|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **□ 수행평가 - 빅데이터를 활용한 IoT 시스템 개발(feat.커넥티드카)** | | | | | | |
|  |  |  | |  | |  |
| **과정명** | | 빅데이터를 활용한 IoT 시스템 개발(feat.커넥티드카) | | | | |
| **교과목명** | | IoT 운영시스템 구축 기반기술 | | **훈련교사** | | 이진만 |
| **과정명** | | SW기초기술이해  응용SW기술이해  Database 구문이해 | | | | |
| **수행날짜** | | 2019.06.10 | 훈련생명 | | 이경헌 | |
| **과제개요** | | | | | | |
| 1. Workshop 교재 124Page의 테이블을 구축 한다. 2. DML, DDL을 작성하고 Table의 구조를 작성한다. 3. UML을 이용하여 프로그램을 설계 한다. 4. JDBC API를 이용하여 상품에 대한 CRUD 프로그램을 작성 한다.   작성 내용   1. ERD 2. DDL, DML 3. UML 4. 작성 코드   1 **ERD**  C:\Users\student\Desktop\erd.PNG   |  | | --- | |  | | | | | | | |
| 2 **DDL**   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10 | CREATE TABLE PRODUCTS(  PDNO NUMBER(10,0),  PDNAME VARCHAR2(10),  PDSUBNAME VARCHAR2(10),  FACTNO VARCHAR2(5),  PDDATE DATE,  PDCOST NUMBER(10,0),  PDPRICE NUMBER(10,0),  PDAMOUNT NUMBER(10,0)  ); | [cs](http://colorscripter.com/info#e) |  |  |  |  | | --- | --- | --- | | 1  2  3  4  5 | CREATE TABLE FACTORY(  FACTNO VARCHAR2(5),  FACNAME VARCHAR2(14),  FACLOC VARCHAR2(13)  ); | [cs](http://colorscripter.com/info#e) |  |  |  |  | | --- | --- | --- | | 1  2  3 | ALTER TABLE PRODUCTS ADD PRIMARY KEY(PDNO);  ALTER TABLE FACTORY ADD PRIMARY KEY(FACTNO);  ALTER TABLE PRODUCTS ADD FOREIGN KEY(FACTNO) REFERENCES FACTORY(FACTNO); | [cs](http://colorscripter.com/info#e) | | | | | | | |
| 2 **DML**   |  |  |  |  | | --- | --- | --- | --- | | 1  2  3  4  5 | -- PRODUCTS Table  INSERT INTO PRODUCTS VALUES(2345,'FRUIT','APPLE',10,TO\_DATE('2019-06-05','YYYY-MM-DD'),1500,2700,85);  INSERT INTO PRODUCTS VALUES(5678,'VEGETABLE','PUMPKIN',10,TO\_DATE('2019-06-08','YYYY-MM-DD'),700,1300,40);  INSERT INTO PRODUCTS VALUES(7937,'FISH','MACKEREL',20,TO\_DATE('2019-06-10','YYYY-MM-DD'),3000,7500,27);  INSERT INTO PRODUCTS VALUES(3859,'MEAT','PIG',20,TO\_DATE('2019-06-07','YYYY-MM-DD'),8000,15000,20); | | [cs](http://colorscripter.com/info#e) | | 1  2  3 | -- FACTORY Table  INSERT INTO FACTORY VALUES('10','ABC','SEOUL');  INSERT INTO FACTORY VALUES('20','DEF','BUSAN'); | [cs](http://colorscripter.com/info#e) | | | | | | | |
| 3 **UML**  **C:\Users\student\Desktop\UML.jpg** | | | | | | |
| 4 **소스코드**  vo.Factory   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | package vo;    public class Factory {      private String factno;      private String facname;      private String facloc;      public String getFactno() {          return factno;      }      public void setFactno(String factno) {          this.factno = factno;      }      public String getFacname() {          return facname;      }      public void setFacname(String facname) {          this.facname = facname;      }      public String getFacloc() {          return facloc;      }      public void setFacloc(String facloc) {          this.facloc = facloc;      }      public Factory() {      }      @Override      public String toString() {          return "Factory [factno=" + factno + ", facname=" + facname + ", facloc=" + facloc + "]";      }      public Factory(String factno, String facname, String facloc) {          super();          this.factno = factno;          this.facname = facname;          this.facloc = facloc;      }    }    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   vo.Products   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47 | package vo;    import java.sql.Date;    public class Products {      private     int pdno;      private String pdname;      private String pdsubname;      private     String factno;      private int pdcost;      private int pdprice;      private    int pdamount;      Date pddate;      public Products() {      }      public Products(int pdno, String pdname, String pdsubname, String factno, Date pddate, int pdcost,              int pdprice,    int pdamount) {          this.pdno = pdno;          this.pdname = pdname;          this.pdsubname = pdsubname;          this.factno = factno;          this.pddate = pddate;          this.pdcost = pdcost;          this.pdprice = pdprice;          this.pdamount = pdamount;      }        public int getPdno() {          return pdno;      }      public void setPdno(int pdno) {          this.pdno = pdno;      }      public String getPdname() {          return pdname;      }      public void setPdname(String pdname) {          this.pdname = pdname;      }      public String getPdsubname() {          return pdsubname;      }      public void setPdsubname(String pdsubname) {          this.pdsubname = pdsubname;      }      public String getFactno() {          return factno;      }      public void setFactno(String factno) {          this.factno = factno;      }      public Date getPddate() {          return pddate;      }      public void setPddate(Date pddate) {          this.pddate = pddate;      }      public int getPdcost() {          return pdcost;      }      public void setPdcost(int pdcost) {          this.pdcost = pdcost;      }      public int getPdprice() {          return pdprice;      }      public void setPdprice(int pdprice) {          this.pdprice = pdprice;      }      public int getPdamount() {          return pdamount;      }      public void setPdamount(int pdamount) {          this.pdamount = pdamount;      }        @Override      public String toString() {          return "Product [pdno=" + pdno + ", pdname=" + pdname + ", pdsubname=" + pdsubname + ", factno=" + factno                  + ", pddate=" + pddate + ", pdcost=" + pdcost + ", pdprice=" + pdprice + ", pdamount=" + pdamount + "]";      }  }    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   vo.Join   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | package vo;    import java.util.ArrayList;    public class Join {      private ArrayList<Factory> f;      private ArrayList<Products> p;      public Join() {      }      public Join(ArrayList<Factory> f, ArrayList<Products> p) {          super();          this.f = f;          this.p = p;      }        public ArrayList<Factory> getF() {          return f;      }        @Override      public String toString() {          return "Join [f=" + f + ", p=" + p + "]";      }      public void print() {          int a = this.p.size();          int b = this.f.size();          for(int i=0;i<a;i++) {              for(int j=0;j<b;j++) {                  if(this.p.get(i).getFactno().equals(this.f.get(j).getFactno()))                      System.out.println(this.p.get(i) + " "+ this.f.get(j));              }          }      }      public void setF(ArrayList<Factory> f) {          this.f = f;      }        public ArrayList<Products> getP() {          return p;      }        public void setP(ArrayList<Products> p) {          this.p = p;      }    }    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   frame.Sql   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | package frame;    public class Sql {      public static String insertProducts = "INSERT INTO PRODUCTS VALUES(?,?,?,?,?,?,?,?)";      public static String DeleteProducts = "DELETE FROM PRODUCTS WHERE PDNO = ?";      public static String UpdateProducts = "UPDATE PRODUCTS SET PDNAME=?,PDSUBNAME=?,FACTNO=?,PDDATE=?,PDCOST=?,PDPRICE=?,PDAMOUNT=? WHERE PDNO=?";      public static String SelectProducts = "SELECT \* FROM PRODUCTS WHERE PDNO = ?";      public static String SelectallProducts = "SELECT \* FROM PRODUCTS";        public static String insertFactory = "INSERT INTO FACTORY VALUES(?,?,?)";      public static String DeleteFactory = "DELETE FROM FACTORY WHERE FACTNO = ?";      public static String UpdateFactory = "UPDATE FACTORY SET FACNAME =?, FACLOC = ? WHERE FACTNO=?";      public static String SelectFactory = "SELECT \* FROM FACTORY WHERE FACTNO = ?";      public static String SelectallFactory = "SELECT \* FROM FACTORY";      public static String selectJoin = "select P.PDNO , P.PDNAME , F.FACTNO , F.FACNAME, P.PDSUBNAME , P.FACTNO , P.PDCOST. P.PDPRICE, P.PDAMOUNT , F.FACLOC from PRODUCTS P, FACTORY F" + "where P.FACTNO = F.FACTNO";    }    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   frame.Dao   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | package frame;    import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import vo.Join;  public abstract class Dao<K, V> {      public void close(Connection con) {          if (con != null)              try {                  con.close();              } catch (SQLException e) {                  e.printStackTrace();              }      }      public void close(PreparedStatement con) {          if (con != null)              try {                  con.close();              } catch (SQLException e) {                  e.printStackTrace();              }      }      public void close(ResultSet con) {          if (con != null)              try {                  con.close();              } catch (SQLException e) {                  e.printStackTrace();              }      }      public abstract void insert(V v,Connection con) throws Exception;      public abstract void delete(K k,Connection con) throws Exception;      public abstract void update(V v,Connection con) throws Exception;      public abstract V select(K k,Connection con) throws Exception;      public abstract ArrayList<V> select(Connection con) throws Exception;  }    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   frame.Biz   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47 | package frame;    import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.SQLException;  import java.util.ArrayList;  import vo.Join;  public abstract class Biz <K,V> {      private String id;      private String pwd;      private String url;      public Biz() {          try {              Class.forName("oracle.jdbc.driver.OracleDriver");              // Oracle DB에 연결 할 것이다! 선언          } catch (ClassNotFoundException e) {              System.out.println("Driver Loading err");          }          id = "db";          pwd = "db";          url = "jdbc:oracle:thin:@" + "70.12.50.228" + ":" + "1521" + ":xe";      }      public Connection getcon() {          Connection con = null;          try {              con = DriverManager.getConnection(url, id, pwd);              con.setAutoCommit(false);          } catch (SQLException e) {              e.printStackTrace();          }          return con;      }      public void close(Connection con) {          if (con != null)              try {                  con.close();              } catch (SQLException e) {                  e.printStackTrace();              }      }      public abstract void register(V v) throws Exception;      public abstract void remove(K k) throws Exception;      public abstract void modify(V v) throws Exception;      public abstract V get(K k) throws Exception;      public abstract ArrayList<V> get() throws Exception;  }    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   com.ProductsDao   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61 | package com;    import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.util.ArrayList;    import frame.Dao;  import frame.Sql;  import vo.Factory;  import vo.Join;  import vo.Products;  public class ProductsDao extends Dao<Integer, Products> {      @Override      public void insert(Products v, Connection con) throws Exception {          PreparedStatement pstmt = null;          pstmt = con.prepareStatement(Sql.insertProducts);          try {              pstmt.setInt(1, v.getPdno());              pstmt.setString(2, v.getPdname());              pstmt.setString(3, v.getPdsubname());              pstmt.setString(4, v.getFactno());              pstmt.setDate(5, v.getPddate());              pstmt.setInt(6, v.getPdcost());              pstmt.setInt(7, v.getPdprice());              pstmt.setInt(8, v.getPdamount());              pstmt.executeUpdate();          } catch (Exception e) {              throw e;          } finally {              close(pstmt);          }      }      @Override      public void delete(Integer k, Connection con) throws Exception {          PreparedStatement pstmt = null;          pstmt = con.prepareStatement(Sql.DeleteProducts);          try {              pstmt.setInt(1, k);              pstmt.executeUpdate();          } catch (Exception e) {              throw e;          } finally {              close(pstmt);          }      }      @Override      public void update(Products v, Connection con) throws Exception {          PreparedStatement pstmt = null;          pstmt = con.prepareStatement(Sql.UpdateProducts);          try {              pstmt.setInt(8, v.getPdno());              pstmt.setString(1, v.getPdname());              pstmt.setString(2, v.getPdsubname());              pstmt.setString(3, v.getFactno());              pstmt.setDate(4, v.getPddate());              pstmt.setInt(5, v.getPdcost());              pstmt.setInt(6, v.getPdprice());              pstmt.setInt(7, v.getPdamount());              pstmt.executeUpdate();          } catch (Exception e) {              throw e;          } finally {              close(pstmt);          }      }      @Override      public Products select(Integer k, Connection con) throws Exception {          PreparedStatement psm = null;          ResultSet rs = null;          Products result;          psm = con.prepareStatement(Sql.SelectProducts);          try {              psm.setInt(1, k);              psm.executeUpdate();              rs = psm.executeQuery();              rs.next();              result = new Products(rs.getInt("PDNO"),rs.getString("PDNAME"),rs.getString("PDSUBNAME"),rs.getString("FACTNO"),rs.getDate("PDDATE"),rs.getInt("PDCOST"),rs.getInt("PDPRICE"),rs.getInt("PDAMOUNT"));          } catch (Exception e) {              throw e;          } finally {              close(psm);              close(rs);          }          return result;      }      @Override      public ArrayList<Products> select(Connection con) throws Exception {          ArrayList<Products> result = new ArrayList<>();          PreparedStatement psm = null;          ResultSet rs = null;          Products u;          psm = con.prepareStatement(Sql.SelectallProducts);          try {              psm.executeUpdate();              rs = psm.executeQuery();              while (rs.next()) {                  u = new Products(rs.getInt("PDNO"),rs.getString("PDNAME"),rs.getString("PDSUBNAME"),rs.getString("FACTNO"),rs.getDate("PDDATE"),rs.getInt("PDCOST"),rs.getInt("PDPRICE"),rs.getInt("PDAMOUNT"));                  result.add(u);              }          } catch (Exception e) {              throw e;          } finally {              close(psm);              close(rs);          }          return result;      }      *sd*[*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   com.FactoryDao   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103 | package com;    import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.util.ArrayList;    import frame.Dao;  import frame.Sql;  import vo.Factory;  import vo.Join;  import vo.Factory;  public class FactoryDao extends Dao<String, Factory> {      @Override      public void insert(Factory v, Connection con) throws Exception {          PreparedStatement pstmt = null;          try {              pstmt = con.prepareStatement(Sql.insertFactory);              pstmt.setString(1, v.getFactno());              pstmt.setString(2, v.getFacname());              pstmt.setString(3, v.getFacloc());              pstmt.executeUpdate();          } catch (Exception e) {              throw e;          } finally {              close(pstmt);          }        }        @Override      public void delete(String k, Connection con) throws Exception {          PreparedStatement pstmt = null;          try {              pstmt = con.prepareStatement(Sql.DeleteFactory);              pstmt.setString(1, k);                pstmt.executeUpdate();          } catch (Exception e) {              throw e;          } finally {              close(pstmt);          }      }      @Override      public void update(Factory v, Connection con) throws Exception {          PreparedStatement pstmt = null;          try {              pstmt = con.prepareStatement(Sql.UpdateFactory);              pstmt.setString(1, v.getFactno());              pstmt.setString(2, v.getFacname());              pstmt.setString(3, v.getFacloc());              pstmt.executeUpdate();          } catch (Exception e) {              throw e;          } finally {              close(pstmt);          }      }      @Override      public Factory select(String k, Connection con) throws Exception {          PreparedStatement pstmt = null;          Factory result = null;          ResultSet rs = null;          try {              pstmt = con.prepareStatement(Sql.SelectFactory);              pstmt.setString(1, k);              pstmt.executeUpdate();              rs = pstmt.executeQuery();              rs.next();              result = new Factory(rs.getString("FACTNO"), rs.getString("FACNAME"), rs.getString("FACLOC"));          } catch (Exception e) {              throw e;          } finally {              close(pstmt);              close(rs);          }          return result;      }      @Override      public ArrayList<Factory> select(Connection con) throws Exception {          ArrayList<Factory> result = new ArrayList<>();          PreparedStatement pstmt = null;          ResultSet rs = null;          Factory u;          try {              pstmt = con.prepareStatement(Sql.SelectallFactory);              pstmt.executeUpdate();              rs = pstmt.executeQuery();              while (rs.next()) {                  u = new Factory(rs.getString("FACTNO"), rs.getString("FACNAME"), rs.getString("FACLOC"));                  result.add(u);              }          } catch (Exception e) {              throw e;          } finally {              close(pstmt);              close(rs);          }          return result;      }      @Override  }    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   com.ProductsBiz   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86 | package com;  import java.sql.Connection;  import java.util.ArrayList;  import frame.Biz;  import frame.Dao;  import vo.Join;  import vo.Products;  public class ProductsBiz extends Biz<Integer, Products> {      private Dao<Integer, Products> dao;      public ProductsBiz() {          dao = new ProductsDao();      }      @Override      public void register(Products v) throws Exception {          Connection con = null;          try {              con = getcon();              dao.insert(v, con);              con.commit();          } catch (Exception e) {              con.rollback();              throw e;          } finally {              close(con);          }      }      @Override      public void remove(Integer k) throws Exception {          Connection con = null;          try {              con = getcon();              dao.delete(k, con);              con.commit();          } catch (Exception e) {              con.rollback();              throw e;          } finally {              close(con);          }      }      @Override      public void modify(Products v) throws Exception {          Connection con = null;          try {              con = getcon();              dao.update(v, con);              con.commit();          } catch (Exception e) {              con.rollback();              throw e;          } finally {              close(con);          }      }      @Override      public Products get(Integer k) throws Exception {          Connection con = null;          Products u = null;          try {              con = getcon();              u = dao.select(k, con);              con.commit();          } catch (Exception e) {              con.rollback();              throw e;          } finally {              close(con);          }          return u;      }      @Override      public ArrayList<Products> get() throws Exception {          ArrayList<Products> u = null;          Connection con = null;          try {              con = getcon();              u = dao.select(con);          } catch (Exception e) {              throw e;          } finally {              close(con);          }          return u;      }    }    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |   com.FactoryBiz   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87 | package com;  import java.sql.Connection;  import java.util.ArrayList;  import frame.Biz;  import frame.Dao;  import vo.Factory;  import vo.Join;  public class FactoryBiz extends Biz<String, Factory> {      Dao<String, Factory> dao;      public FactoryBiz() {          dao = new FactoryDao();      }      @Override      public void register(Factory v) throws Exception {          Connection con = null;          try {              con = getcon();              dao.insert(v, con);                con.commit();          } catch (Exception e) {              con.rollback();              throw e;          } finally {              close(con);          }      }      @Override      public void remove(String k) throws Exception {          Connection con = null;          try {              con = getcon();              dao.delete(k, con);              con.commit();          } catch (Exception e) {              con.rollback();              throw e;          } finally {              close(con);          }      }      @Override      public void modify(Factory v) throws Exception {          Connection con = null;          try {              con = getcon();              dao.update(v, con);              con.commit();          } catch (Exception e) {              con.rollback();              throw e;          } finally {              close(con);          }      }      @Override      public Factory get(String k) throws Exception {          Connection con = null;          Factory u = null;          try {              con = getcon();              u = dao.select(k, con);          } catch (Exception e) {                throw e;          } finally {              close(con);          }          return u;      }      @Override      public ArrayList<Factory> get() throws Exception {          Connection con = null;          ArrayList<Factory> u = null;          try {              con = getcon();              u = dao.select(con);          } catch (Exception e) {              con.rollback();              throw e;          } finally {              close(con);          }          return u;      }    }    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |     test.WorkshopTest     |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | package test;  import java.sql.Date;  import java.util.ArrayList;  import java.util.Scanner;  import com.FactoryBiz;  import com.ProductsBiz;  import frame.Biz;  import vo.Factory;  import vo.Join;  import vo.Products;  public class WorkshopTest{      public static void main(String[] args) {          Products p = new Products(75, "가방", "에어백", "30", null, 2000, 7000, 10);          //Factory f = new Factory("30", "NIKE", "JEJU");          Scanner sc = new Scanner(System.in);          Biz<Integer, Products> biz = new ProductsBiz();          Biz<String, Factory> biz2 = new FactoryBiz();          ArrayList<Products> pin =null;          ArrayList<Factory> fin = null;          try {              pin = biz.get();              fin = biz2.get();              biz.register(p);              p.setPdamount(9);              biz.modify(p);              Join jj = new Join(fin, pin);              for(Factory tmp : fin)                  System.out.println(tmp);              System.out.println(jj.print());            } catch (Exception e) {              e.printStackTrace();          }      }    }    *[Colored by Color Scripter](http://colorscripter.com/info" \l "e" \t "_blank)* | [cs](http://colorscripter.com/info#e) | | | | | | | |
|  | | | | | | |