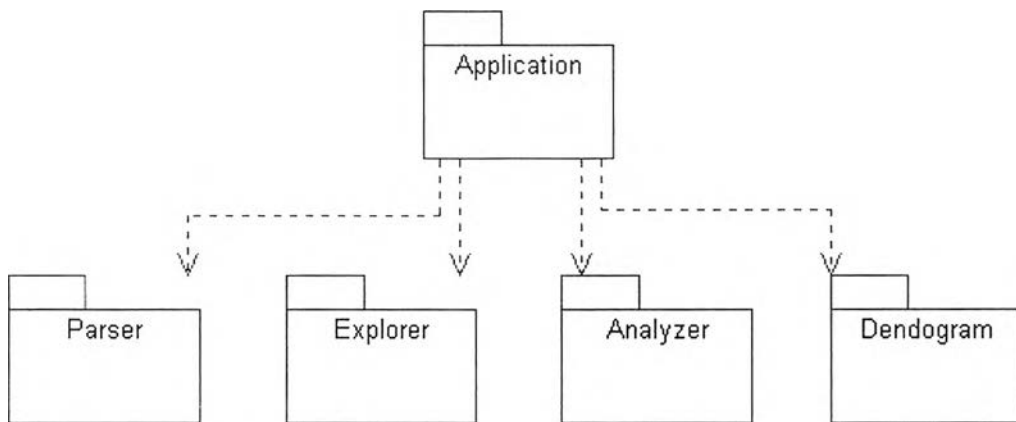


บทที่ 4

การออกแบบเครื่องมือระบุวัตถุซอฟต์แวร์ที่เป็นไปได้ ซึ่งประยุกต์ใช้วิธีการจัดกลุ่มข้อมูลแบบลำดับชั้นและการให้ค่าน้ำหนัก

เนื้อหาในบทนี้เกี่ยวข้องกับการออกแบบและเครื่องมือระบุวัตถุซอฟต์แวร์ที่เป็นไปได้ ซึ่งประยุกต์ใช้วิธีการจัดกลุ่มข้อมูลแบบลำดับชั้นและการให้ค่าน้ำหนัก โดยได้แบ่งการอธิบายเนื้อหาออกเป็นสองส่วน ได้แก่ (1) โครงสร้างของเครื่องมือ และ (2) การพัฒนาเครื่องมือ ซึ่งมีรายละเอียดดังต่อไปนี้

4.1 โครงสร้างของเครื่องมือ

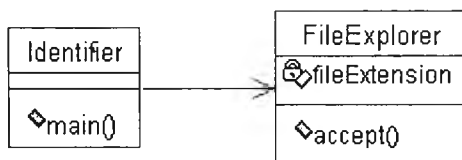


รูปที่ 4.1 แพคเกจไดอะแกรมของเครื่องมือระบุวัตถุซอฟต์แวร์ที่เป็นไปได้

การออกแบบเครื่องมือระบุวัตถุซอฟต์แวร์ที่เป็นไปได้ ทำตามหลักการออกแบบและเขียนโปรแกรมเชิงวัตถุ โดยแบ่งการทำงานของโปรแกรมออกเป็น ส่วน 5 ส่วน การทำงานแต่ละส่วนรับผิดชอบโดยกลุ่มคลาส (Package) ที่มีความเกี่ยวข้องกันจำนวน 5 กลุ่ม ดังแสดงในรูปที่ 4.1 กลุ่มคลาสแต่ละกลุ่มมีหน้าที่ต่างๆ กันดังนี้

- 1) กลุ่มตัวโปรแกรม (Application Package) ทำหน้าที่เป็นส่วนต่อประสานผู้ใช้ (User Interface) เพื่อรับคำสั่งและค่าต่างๆ เช่น ตำแหน่งของไฟล์โปรแกรมต้นฉบับ ค่าน้ำหนัก ฯลฯ
- 2) กลุ่มตัวแจง (Parser Package) ทำหน้าที่เกี่ยวกับการจัดการโปรแกรมต้นฉบับ เช่น การกำจัดคอมเมนต์ การคัดแยกส่วนประกอบเดิม การบันทึกส่วนประกอบเดิมลงในไฟล์ ฯลฯ
- 3) กลุ่มตัวค้นหา (Explorer Package) ทำหน้าที่ตรวจสอบความสัมพันธ์ระหว่างส่วนประกอบเดิม และบันทึกผลลงในตารางความสัมพันธ์
- 4) กลุ่มตัววิเคราะห์จัดกลุ่ม (Analyzer Package) ทำหน้าที่สร้างเมตริกซ์ข้อมูล คำนวณค่าระยะทาง สร้างเมตริกซ์ความแตกต่าง และจัดกลุ่มส่วนคำสั่ง
- 5) กลุ่มเดนโดแกรม (Dendogram) ทำหน้าที่สร้างเดนโดแกรม

4.1.1 กลุ่มตัวโปรแกรม



รูปที่ 4.2 คลาสภายในกลุ่มตัวโปรแกรม

กลุ่มตัวโปรแกรมประกอบด้วยคลาสจำนวน 2 คลาส ดังแสดงในรูปที่ 4.2 ซึ่งมีรายละเอียดดังนี้

- 1) คลาสไอดีเดนติไฟเออร์ (Class Identifier) ทำหน้าที่เป็นคลาสโปรแกรม (Application Class) หรือคลาสที่ถูกเรียกโดยจาวาอินเทอร์พรีเตอร์ (Java Interpreter) เพื่อเริ่มทำงานของโปรแกรม นอกจากนี้คลาสไอดีเดนติไฟเออร์ยังทำหน้าที่เป็นส่วนต่อประสานผู้ใช้โปรแกรมอีกด้วย
- 2) คลาสไฟล์เอ็กซ์พลอเรอร์ (Class FileExplorer) ทำหน้าที่กรองชื่อไฟล์โปรแกรมต้นฉบับที่แสดงในโปรแกรม

4.1.2 กลุ่มตัวแฉง

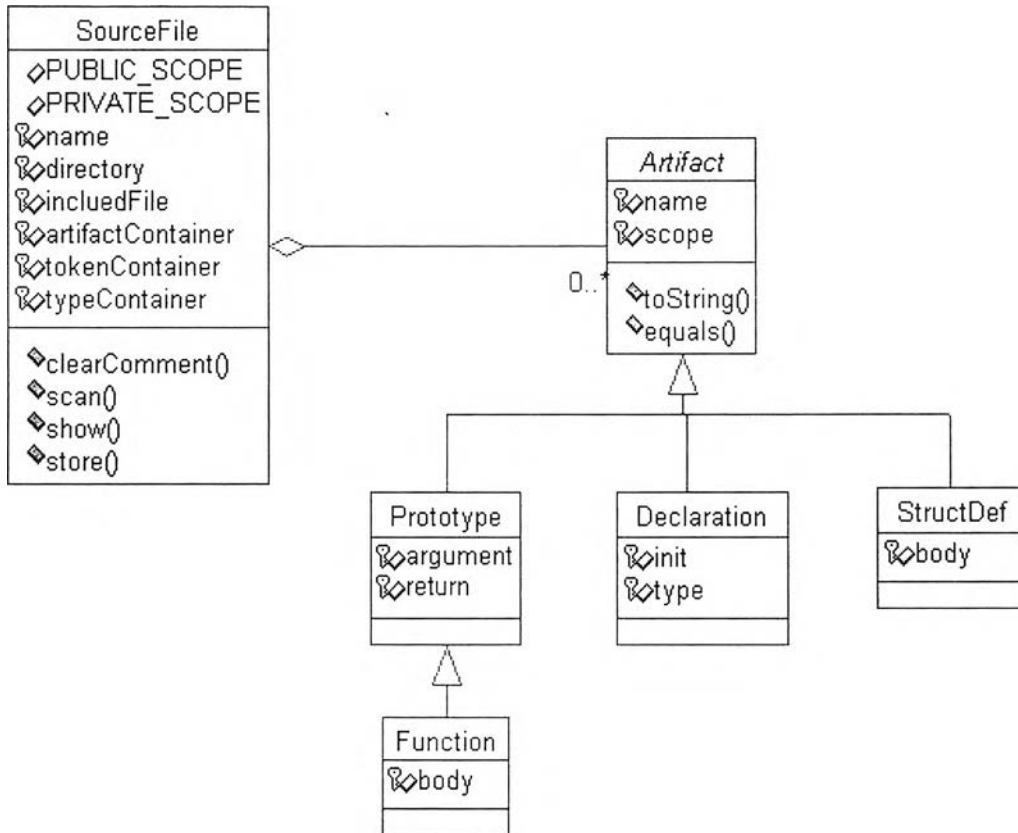
กลุ่มตัวแฉงประกอบด้วยคลาส 6 คลาส ดังแสดงในรูปที่ 4.3 ซึ่งมีรายละเอียดดังนี้

- 1) คลาสซอร์สไฟล์ (Class SourceFile) ทำหน้าที่เป็นตัวแทนโปรแกรมต้นฉบับ มีหน้าที่เกี่ยวกับการจัดการโปรแกรมต้นฉบับ เช่น กำจัดคอมเมนต์ ค้นหาส่วนประกอบเดิม ฯลฯ
- 2) คลาสอาร์ทิแฟกต์ (Class Artifact) ทำหน้าที่เป็นคลาสต้นแบบ (Abstract Class) เพื่อให้คลาสอื่นๆ เข้ามาสืบทอด (Inherit)
- 3) คลาสโปรโตไทป์ (Class Prototype) เป็นคลาสที่สืบทอดมาจากคลาสอาร์ทิแฟกต์ ทำหน้าที่เป็นตัวแทนฟังก์ชันโปรโตไทป์
- 4) คลาสฟังก์ชัน (Class Function) เป็นคลาสที่สืบทอดมาจากคลาสอาร์ทิแฟกต์ ทำหน้าที่เป็นตัวแทนฟังก์ชัน
- 5) คลาสดีคลิเรชัน (Class Declaration) เป็นคลาสที่สืบทอดมาจากคลาสอาร์ทิแฟกต์ ทำหน้าที่เป็นตัวแทนตัวแปร
- 6) คลาสสตรักเจอร์ (Class StructDef) เป็นคลาสที่สืบทอดมาจากคลาสอาร์ทิแฟกต์ ทำหน้าที่เป็นตัวแทนสตรักเจอร์

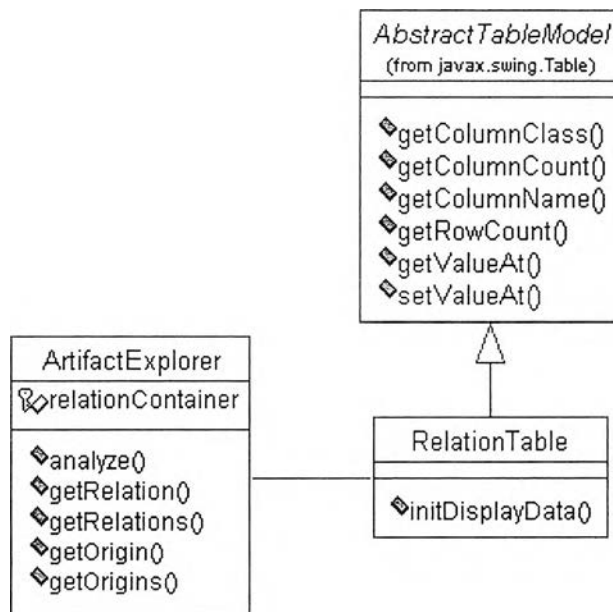
4.1.3 กลุ่มตัวค้นหา

กลุ่มตัวค้นหาประกอบไปด้วยคลาสที่เกี่ยวข้องจำนวน 2 คลาส ดังแสดงในรูปที่ 4.4 ซึ่งมีรายละเอียดดังนี้

- 1) คลาสอาร์ทิแฟกต์เอ็กซ์พลอเรอร์ (Class ArtifactExplorer) ทำหน้าที่ค้นหาความสัมพันธ์ระหว่างส่วนคำสั่งกับส่วนข้อมูล



รูปที่ 4.3 คลาสภายในกลุ่มตัวแฉง



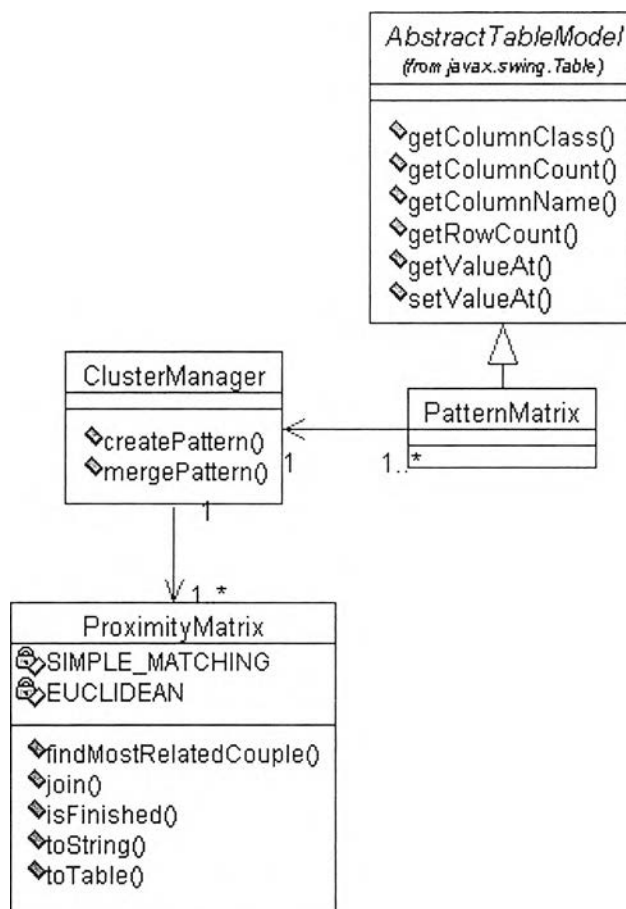
รูปที่ 4.4 คลาสภายในกลุ่มตัวค้นหา

- 2) คลาสรีเลย์ชั้นเทเบิล (Class RelationTable) เป็นคลาสที่สืบทอดมาจากคลาสแอ็บสแตรกต์เทเบิลโมเดล (Class AbstractTableModel) ซึ่งเป็นคลาสมาตรฐานของชุดพัฒนาซอฟต์แวร์จาวารุ่น 1.3 มีหน้าที่แสดงความสัมพันธ์ระหว่างส่วนคำสั่งและส่วนข้อมูล

4.1.4 กลุ่มตัววิเคราะห์จัดกลุ่ม

กลุ่มตัววิเคราะห์จัดกลุ่มประกอบด้วยคลาส 3 คลาส ดังแสดงในรูปที่ 4.5 ซึ่งมีรายละเอียดดังนี้

- 1) คลาสคลัสเตอร์แมนเนเจอร์ (Class ClusterManager) ทำหน้าที่ควบคุมการจัดกลุ่มข้อมูล
- 2) คลาสแพทเทิร์นเมตริกซ์ (Class PatternMatrix) เป็นคลาสที่สืบทอดมาจากคลาสแอ็บสแตรกต์เทเบิลโมเดล (Class AbstractTableModel) ทำหน้าที่เป็นตัวแทนเมตริกซ์ข้อมูล
- 3) คลาสพร็อกซิมีตีเมตริกซ์ (Class ProximityMatrix) ทำหน้าที่เป็นตัวแทนเมตริกซ์ความแตกต่าง



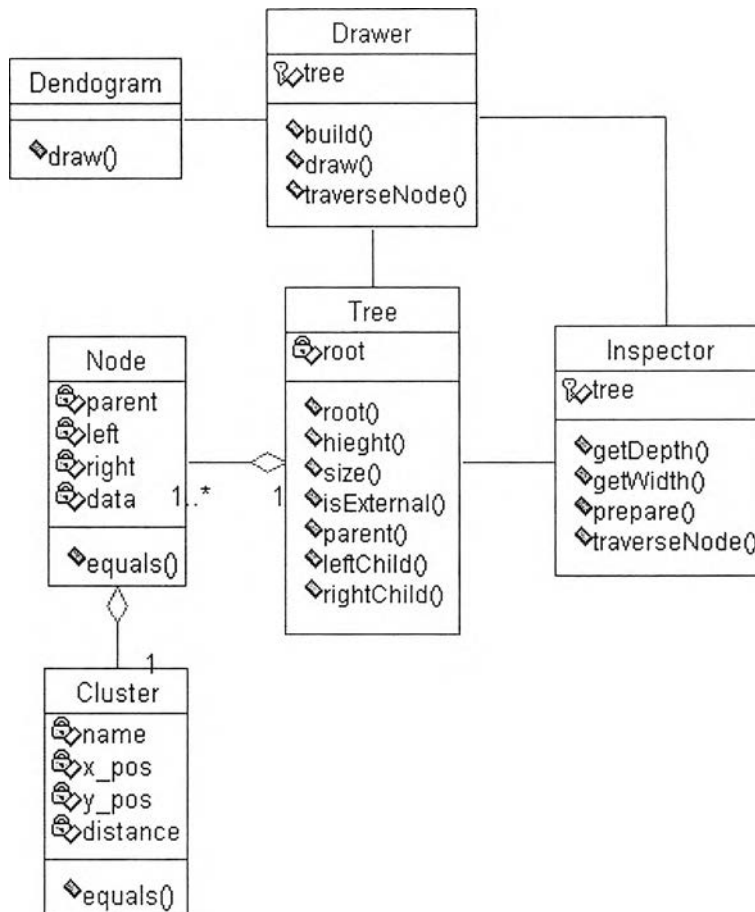
รูปที่ 4.5 คลาสในกลุ่มตัววิเคราะห์จัดกลุ่ม

4.1.5 กลุ่มตัววาด

กลุ่มตัววาดประกอบด้วยคลาส 6 คลาส ดังแสดงในรูปที่ 4.6 ซึ่งมีรายละเอียดดังนี้

- 1) คลาสเดนโดแกรม (Class Dendogram) ทำหน้าที่ควบคุมการวาดเดนโดแกรม
- 2) คลาสทรี (Class Tree) ทำหน้าที่เป็นตัวแทนต้นไม้ข้อมูล
- 3) คลาสโหนด (Class Node) ทำหน้าที่เป็นตัวแทนโหนดของต้นไม้ข้อมูล

- 4) คลาสคลัสเตอร์ (Class Cluster) ทำหน้าที่เป็นตัวแทนคลัสเตอร์
- 5) คลาสอินสเปกเตอร์ (Class Inspector) ทำหน้าที่คำนวณพิกัดการวาดของโหนดแต่ละโหนด ตลอดจนความสูงและความกว้างของต้นไม้ข้อมูล
- 6) คลาสดรอเวอร์ (Class Drawer) ทำหน้าที่วาดข้อมูลที่เก็บไว้ในโหนด ณ พิกัดที่คลาสอินสเปกเตอร์ได้คำนวณไว้



รูปที่ 4.6 คลาสในกลุ่มตัววาด

4.2 การพัฒนาเครื่องมือ

เครื่องมือระบุวัตถุซอฟต์แวร์ที่เป็นไปได้ๆ ถูกพัฒนาขึ้นด้วยภาษาจาวา โดยใช้ชุดพัฒนาซอฟต์แวร์จาวา (Java Software Development Kit) รุ่น 1.3 ร่วมกับโปรแกรมสภาพแวดล้อมการพัฒนาแบบรวม (Integrated Development Environment) ฟอर्ट (Forte) เวอร์ชัน 2.0 โปรแกรมต้นฉบับของเครื่องมือดังกล่าวประกอบด้วยคลาสจำนวน 21 คลาส นับจำนวนบรรทัดคำสั่งได้ทั้งหมด 5,000 บรรทัด (โดยประมาณ)