } **catch** (Exception e) {  System.***out***.println("添加异常，请稍后重试");  }  } |

## 修改功能

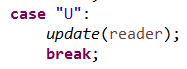
dao层

|  |
| --- |
| /\*\*  \* 修改  \* **@param** product  \* **@throws** SQLException  \*/  **public** **void** update(Product product) **throws** SQLException {  QueryRunner queryRunner = **new** QueryRunner(JdbcUtils.*getDataSource*());  String sql = "update products set pname=?,price=?,flag=?,category\_id=? where pid = ?";  Object[] params = {product.getPname(),product.getPrice(),product.getFlag(),  product.getCategory\_id() , product.getPid()};  queryRunner.update(sql, params);  } |

#### service层

|  |
| --- |
| /\*\*  \* 修改商品  \*  \* **@param** product  \* **@throws** SQLException  \*/  **public** **void** editProduct(Product product) **throws** SQLException {  productDao.update(product);  } |

#### 入口



|  |
| --- |
| /\*\*  \* 修改商品  \*  \* **@Title**: update  \* **@Description**: **TODO**(这里用一句话描述这个方法的作用)  \* **@param** reader  \*/  **private** **static** **void** update(BufferedReader reader) {  **try** {  // 获得数据  System.***out***.println("请输入要编辑的商品编号：");  String pidStr = reader.readLine();  Integer pid = Integer.*parseInt*(pidStr);  // 查询  ProductService productService = **new** ProductService();  Product product = productService.findById(pid);  **if** (product == **null**) {  System.***out***.println("修改商品已不存在");  **return**;  }  // 修改  System.***out***.println("请输入需要修改的商品名称：");  String pname = reader.readLine();  System.***out***.println("请输入需要修改商品价格：");  String priceStr = reader.readLine();  Integer price = Integer.*parseInt*(priceStr);  // \* 将用户输入的数据修改查询数据，最后更新  product.setPname(pname);  product.setPrice(price);  // \* 修改操作  productService.editProduct(product);  System.***out***.println("修改成功");  } **catch** (Exception e) {  System.***out***.println("修改异常，请稍后重试");  }  } |

## 删除功能

### 通过id删除

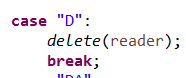
#### dao层

|  |
| --- |
| /\*\*  \* 删除，支持事务  \* **@param** pid  \* **@throws** SQLException  \*/  **public** **void** delete(Integer pid) **throws** SQLException {  QueryRunner queryRunner = **new** QueryRunner();  String sql = "delete from products where pid = ?";  Object[] params = {pid};  queryRunner.update(JdbcUtils.*getConnection*() , sql, params);  } |

#### service层

|  |
| --- |
| /\*\*  \* 删除  \*  \* **@param** pid  \*/  **public** **void** delete(Integer pid) {  Connection conn = **null**;  **try** {  conn = JdbcUtils.*getConnection*();  // 开启  conn.setAutoCommit(**false**);  productDao.delete(pid);  // 提交  DbUtils.*commitAndClose*(conn);  } **catch** (Exception e) {  // 回滚  DbUtils.*rollbackAndCloseQuietly*(conn);  // 通知调用者  **throw** **new** RuntimeException(e);  }  } |

#### 入口



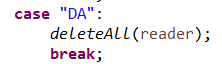
|  |
| --- |
| /\*\*  \*  \* **@Title**: delete  \* **@Description**: 删除  \* **@param** reader  \*/  **private** **static** **void** delete(BufferedReader reader) {  **try** {  // 获得数据  System.***out***.println("请输入删除商品编号：");  String pidStr = reader.readLine();  Integer pid = Integer.*parseInt*(pidStr);  // 查询  ProductService productService = **new** ProductService();  Product product = productService.findById(pid);  **if** (product == **null**) {  System.***out***.println("删除商品不存在");  **return**;  }  System.***out***.println("将要删除的商品：" + product);  System.***out***.println("您确定要删除吗？请输入y");  String yes = reader.readLine();  **if** ("y".equalsIgnoreCase(yes)) {  productService.delete(pid);  System.***out***.println("删除成功");  } **else** {  System.***out***.println("操作已取消");  }  } **catch** (Exception e) {  System.***out***.println("删除异常，请稍后重试");  }  } |

### 批量删除

#### service层

|  |
| --- |
| /\*\*  \* 批量删除  \*  \* **@param** idList  \*/  **public** **void** deleteAll(List<Integer> idList) {  Connection conn = **null**;  **try** {  conn = JdbcUtils.*getConnection*();  // 开启  conn.setAutoCommit(**false**);  **if** (idList != **null**) {  **for** (Integer id : idList) {  productDao.delete(id);  }  }  // 提交  DbUtils.*commitAndClose*(conn);  } **catch** (Exception e) {  // 回滚  DbUtils.*rollbackAndCloseQuietly*(conn);  // 通知调用者  **throw** **new** RuntimeException(e);  }  } |

#### 入口



|  |
| --- |
| /\*\*  \*  \* **@Title**: deleteAll  \* **@Description**: 批量删除  \* **@param** reader  \*/  **private** **static** **void** deleteAll(BufferedReader reader) {  **try** {  System.***out***.println("进入批量删除模式：(输入-1退出)");  List<Integer> idList = **new** ArrayList<Integer>();  ProductService productService = **new** ProductService();  **while** (**true**) {  // 获得数据  System.***out***.println("请输入删除商品编号：");  String pidStr = reader.readLine();  Integer pid = Integer.*parseInt*(pidStr);  // -1 退出  **if** (pid == -1) {  **break**;  }  // 查询  Product product = productService.findById(pid);  **if** (product == **null**) {  System.***out***.println("商品【" + pid + "】已不存在");  **continue**;  }  idList.add(pid);  System.***out***.println("已标记要删除的商品" + product);  }  // 查询批量  System.***out***.println("您确定要删除，已经标记[数量为" + idList.size() + "]的商品吗？请输入y");  String yes = reader.readLine();  **if** ("y".equalsIgnoreCase(yes)) {  productService.deleteAll(idList);  System.***out***.println("删除成功");  } **else** {  System.***out***.println("操作已取消");  }  } **catch** (Exception e) {  System.***out***.println("删除异常，请稍后重试");  }  } |