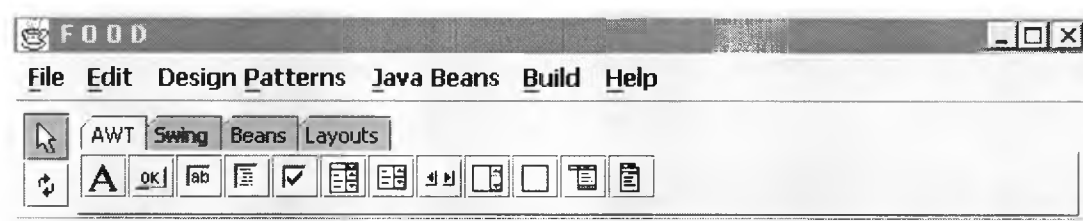


การใช้งานโครงสร้าง FOOD

ในบทนี้จะอธิบายถึงส่วนติดต่อกับผู้ใช้งานของสภาพแวดล้อมในการพัฒนาโปรแกรมประยุกต์เชิงวัตถุด้วย FOOD และวิธีการใช้ FOOD เพื่อพัฒนาโปรแกรมประยุกต์เชิงวัตถุ ซึ่งในการใช้งานแบ่งเป็น 3 ขั้นตอนหลัก คือ ขั้นตอนการสร้างโปรแกรมประยุกต์ ขั้นตอนการสร้างหรือนำเข้าจาวาบี๋น และขั้นตอนการสร้างโครงชุดคำสั่งรูปแบบการออกแบบ


5.1 ขั้นตอนการสร้างโปรแกรมประยุกต์


เมื่อผู้ใช้งานเข้าสู่ระบบ หน้าจอหลัก (Main window) จะเป็นตัวควบคุมสภาพแวดล้อมในการทำงานของระบบ ส่วนของหน้าจอหลักจะแบ่งออกเป็น 2 ส่วน คือ ส่วนของแถบเมนู และแถบองค์ประกอบของซอฟต์แวร์ ที่จะนำมาใช้พัฒนาโปรแกรมในส่วนติดต่อกับผู้ใช้งาน ดังแสดงในรูปที่ 5.1



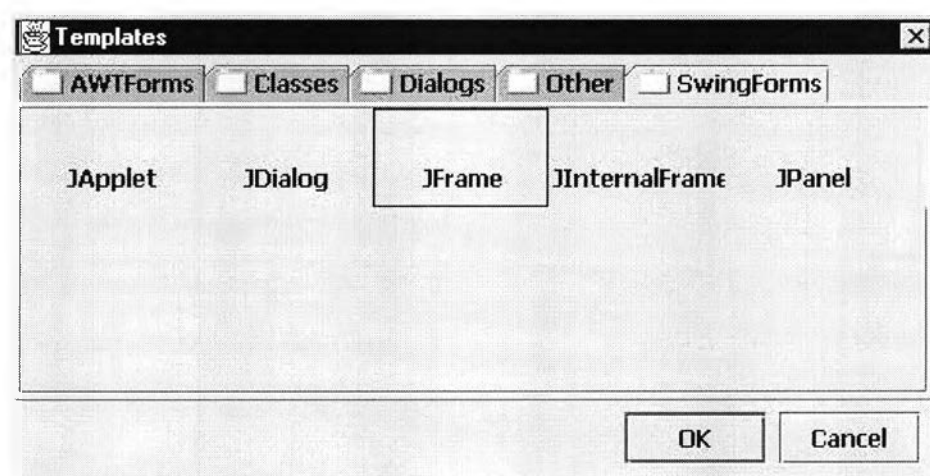
รูปที่ 5.1 หน้าจอหลักของ FOOD ประกอบด้วยเมนูและแถบเครื่องมือ

ส่วนของแถบเมนู และแถบเครื่องมือ ประกอบด้วยรายการต่างๆ ที่ให้ผู้ใช้สามารถเลือกการทำงานได้ ดังต่อไปนี้

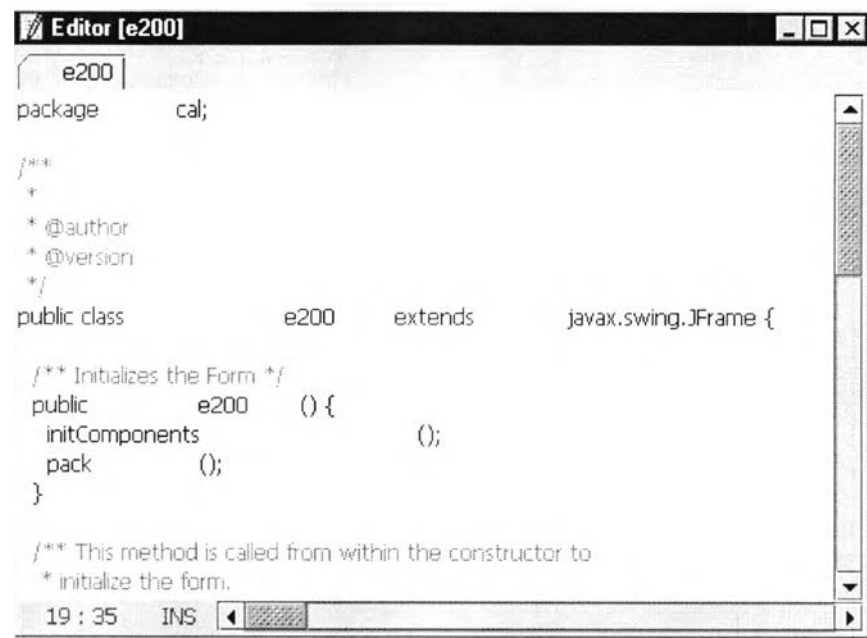
1. “File Menu” จะมีรายการเมนูย่อยเพื่อให้ผู้ใช้เลือกสร้างจาวาออปเจกใหม่ บันทึกไฟล์ จาวาซอร์สโปรแกรม และออกจากระบบ
2. “Design Patterns Menu” เพื่อเรียกใช้งาน ส่วนจัดการรูปแบบการออกแบบ
3. “Java Beans Menu” เพื่อเรียกใช้งาน ส่วนจัดการจาวาบี๋น
4. “Build Menu” เพื่อให้ผู้ใช้ทำการคอมไพล์โปรแกรม และสั่งรัน โปรแกรมที่พัฒนา
5. “Help Menu” เป็นการแสดงข้อมูลเพิ่มเติมให้ผู้ใช้สามารถศึกษาทำความเข้าใจการใช้งาน
6.  เป็นปุ่มกำหนดสถานะออกแบบ เพื่อใช้เมาส์ควบคุมการกำหนดตำแหน่ง และขนาดองค์ประกอบของซอฟต์แวร์บนฟอร์มวินโดว์

7.  เป็นการกำหนดสถานะเชื่อมองค์ประกอบ เพื่อให้เมาส์ควบคุมการเชื่อมความสัมพันธ์ขององค์ประกอบของซอฟต์แวร์บนฟอร์มวินโดว์
8. แพคเกจ “AWT” ประกอบด้วยปุมสัญลักษณ์องค์ประกอบของซอฟต์แวร์ในแพ็คเกจ java.awt
9. แพคเกจ “Swing” ประกอบด้วยปุมองค์ประกอบของซอฟต์แวร์ในแพ็คเกจ javax.swing
10. แพคเกจ “Beans” ประกอบด้วยปุมองค์ประกอบของจาวาเบิน
11. แพคเกจ “Layouts” ประกอบด้วยปุมสัญลักษณ์ ของรูปแบบการจัดวางองค์ประกอบอัตโนมัติที่ใช้ในจาวา

เมื่อผู้ใช้เลือกเริ่มงานใหม่จากแถบเมนู (File/New) จะมีวินโดว์แสดงเทมเพลต (Templates) ให้ผู้ใช้เลือกประเภทของออปเจกต์ที่ต้องการสร้างใหม่ ดังแสดงในรูปที่ 5.2 และเมื่อผู้ใช้เลือกประเภทของออปเจกต์แล้ว โปรแกรมจะให้ผู้ตั้งชื่อ เพื่อจะใช้บันทึกเป็นชื่อของไฟล์ชุดคำสั่ง จากนั้นระบบจะทำการเตรียมสภาพแวดล้อมสำหรับพัฒนาโปรแกรมแบ่งเป็น 3 วินโดว์ วินโดว์แรกเป็นวินโดว์สำหรับแก้ไขชุดคำสั่งดังแสดงในรูปที่ 5.3 โดยระบบจะสร้างโครงชุดคำสั่งเริ่มต้นให้อัตโนมัติตามประเภทที่เลือกจากเทมเพลต วินโดว์ที่สองเป็นฟอร์มวินโดว์จะแสดงเป็นวินโดว์ว่างไว้สำหรับ ผู้พัฒนาโปรแกรมใช้ออกแบบส่วนติดต่อกับผู้ใช้งานโดยนำองค์ประกอบของซอฟต์แวร์มาวางจัดเรียงบนฟอร์มวินโดว์ ดังแสดงในรูปที่ 5.4 และวินโดว์ที่สามมีรูปแบบเป็นโครงสร้างต้นไม้แสดงรายการ ออปเจกต์ทั้งหมดที่สร้างขึ้นระหว่างการพัฒนาโปรแกรมและแสดงคุณสมบัติต่างๆของออปเจกต์ที่ถูกเลือก โดยผู้ใช้สามารถเปลี่ยนแปลงค่าของคุณสมบัติได้ตามต้องการ ดังแสดงในรูปที่ 5.5



รูปที่ 5.2 เทมเพลตให้ผู้ใช้เลือกประเภทของออปเจกต์ที่ต้องการสร้างใหม่



```

Editor [e200]
e200
package    cal;

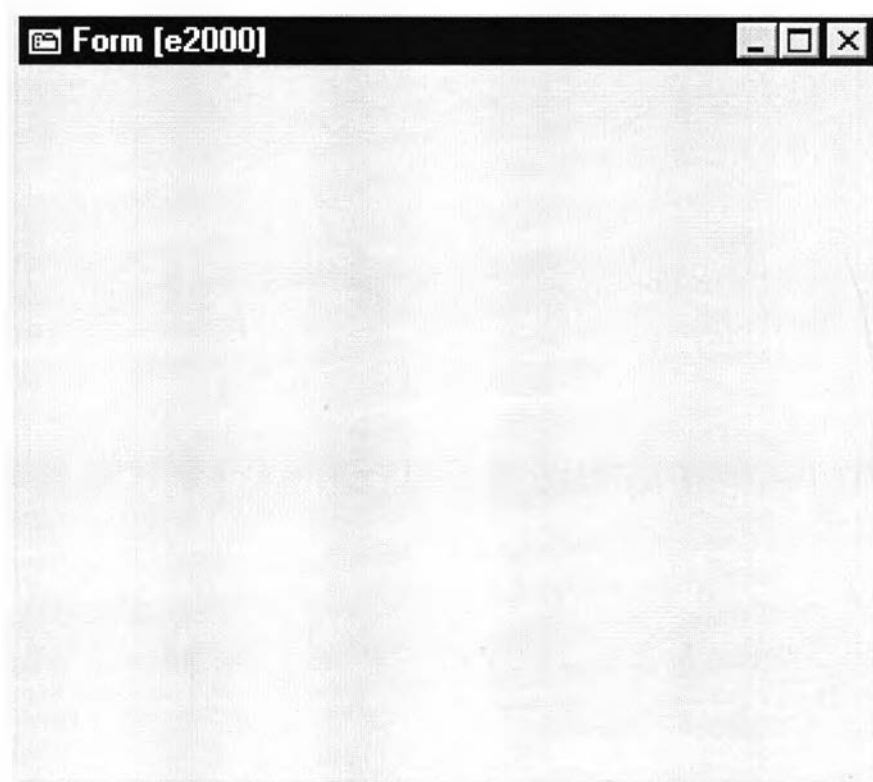
/**
 *
 * @author
 * @version
 */
public class    e200    extends    javax.swing.JFrame {

    /** Initializes the Form */
    public    e200    () {
        initComponents    ();
        pack    ();
    }

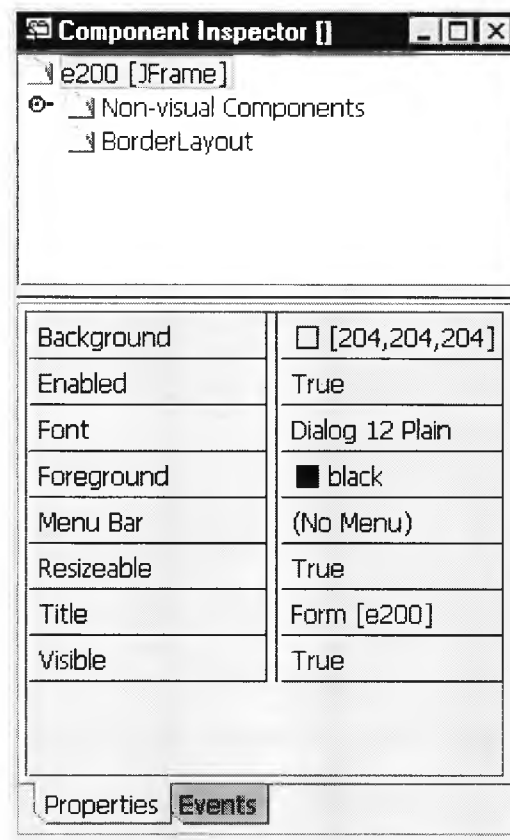
    /** This method is called from within the constructor to
     * initialize the form.
    */
19 : 35    INS

```

รูปที่ 5.3 วินโดว์สำหรับแก้ไขชุดคำสั่ง

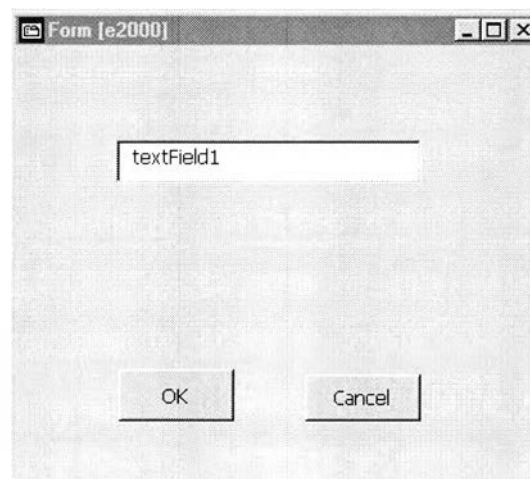


รูปที่ 5.4 ฟอรั่มวินโดว์ แสดงวินโดว์ว่างให้ผู้ใช้งานองค์ประกอบมาวางจัดเรียงเพื่อ
สร้างส่วนติดต่อกับผู้ใช้




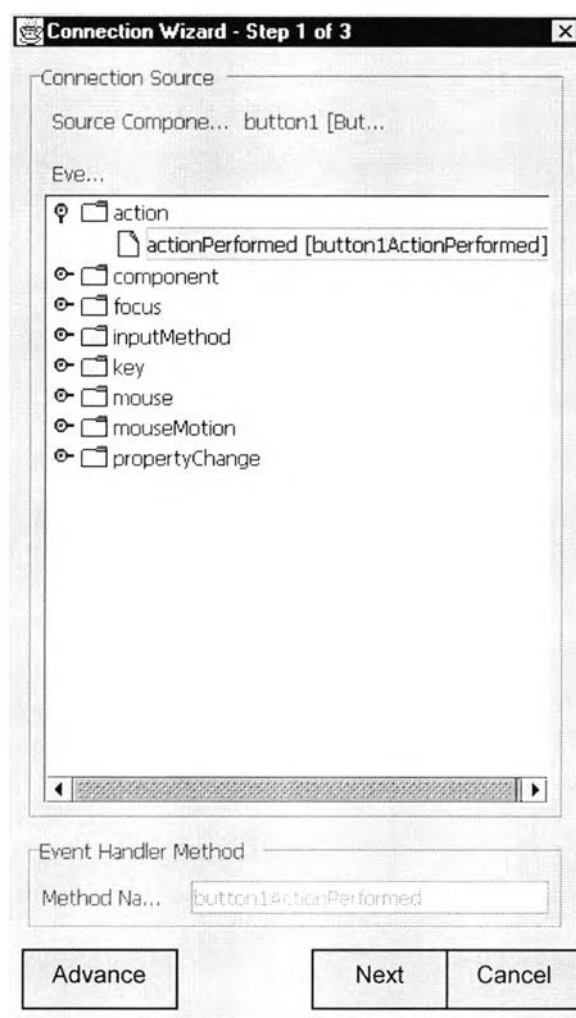
รูปที่ 5.5 วินโดว์แสดงรายการออปเจกต์ที่ถูกสร้างและคุณสมบัติของออปเจกต์ที่ถูกเลือก

จากนั้นผู้ใช้สามารถใช้เมาส์คลิกเลือกองค์ประกอบของซอฟต์แวร์ จากแถบเครื่องมือมาวางลงบนฟอร์มวินโดว์รายละเอียด ดังแสดงในรูปที่ 5.6 โดยองค์ประกอบแต่ละตัวสามารถกำหนดคุณสมบัติได้จากวินโดว์แสดงคุณสมบัติ และระบบจะทำการสร้างโครงสร้างคำสั่งขององค์ประกอบที่สร้างใหม่ให้อัตโนมัติแทรกในโครงสร้างคำสั่งเดิมที่แสดงบนวินโดว์สำหรับแก้ไขชุดคำสั่ง



รูปที่ 5.6 การนำองค์ประกอบของซอฟต์แวร์มาวางในฟอร์มวินโดว์เพื่อสร้างส่วนติดต่อกับผู้ใช้

ผู้ใช้สามารถเขียนชุดคำสั่งเพิ่มเติมให้สมบูรณ์ได้เองในวินโดว์สำหรับแก้ไขชุดคำสั่ง หรืออีกวิธีหนึ่งเพื่อช่วยลดการเขียนชุดคำสั่งโดยการเลือกให้โปรแกรมเชื่อมโยงความสัมพันธ์ระหว่างองค์ประกอบ 2 ตัวโดยใช้เมาส์คลิกปุ่ม  จากแถบเมนู จากนั้นให้ใช้เมาส์คลิกเลือกองค์ประกอบแรกเป็นต้นทาง (Source component) แล้วให้คลิกองค์ประกอบที่สองเป็น ปลายทาง (Target component) ระบบจะแสดงรายการเหตุการณ์ในวินโดว์ลำดับขั้นที่ 1 ให้ผู้ใช้เลือกเหตุการณ์ ที่จะให้ องค์ประกอบต้นทางเริ่มทำงาน ดังแสดงในรูปที่ 5.7 จากนั้นระบบจะให้ผู้ใช้เลือกสิ่งที่องค์ประกอบ ปลายทางถูกกระทำ อย่างใดอย่างหนึ่ง คือ เปลี่ยนแปลงคุณสมบัติ (Set property) เรียกใช้เมทอด (Method call) หรือ ผู้ใช้เขียนชุดคำสั่งเอง (User code) จากวินโดว์ลำดับขั้นที่ 2 ดังแสดงในรูปที่ 5.8 จากนั้นในวินโดว์ลำดับขั้นที่ 3 ระบบจะให้ผู้ใช้ป้อนค่าพารามิเตอร์ที่จะส่งค่าคุณสมบัติเมทอด หรือชุดคำสั่งขึ้นอยู่กับทางเลือกจากลำดับขั้นที่ 2 ดังแสดงในรูปที่ 5.9 สุดท้ายระบบจะทำการสร้างชุดคำสั่งให้อัตโนมัติในวินโดว์สำหรับแก้ไขชุดคำสั่ง ซึ่งผู้ใช้ก็สามารถเพิ่มเติมชุดคำสั่งได้อย่างอิสระ โดยผู้ใช้สามารถใช้รูปแบบการออกแบบช่วยในการเขียนชุดคำสั่ง และทำการคอมไพล์ชุดคำสั่งจาวาที่ได้ต่อไป



รูปที่ 5.7 วินโดว์ลำดับขั้นที่ 1 ให้ผู้ใช้เลือกเหตุการณ์ที่จะให้องค์ประกอบต้นทางเริ่มทำงาน



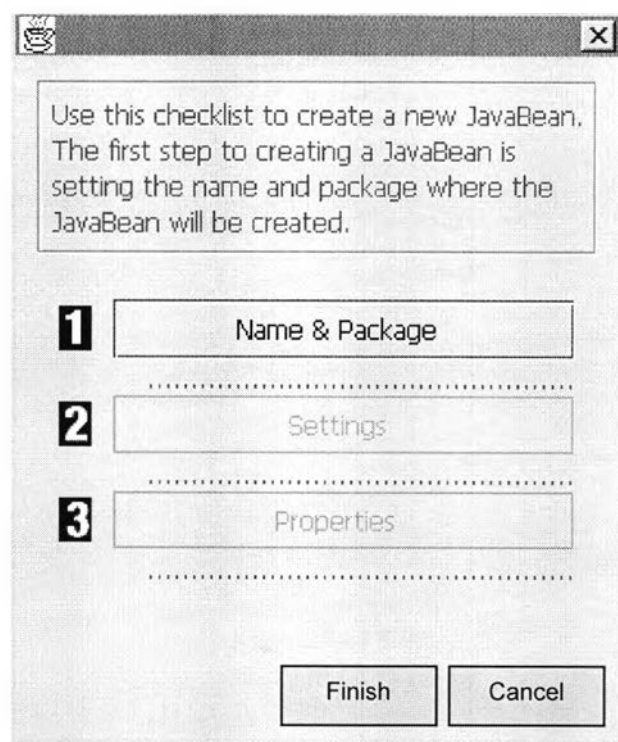
รูปที่ 5.8 วินโดว์ลำดับขั้นที่ 2 ให้ผู้ใช้เลือกสิ่งที่จะต้องประกอบปลายทางถูกกระทำ



รูปที่ 5.9 วินโดว์ลำดับขั้นที่ 3 ให้ผู้ใช้ป้อนค่าพารามิเตอร์ที่จะส่งค่าคุณสมบัติเมทอดหรือชุดคำสั่ง

5.2 ขั้นตอนการสร้างหรือนำเข้าจาวาบี๋น

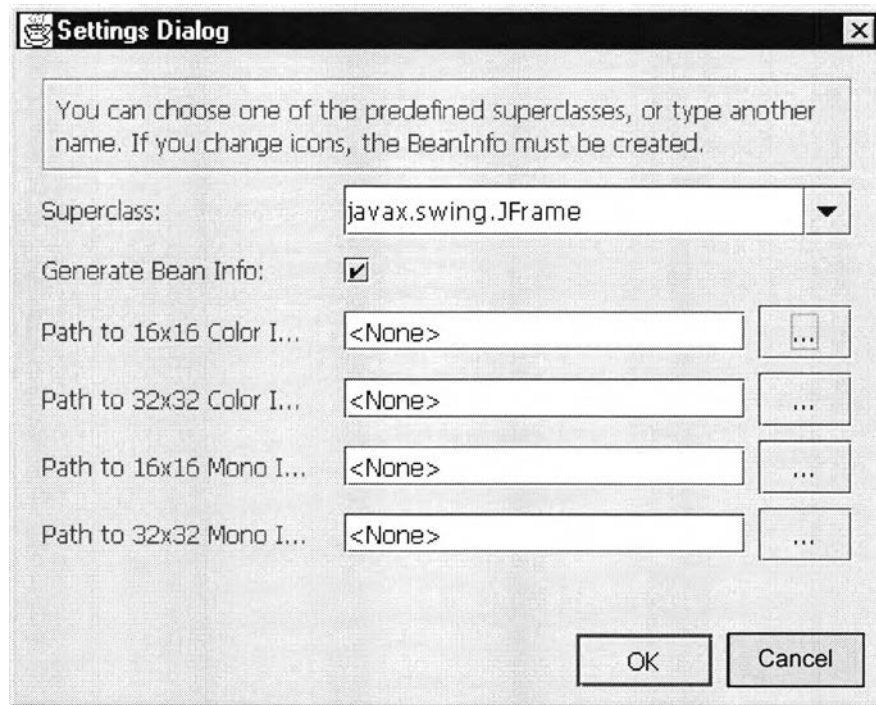
เมื่อผู้ใช้เลือกเมนู “Java Beans\Bean Manager” จะเป็นการเรียกใช้งาน ส่วนสำหรับจัดการจาวาบี๋น ซึ่งจะแบ่งลำดับออกเป็น 3 ขั้นตอน ในการสร้างจาวาบี๋น ดังแสดงในรูปที่ 5.10 ขั้นตอนแรกระบบจะให้ผู้ใช้กำหนดชื่อของจาวาบี๋นที่สร้าง และไคเรคทอรีที่จะจัดเก็บจาวาบี๋น ดังแสดงในรูปที่ 5.11 ลำดับขั้นที่สอง ระบบจะให้ผู้ใช้กำหนดซูเปอร์คลาส (Super Class) ที่จะรับถ่ายทอดคุณสมบัติ และกำหนดภาพสัญลักษณ์ของจาวาบี๋นที่ใช้แสดงบนแถบเครื่องมือเมื่อนำเข้าสู่ระบบ ดังแสดงในรูปที่ 5.12 และลำดับขั้นที่ 3 ระบบจะให้ผู้ใช้กำหนดสร้างคุณสมบัติ ให้แก่จาวาบี๋น ดังแสดงในรูปที่ 5.13 เมื่อผู้ใช้กดปุ่ม ระบบจะสร้างชุดคำสั่งโดยอัตโนมัติในวินโดว์สำหรับแก้ไขชุดคำสั่ง ดังแสดงในรูปที่ 5.14 และ ผู้ใช้สามารถเพิ่มเติมชุดคำสั่ง ได้อิสระ



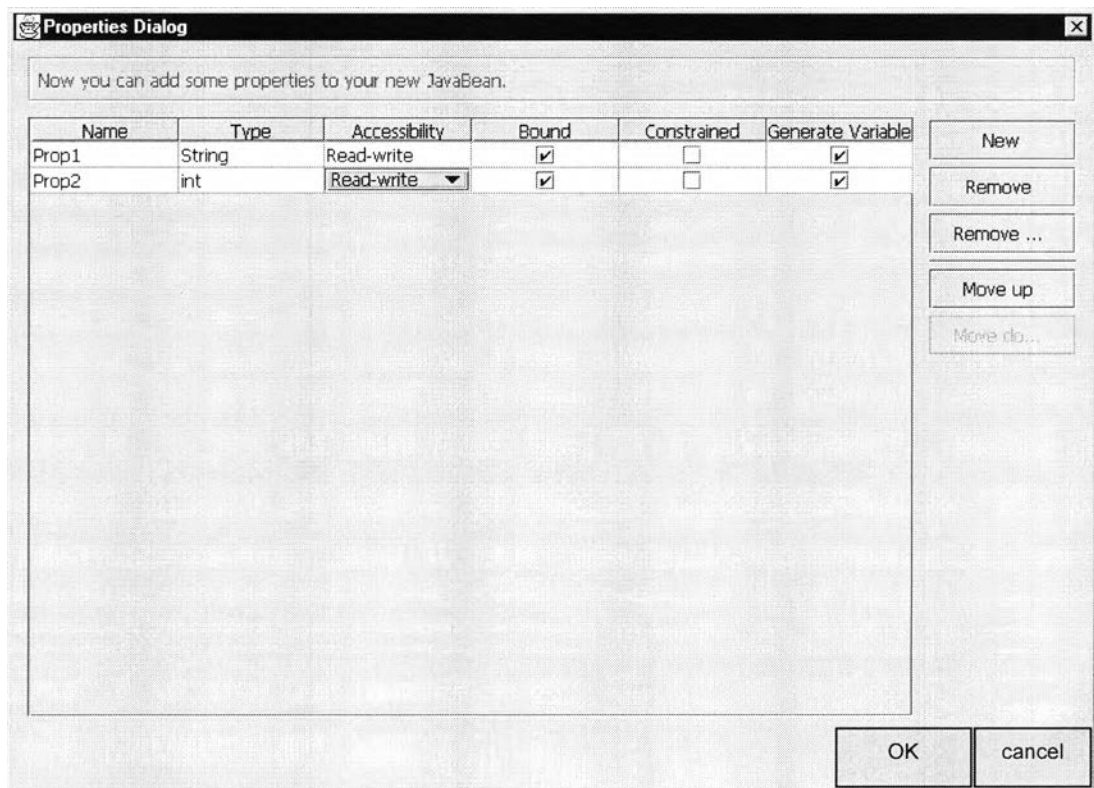
รูปที่ 5.10 หน้าจอหลักส่วนจัดการจาวาบี๋น



รูปที่ 5.11 ส่วนจัดการจาวาบี๋นขั้นตอนที่ 1 ตั้งชื่อจาวาบี๋น



รูปที่ 5.12 ส่วนจัดการจาวาเบินขั้นตอนที่ 2 เลือกซูเปอร์คลาสและกำหนดภาพสัญลักษณ์



รูปที่ 5.13 ส่วนจัดการจาวาเบินขั้นตอนที่ 3 กำหนดรายการคุณสมบัติให้จาวาเบิน

```

Editor [MyBeanBeanInfo]
e2000 * MyBean MyBeanBeanInfo
package      cal;

import      java.awt.Image;
import      java.beans.*;

/**
 * A BeanInfo class for the MyBean JavaBean.
 *
 * @version 1.00
 */
public class      MyBeanBeanInfo      extends      SimpleBeanInfo

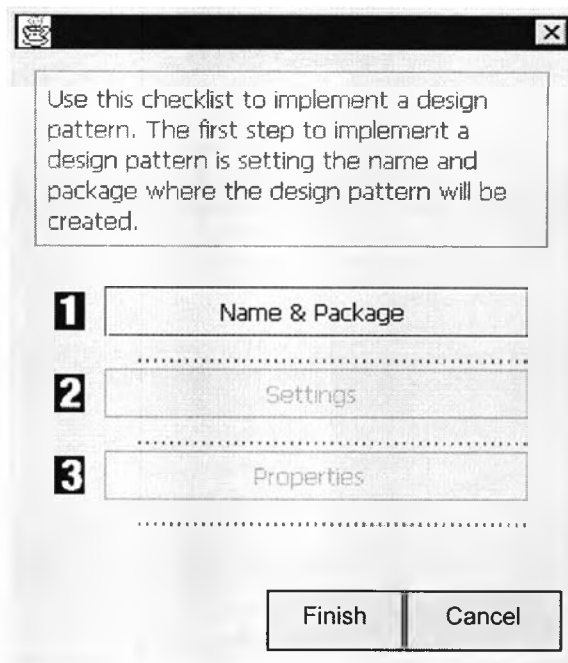
/**
 * Constructs a new BeanInfo class for the MyBean JavaBean.
 */
public      MyBeanBeanInfo      () {
}
1 : 1  INS

```

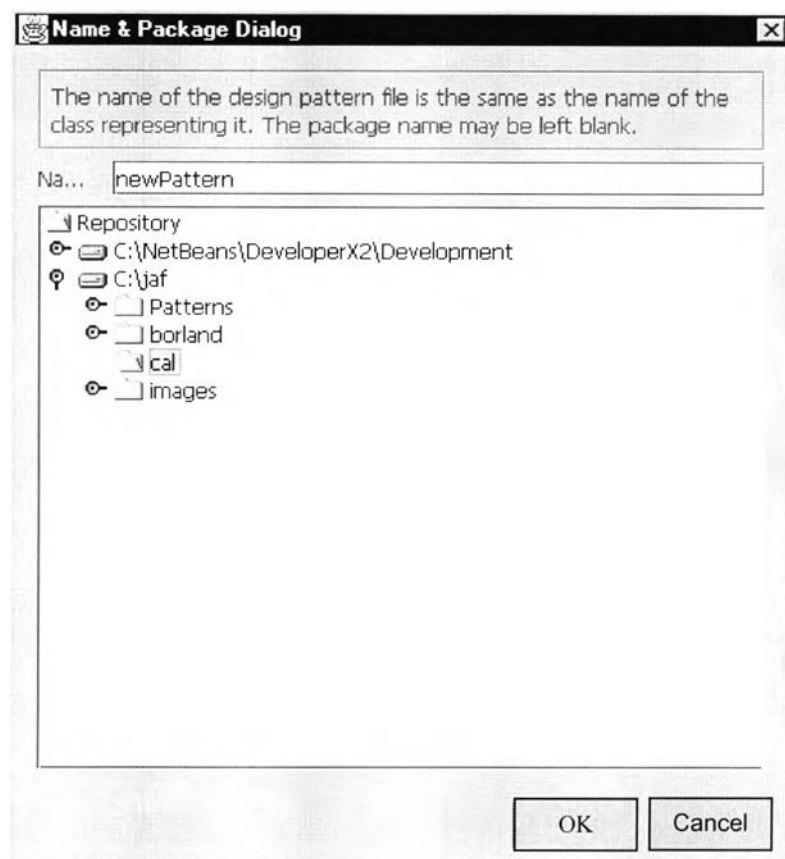
รูปที่ 5.14 ชุดคำสั่งที่ได้จากการสร้างจาวาบีน

5.3 ขั้นตอนการสร้างโครงชุดคำสั่งรูปแบบการออกแบบ

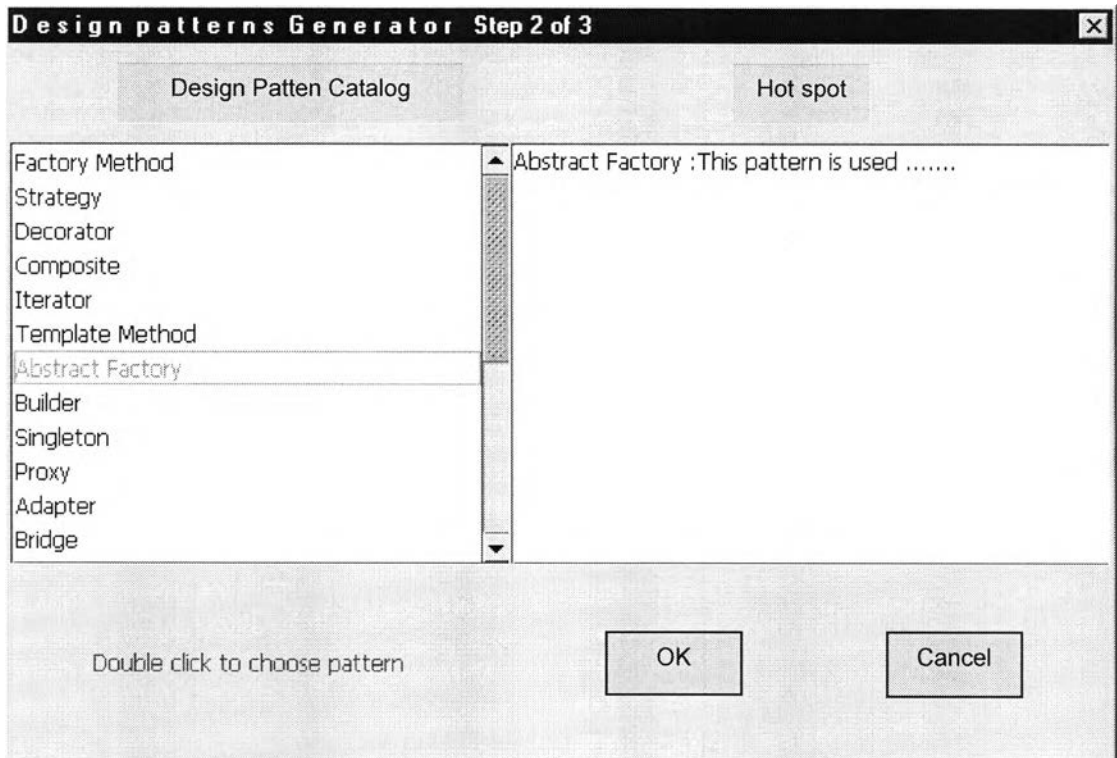
เมื่อผู้ใช้เลือกเมนู “Design Pattern\Pattern Manager” จะเป็นการเรียกใช้งาน ส่วนจัดการรูปแบบการออกแบบ ซึ่งจะแบ่งลำดับออกเป็น 3 ขั้นตอน ในการสร้างไฟล์ชุดคำสั่งของรูปแบบการออกแบบ ดังแสดงในรูปที่ 5.15 ขั้นตอนแรกระบบจะให้ผู้ใช้กำหนดชื่อไฟล์ของรูปแบบการออกแบบที่สร้าง และไดเรกทอรีที่จะจัดเก็บ ดังแสดงในรูปที่ 5.16 ขั้นตอนที่สอง ระบบจะให้ผู้ใช้เลือกรูปแบบการออกแบบที่เหมาะสมจาก 23 รายการ ดังแสดงในรูปที่ 5.17 และขั้นตอนที่สาม ระบบจะให้ผู้ใช้กำหนดชื่อคลาส ชื่อเมธอด พารามิเตอร์ต่างๆ ตามแต่ละรูปแบบที่ต้องการเพื่อนำไปสร้างเป็นโครงชุดคำสั่งรูปแบบการออกแบบ ดังแสดงในรูปที่ 5.18 ถ้าผู้ใช้ไม่เข้าใจวิธีการกำหนดค่าดังกล่าวสามารถดูรูปแผนภาพคลาสของรูปแบบการออกแบบเพื่อเป็นตัวอย่างในการกำหนดค่าตัวอย่าง เช่น เลือกรูปแบบการออกแบบที่เรียกว่า “Abstract Factory Pattern” โดยการกดปุ่ม **view** จะแสดงแผนภาพคลาสของรูปแบบการออกแบบที่เลือก ดังแสดงในรูปที่ 5.19 เมื่อผู้ใช้กดปุ่ม **Ok** ระบบจะสร้างชุดคำสั่งโดยอัตโนมัติในวินโดว์สำหรับแก้ไขชุดคำสั่ง ดังแสดงในรูปที่ 5.20 ซึ่งผู้ใช้สามารถเพิ่มเติมชุดคำสั่งได้อิสระ โดยนำไฟล์โครงชุดคำสั่งรูปแบบการออกแบบที่ได้ไปใช้เป็นแบบในการสร้างโปรแกรมประยุกต์เชิงวัตถุ



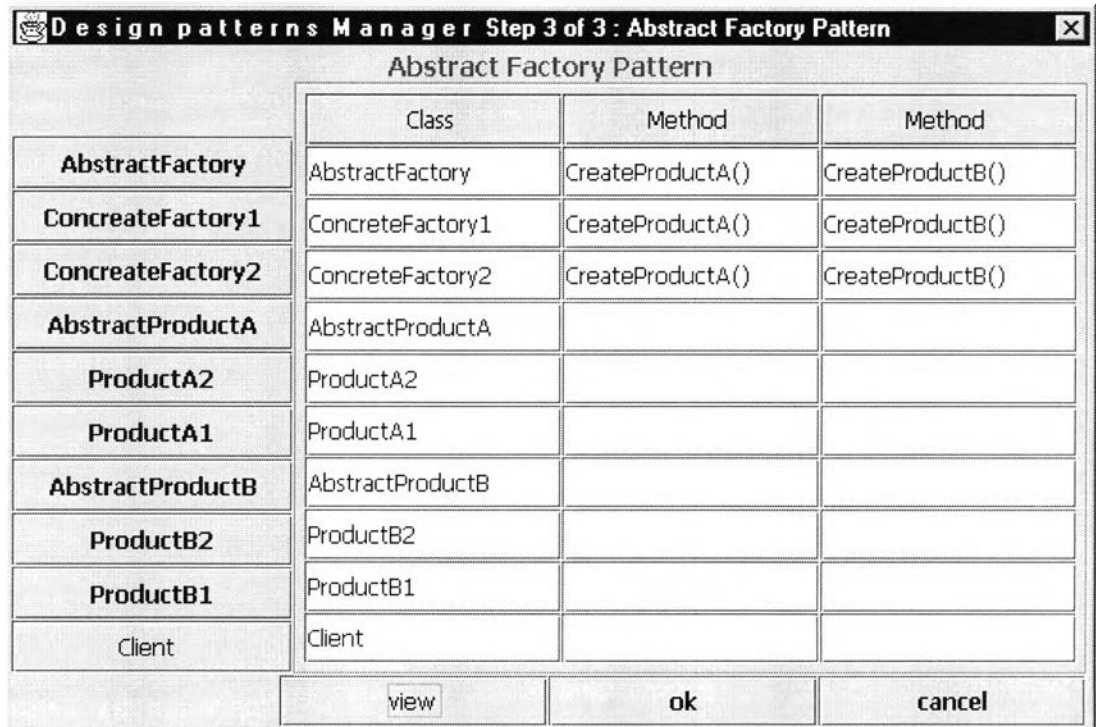
รูปที่ 5.15 หน้าจอหลักส่วนจัดการรูปแบบการออกแบบ



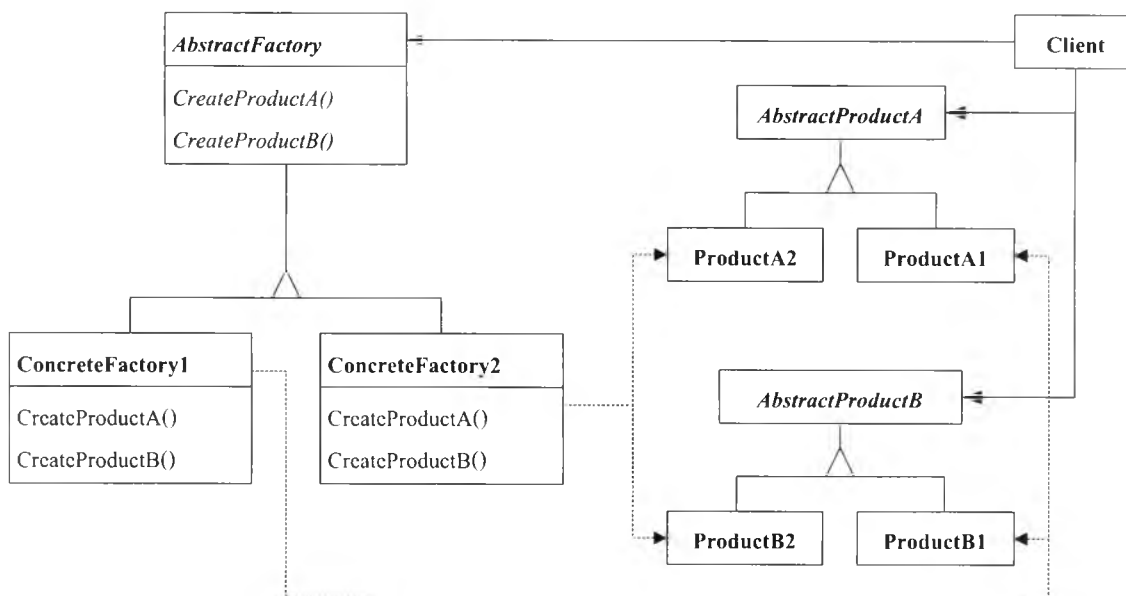
รูปที่ 5.16 ส่วนจัดการรูปแบบการออกแบบขั้นตอนที่ 1 กำหนดชื่อไฟล์ที่จะจัดเก็บ



รูปที่ 5.17 ส่วนจัดการรูปแบบการออกแบบขั้นตอนที่ 2 เลือกรูปแบบการออกแบบ



รูปที่ 5.18 ส่วนจัดการรูปแบบการออกแบบขั้นตอนที่ 3 กำหนดค่าเพื่อนำไปสร้างโครงสร้างชุดคำสั่งรูปแบบการออกแบบ



รูปที่ 5.19 แผนภาพคลาสรูปแบบการออกแบบ Abstract Factory Pattern เมื่อผู้ใช้เรียกดูเพื่อเป็นตัวอย่างในการกำหนดค่า

```

abstract class Product {
}

class ConcreteProduct extends Product {
}

abstract class Creator {
    abstract void FactoryMethod();
    abstract void AnOperation();
}

class ConcreteCreator extends Creator {
    /* It Creates Instance of ConcreteProduct Class */
    void FactoryMethod() {
        return new ConcreteProduct();
    }
}
  
```

รูปที่ 5.20 โครงสร้างคำสั่งรูปแบบการออกแบบที่ได้