

Data Access Object (DAO)

SC363204

Java Web Application Development

การพัฒนาโปรแกรมประยุกต์บนเว็บด้วยภาษาจาวา



Data Access Object

- DAO (หรือ Repository) คือ object ของคลาส ที่มีเมธอดที่เกี่ยวข้องกับการทำงานกับฐานข้อมูล โดยเฉพาะ
- ตัวอย่างคลาส **ProductDAO** อาจประกอบด้วยเมธอด
 - Product **getProduct** (int productId) ใช้ดึงข้อมูลสินค้าตามรหัส
 - ArrayList<Product> **getAllProduct**() ใช้ดึงข้อมูลสินค้าทั้งหมด
 - **createProduct** (Product product) ใช้เพิ่มข้อมูลสินค้าในฐานข้อมูล
 - **updateProduct** (Product product) ใช้แก้ไขข้อมูลสินค้า
 - **deleteProduct** (int productId) ใช้ลบข้อมูลสินค้า

ArrayList

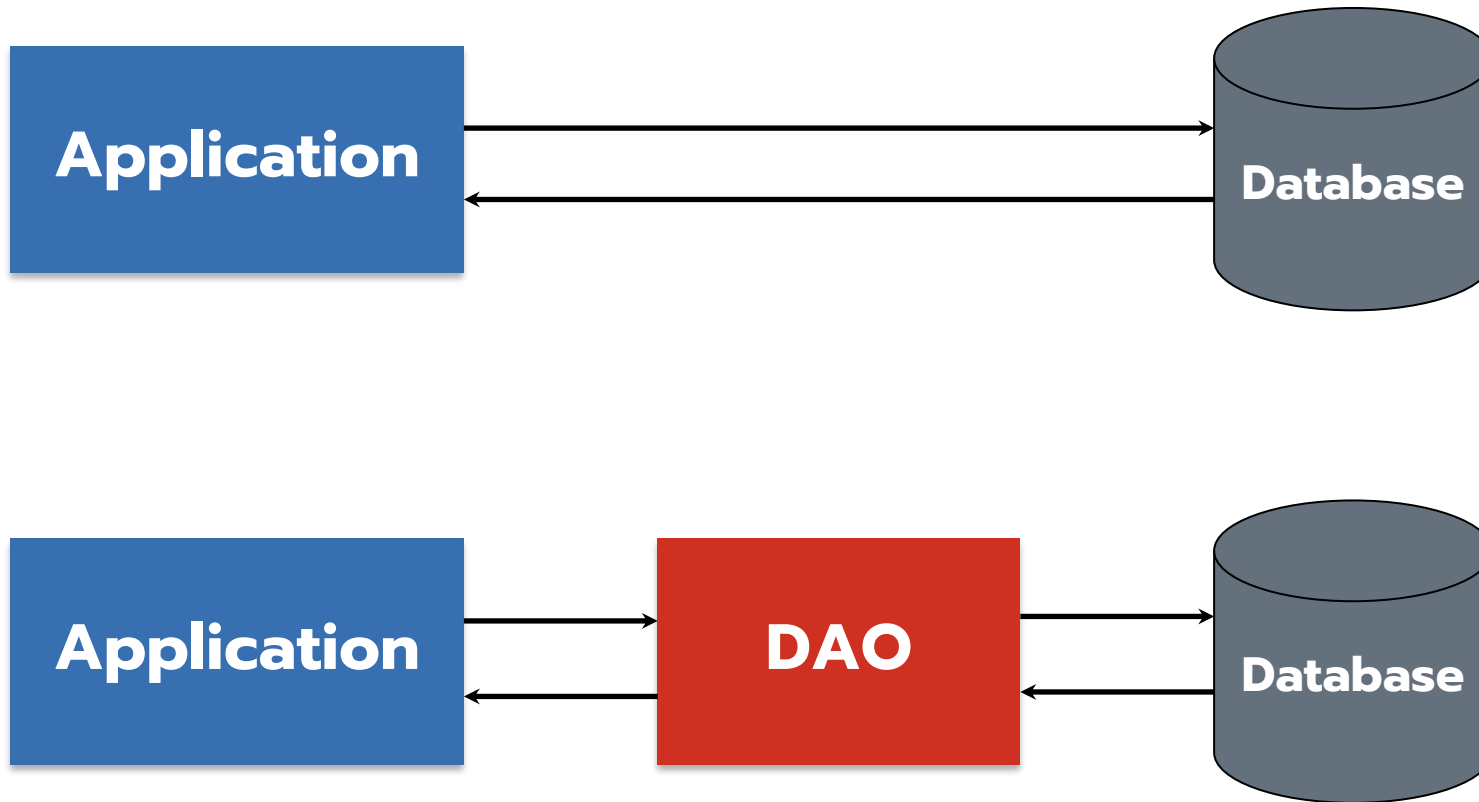
- ArrayList เป็นคลาสมาตรฐานใน Java
- สามารถบรรจุ Object ลงไปได้ Class
- คำสั่งเข้าถึงตำแหน่งของแต่ละค่าข้อมูลจะใช้คำสั่ง ชื่อตัวแปร.get(x);
- x แทนลำดับ index ใน array
- การเรียงลำดับ (Sort) ใน Array จะใช้คำสั่ง Collections.sort(ชื่อตัวแปร);

Assignment

- ให้ทำการวนลูปแสดงข้อมูลทั้งหมดใน List โดยข้อมูลที่แสดงมีการเรียงลำดับตัวอักษร

```
ArrayList<String> cars = new ArrayList<String>();  
cars.add("Volvo");  
cars.add("BMW");  
cars.add("Ford");  
cars.add("Mazda");  
System.out.println(cars);
```

Data Access Object



องค์ประกอบของ DAO

- คลาสเก็บข้อมูล (Entity Class หรือ JavaBeans)
 - มี attribute ที่สอดคล้องกับตาราง 1 ตารางในฐานข้อมูล
 - อาจตั้งชื่อเดียวกับชื่อตาราง เช่น คลาส Product
- คลาส DAO
 - เมธอดสำหรับเชื่อม และปิดการติดต่อฐานข้อมูล
 - เมธอดในการแปลงข้อมูลใน ResultSet เป็น JavaBeans
 - เมธอดค้นหา เพิ่ม ลบ หรือแก้ไข ตามความต้องการของ Application
 - อาจสร้างคลาส DAO 1 คลาส เพื่อทำงานเฉพาะกับตาราง 1 ตาราง โดยตั้งชื่อเดียวกับตาราง เช่น คลาส ProductDAO



JavaBeans

- **JavaBeans** คือ คลาสที่ใช้**ห่อหุ้มข้อมูล**หลากหลายชนิดไว้ด้วยกัน เพื่อใช้ในการ**สร้าง object** สำหรับเก็บข้อมูล**เพื่อส่งไปทำงานที่เมธอด**ต่างๆ เช่น คลาส Person ใช้สร้าง object ในการเก็บข้อมูลลูกค้า ได้แก่ ชื่อ ที่อยู่ เบอร์โทรศัพท์
- คุณสมบัติของคลาส JavaBeans
 - ชื่อตัวแปรควรตั้งชื่อในรูปแบบ camel case
 - การเข้าถึง attribute (หรือตัวแปร) เป็น private
 - มี method สำหรับกำหนดค่าให้กับ attribute ของ Object ชื่อ setXxx()
 - มี method สำหรับอ่านค่า attribute ของ Object ชื่อ getXxx()
 - จะมี constructor หรือไม่มีก็ได้ ถ้ามี constructor นั้นจะต้องไม่มีพารามิเตอร์



JavaBeans

- JavaBeans คือ คลาสที่นิยามขึ้น เพื่อใช้ในการเก็บข้อมูลตามความต้องการของโปรแกรม มี
วัตถุประสงค์เพื่อห่อหุ้มข้อมูลเป็นก้อนเดียว ง่ายต่อการล้าเสี่ยง

ประกาศตัวแปรที่จะใช้
เก็บข้อมูล

Method สำหรับรับค่า
มาบรรจุใน JavaBeans
(เรียกว่า Constructor)

```
public class Product {  
  
    private int pid;  
    private String productName;  
    private String productDetail;  
    private int price;  
  
    public Product(int pid, String productName, String productDetail, int price) {  
        this.pid = pid;  
        this.productName = productName;  
        this.productDetail = productDetail;  
        this.price = price;  
    }  
  
    // Getter & Setter Method  
}
```

รายการตัวแปรที่รับค่ามา

Product

pid: int productName: string productDetail: string price: int
--



JavaBeans

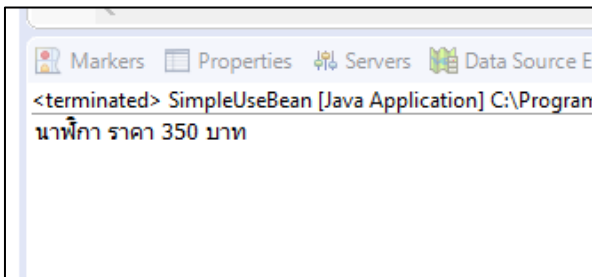
- เมื่อต้องการใช้ **JavaBeans** ในการเก็บข้อมูล จะต้องสร้าง object ใหม่

```
public class SimpleUseBean {  
    public static void main(String[] args) {  
        Product clock = new Product(15, "นาฬิกา", "ดูเวลา", 350);  
        System.out.println(clock.getProduct() + " ราคา "  
                           + clock.getPrice() + " บาท");  
    }  
}
```

สร้าง object และบรรจุข้อมูล

↓

ดึงข้อมูลออกมาแสดงผล

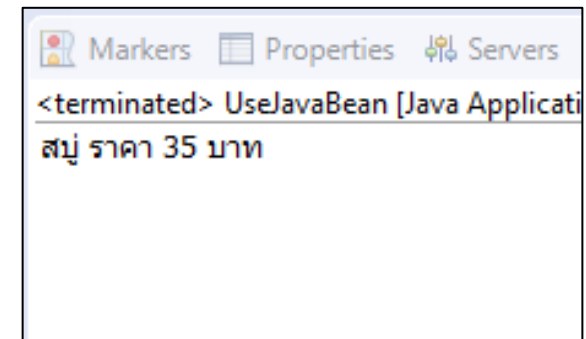




JavaBeans

- การลำเลียงข้อมูลด้วย JavaBeans

```
public class UseJavaBean1 {  
  
    // เมธอดที่ส่ง JavaBean กลับ  
    public static Product getProduct() {  
        Product product = new Product(12, "สนุ", "ใช้ทำความสะอาด", 35);  
        return product;  
    }  
  
    // เมธอดที่รับข้อมูลจาก JavaBean  
    public static void print(Product p) {  
        System.out.println(p.getProductName() + " ราคา " + p.getPrice() + " บาท");  
    }  
  
    public static void main(String[] args) {  
        Product p = getProduct();  
        print(p);  
    }  
}
```



- Array in JavaBeans

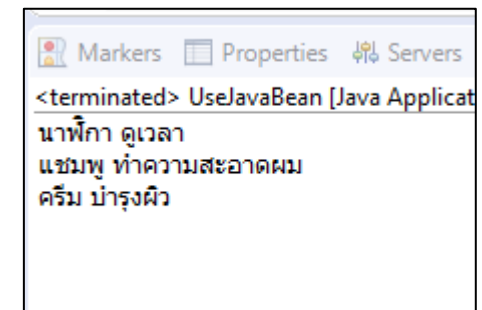
```
import java.util.ArrayList;

public class UseJavaBean2 {

    public static ArrayList<Product> getProductList() {
        Product clock = new Product(15, "นาฬิกา", "ดูเวลา", 350);
        Product shampoo = new Product(16, "แชมพู", "ทำความสะอาดผม", 135);
        Product cream = new Product(17, "ครีม", "บำรุงผิว", 69);

        ArrayList<Product> list = new ArrayList<Product>();
        list.add(clock);
        list.add(shampoo);
        list.add(cream);
        return list;
    }

    public static void main(String[] args) {
        ArrayList<Product> products = getProductList();
        for (Product product : products) {
            System.out.println(product.getProductDetail() );
        }
    }
}
```



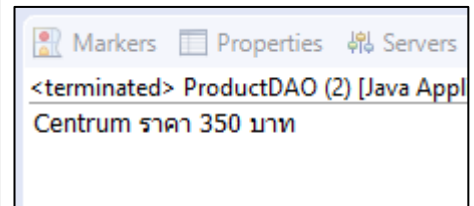
รูปแบบคลาส DAO

```
public class ProductDAO {  
  
    // ประกาศตัวแปรเพื่อเก็บสถานะการเชื่อมต่อสำหรับทุก method  
    private Connection con;  
  
    // constructor สำหรับเชื่อมต่อฐานข้อมูล  
    public ProductDAO() throws SQLException, ClassNotFoundException {  
        Class.forName("com.mysql.jdbc.Driver");  
        con = DriverManager.getConnection("jdbc:mysql://localhost/blueshop?characterEncoding=utf-8", "root", "");  
    }  
  
    // เมธอดปิดการเชื่อมต่อฐานข้อมูล  
    public void closeConnection() throws SQLException {  
        con.close();  
    }  
  
    // เมธอดดึงข้อมูลจาก ResultSet มาเก็บใน JavaBeans  
    private Product mappingProduct(ResultSet resultSet) throws SQLException {  
        Product product = new Product();  
        product.setPid( resultSet.getInt("pid") );  
        product.setProductName( resultSet.getString("pname") );  
        product.setProductDetail( resultSet.getString("pdetail") );  
        product.setPrice( resultSet.getInt("price") );  
        return product; // ส่งกลับเป็น javabean  
    }  
  
    // เมธอดที่ทำงานกับฐานข้อมูลอื่นๆ  
    public Product getProduct (int pid)                throws SQLException { ... }  
    public ArrayList<Product> getAllProduct()          throws SQLException { ... }  
    public void createProduct (Product product)        throws SQLException { ... }  
    public void updateProduct (Product product)        throws SQLException { ... }  
    public void deleteProduct (int pid)                throws SQLException { ... }  
}
```

รูปแบบคลาส DAO

- Method ที่ใช้ดึงข้อมูล 1 แถวตามคีย์หลัก

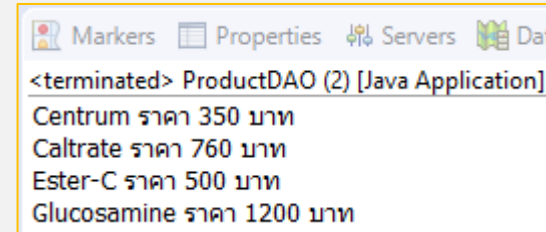
```
public Product getProduct(int pid) throws SQLException {  
  
    // เตรียมคำสั่ง SQL  
    PreparedStatement pStatement = con.prepareStatement("SELECT * FROM product WHERE pid = ?");  
    pStatement.setInt(1, pid);  
  
    // ส่งคำสั่ง SQL ไปยังฐานข้อมูล  
    ResultSet resultSet = pStatement.executeQuery();  
  
    if (resultSet.next()) { // ถ้าพบข้อมูล  
        Product product = mappingProduct(resultSet); // นำผลลัพธ์ที่ฐานข้อมูลส่งกลับแปลงเป็น object  
        return product; // ส่งกลับเป็น javabean  
    } else { // ถ้าไม่พบข้อมูล  
        return null;  
    }  
}  
  
public static void main(String[] args) {  
    try {  
        ProductDAO productDAO = new ProductDAO(); // สร้าง object DAO  
        Product p = productDAO.getProduct(1); // เรียกเมธอดจาก DAO  
        System.out.println(p.getName() + " ราคา " + p.getPrice() + " บาท"); // นำผลลัพธ์มาแสดง  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```



รูปแบบคลาส DAO

- Method ที่ใช้ดึงข้อมูลทุกแถว

```
public ArrayList<Product> getAllProduct() throws SQLException {  
  
    // เตรียมคำสั่ง SQL  
    PreparedStatement pStatement = con.prepareStatement("SELECT * FROM product");  
  
    // ส่งคำสั่ง SQL ไปยังฐานข้อมูล  
    ResultSet resultSet = pStatement.executeQuery();  
  
    // ประกาศอาร์เรย์สำหรับเก็บ javabeans  
    ArrayList<Product> productList = new ArrayList<Product>();  
  
    while (resultSet.next()) {  
        Product p = mappingProduct(resultSet); // แปลงข้อมูลเป็น javabean  
        productList.add(p); // เก็บ javabean ในอาร์เรย์  
    }  
    return productList; // ส่งอาร์เรย์กลับ  
}  
  
public static void main(String[] args) {  
    try {  
        ProductDAO productDAO = new ProductDAO(); // สร้าง object DAO  
        ArrayList<Product> products = productDAO.getAllProduct(); // เรียกเมธอดจาก DAO  
        for (Product p : products) {  
            System.out.println(p.getProductName() + " ราคา " + p.getPrice() + " บาท"); // นำผลลัพธ์มาแสดง  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```



<terminated> ProductDAO (2) [Java Application]
Centrum ราคา 350 บาท
Caltrate ราคา 760 บาท
Ester-C ราคา 500 บาท
Glucosamine ราคา 1200 บาท

รูปแบบคลาส DAO

- Method เพิ่มข้อมูล

```
public void createProduct (Product product) throws SQLException {  
    // เตรียมคำสั่ง SQL  
    PreparedStatement pStatement = con.prepareStatement("INSERT INTO product (pname, pdetail, price) VALUES (?, ?, ?)");  
    pStatement.setString(1, product.getProductName() );  
    pStatement.setString(2, product.getProductDetail() );  
    pStatement.setInt(3, product.getPrice() );  
  
    // ส่งคำสั่ง SQL ไปยังฐานข้อมูล  
    pStatement.executeUpdate();  
}  
  
public static void main(String[] args) {  
    try {  
        ProductDAO productDAO = new ProductDAO(); // สร้าง object DAO  
  
        // เตรียม product object ที่จะนำไปเพิ่มลงฐานข้อมูล (ส่ง pid=0 เมื่อเป็น auto increment)  
        Product product = new Product(0, "นาฬิกา", "ดูเวลา", 350);  
  
        // เรียกเมธอดจาก DAO  
        productDAO.createProduct(product);  
        productDAO.closeConnection(); // การเพิ่ม/ลบ/แก้ไข ต้องปิดการติดต่อฐานข้อมูลเสมอ  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

รูปแบบคลาส DAO

- Method แก้ไขข้อมูล

```
public void updateProduct (Product product) throws SQLException {
    // เตรียมคำสั่ง SQL
    PreparedStatement pStatement = con.prepareStatement("UPDATE product SET pname = ?, pdetail = ?,
                                                         price = ? WHERE pid = ?");

    pStatement.setString(1, product.getProductname() );
    pStatement.setString(2, product.getProductDetail() );
    pStatement.setInt(3, product.getPrice() );
    pStatement.setInt(4, product.getPid() );

    // ส่งคำสั่ง SQL ไปยังฐานข้อมูล
    pStatement.executeUpdate();
}

public static void main(String[] args) {
    try {
        ProductDAO productDAO = new ProductDAO(); // สร้าง object DAO

        // เตรียม product object ที่จะนำไปแก้ไขในฐานข้อมูล
        Product product = new Product(29, "นาฬิกาใหม่", "ดูเวลา + ดูอัตราเฟาผลาญ", 350);

        // เรียกเมธอดจาก DAO
        productDAO.updateProduct(product);
        productDAO.closeConnection();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```


รูปแบบคลาส DAO

- Method ลบข้อมูล

```
public void deleteProduct (int id) throws SQLException {  
    // เตรียมคำสั่ง SQL  
    PreparedStatement pStatement = con.prepareStatement("DELETE FROM product WHERE pid = ?");  
    pStatement.setInt(1, id);  
  
    // ส่งคำสั่ง SQL ไปยังฐานข้อมูล  
    pStatement.executeUpdate();  
}  
  
public static void main(String[] args) {  
  
    try {  
        ProductDAO productDAO = new ProductDAO(); // สร้าง object DAO  
  
        // เรียกเมธอดจาก DAO  
        productDAO.deleteProduct(20);  
        productDAO.closeConnection();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

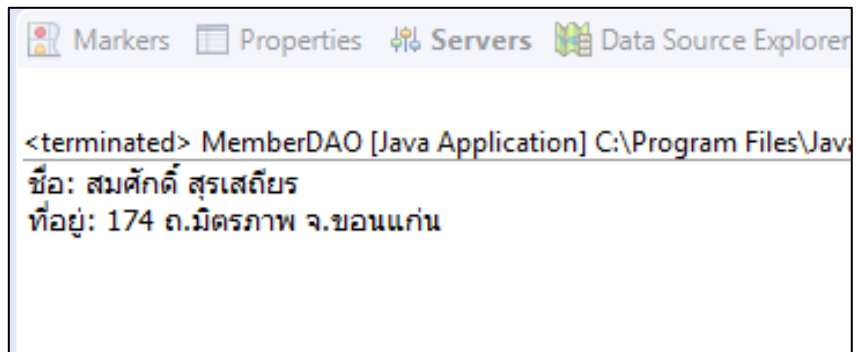
Assignment

- ให้สร้างคลาส JavaBeans ชื่อ Member โดยมีสมาชิกแสดงดังคลาสที่ให้

Member
username: string password: string name: string address: string mobile: string email: string

Assignment

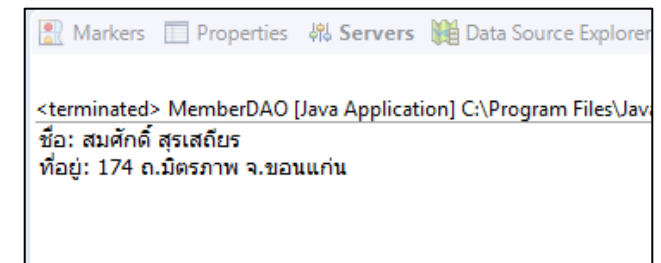
- สร้าง DAO สำหรับตาราง member ในฐานข้อมูล blueshop ซึ่งมีขั้นตอนดังนี้
 - ใช้คลาส JavaBeans ชื่อ **Member**
 - สร้าง DAO ชื่อ **MemberDAO**
 - สร้างเมธอดสำหรับดึงข้อมูลตาม username
 - **public Member getMember (String username) { }**
 - ทดสอบการทำงาน



Assignment

- สร้าง DAO สำหรับตาราง member ในฐานข้อมูล blueshop ซึ่งมีขั้นตอนดังนี้
 - ใช้คลาส JavaBeans ชื่อ **Member**
 - สร้าง DAO ชื่อ **MemberDAO**
 - สร้างเมธอดสำหรับดึงข้อมูลตาม username
 - `public Member getMember (String username) { }`
- ทดสอบการทำงาน

```
public static void main(String[] args) {  
    try {  
        MemberDAO memberDAO = new MemberDAO(); // สร้าง object DAO  
        Member member = memberDAO.getMember("somsak"); // เรียกเมธอดจาก DAO  
        System.out.println("ชื่อ: " + member.name);  
        System.out.println("ที่อยู่: " + member.address);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```



- สร้างคลาส MemberDAO ที่มีเมธอดให้บริการเกี่ยวกับฐานข้อมูล blueshop ตาราง member ซึ่งประกอบด้วย 5 เมธอด ดังนี้
 - Member **getMember (String username)** ใช้ดึงข้อมูลสมาชิกตาม username
 - ArrayList<Member> **getAllMember()** ใช้ดึงข้อมูลสมาชิกทั้งหมด
 - void **createMember (Member member)** ใช้เพิ่มข้อมูลสมาชิกในฐานข้อมูล
 - void **updateMember (Member member)** ใช้แก้ไขข้อมูลสมาชิก
 - void **deleteMember (String username)** ใช้ลบข้อมูลสมาชิก
- ทดสอบการทำงานโดยเขียน Console Application เรียกเมธอดทั้งหมดจากคลาส DAO โดยสร้าง Class main เพื่อใช้ในการทดสอบ

member
username (PK)
password
name
address
mobile
email