บทความวิจัย

ประสิทธิภาพและประสิทธิผลของซอฟต์แวร์สนับสนุนการเขียนโปรแกรม : กรณีศึกษา "แนวคิดเชิงวัตถุ"

The Efficiency and Effectiveness of Programming Supporting Software: A Case Study with Object Oriented Concepts

สุดา เธียรมนตรี[1*] อรยา ปรีชาพานิช[1] และปรีดาภรณ์ กาญจนสำราญวงศ์[2]

Suda Thianmontri[1*], Oraya Preechapanich[1] and Preedaporn Kanjanasumranwong[2]

บทคัดย่อ

ซอฟต์แวร์สนับสนุนการเขียนโปรแกรม เป็นเครื่องมือที่พัฒนาขึ้นมาเพื่อช่วยนักเขียนโปรแกรมมือใหม่ที่มักพบปัญหาในการเขียนโปรแกรมด้วยแนวคิดเชิงวัตถุ ซึ่งเป็นแนวคิดที่มีลักษณะเป็นนามธรรม ยากต่อการทำความเข้าใจ ผู้วิจัยพัฒนาซอฟต์แวร์สนับสนุนการเขียนโปรแกรม โดยใช้กรณีศึกษาเรื่องแนวคิดเชิงวัตถุ เพื่อใช้เรียนรู้และช่วยเขียนโปรแกรมในส่วนของแนวคิดที่ประกอบด้วยคลาส เมธอด แอททริบิวท์ อินเฮอริแทนซ์ และโพลิมอร์ฟีซึม โดยการป้อนข้อมูลผ่านส่วนติดต่อผู้ใช้ที่มีลำดับขั้นตอน แล้วซอฟต์แวร์จะทำการสร้างโปรแกรมเป็นภาษาจาวา ผู้ใช้ยังสามารถเปลี่ยนแปลงหรือเพิ่มเติมคำสั่งในโปรแกรมได้ ซอฟต์แวร์นี้ได้รับการตรวจสอบด้านความสอดคล้องกันของผลลัพธ์กับข้อมูลที่ใช้ตรวจสอบโดยผู้เชี่ยวชาญ จำนวน 10 คน พบว่าซอฟต์แวร์มีค่าดัชนีความสอดคล้องเฉลี่ยเท่ากับ 0.90 ผู้วิจัยนำซอฟต์แวร์ไปทดลองใช้จริงกับผู้เรียน จำนวน 30 คน เพื่อประเมินประสิทธิภาพและประสิทธิผลของซอฟต์แวร์จากผลการทำแบบฝึกหัด แบบทดสอบก่อนเรียนและหลังเรียน ที่มีความยากง่ายและอำนาจจำแนกรายข้อเท่ากับ 0.23-0.77 และ 0.20-0.60 ตามลำดับ ผลการประเมินพบว่าประสิทธิภาพของซอฟต์แวร์เท่ากับ 81.11/81.17 และประสิทธิผลของซอฟต์แวร์เท่ากับ 0.88 ซึ่งอยู่ในระดับที่ยอมรับได้

คำสำคัญ : ประสิทธิภาพและประสิทธิผลของซอฟต์แวร์ ซอฟต์แวร์สนับสนุนการเขียนโปรแกรม แนวคิดเชิงวัตถุ

---

[1] อาจารย์ สาขาวิชาคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยทักษิณ พัทลุง 93110
Lecturer, Department of Computer and Information Technology, Faculty of Science, Thaksin University, Phatthalung, 93110

[2] อาจารย์ สาขาวิชาคณิตศาสตร์และสถิติ คณะวิทยาศาสตร์ มหาวิทยาลัยทักษิณ พัทลุง 93110
Lecturer, Department of Mathematics and Statistic, Faculty of Science, Thaksin University, Phatthalung, 93110

Abstract

Programming Supporting Software is a tool which developed to aid novices with their difficulties in learning to program with object oriented concepts. The concepts those are abstracts and difficulty to understand. We have developed the programming supporting software in the topic of object oriented concepts: class, method, attributes inheritance and polymorphism. The software offer a graphical environment for user to input data and the construc tion of java source code is performed. It also allows user to modify and update some instruction code. The functional test is evaluated by 10 specialists and the result is shown as Index of Item Objective Congruence is 0.90. We conduct the study of software efficiency and effectiveness on 30 students with practically learn to pro gram, do an exercise and take a pretest - posttest exam which has the difficulty index between 0.23 and 0.77, the discrimination power between 0.20 and 0.60. The software efficiency and effectiveness index is 81.11/81.17 and 0.88 respectively. The result demonstrates that the efficiency and effectiveness of programming supporting software are satisfied.

Keywords : Software Efficiency and Effectiveness, Programming Supporting Software, Object Oriented Concepts

คำนำ

แนวคิดเชิงวัตถุได้ถูกนำมาใช้ในการเขียนโปรแกรมและได้รับความนิยมเป็นอย่างมาก เนื่องจากมีลักษณะที่เน้นการเขียนโปรแกรมแยกเป็นส่วน ๆ เป็นอิสระจากกัน ทำให้ลดระดับความซับซ้อนของโปรแกรม ซึ่งเป็นวิธีที่แก้ปัญหาได้ระดับหนึ่ง [1] ปัญหาของการเขียนโปรแกรมด้วยแนวคิดเชิงวัตถุคือ แนวคิดที่มีลักษณะค่อนข้างเป็นนามธรรม ยากต่อการทำความเข้าใจประกอบกับการใช้เครื่องมือเขียนโปรแกรมประเภท Text - Editor ที่ถูกออกแบบมาเพื่อการเขียนไฟล์แบบข้อความและเหมาะกับโปรแกรมเมอร์ที่มีความเชี่ยวชาญในระดับหนึ่ง ทำให้ผู้ที่เรียนในรายวิชาการเขียนโปรแกรมเชิงวัตถุหรือนักเขียนโปรแกรมเมอร์ใหม่มักพบปัญหาและไม่ประสบความ สำเร็จในการเขียนโปรแกรม ผู้วิจัยจึงมีแนวคิดที่จะพัฒนาซอฟต์แวร์สนับสนุนการเขียนโปรแกรม โดยใช้กรณีศึกษาเรื่องแนวคิดเชิงวัตถุที่อำนวยความสะดวกให้ผู้ใช้สามารถป้อนข้อมูลผ่านหน้าจอแทนการเขียนโปรแกรมโดยตรง ผู้วิจัยเลือกใช้ภาษาจาวาและข้อมูลเรื่องแนวคิดเชิงวัตถุสำหรับการพัฒนาซอฟต์แวร์นี้ เนื่องจากแนวคิดเชิงวัตถุเป็นพื้นฐานสำคัญของการเขียนโปรแกรมเชิงวัตถุและจาวาเป็นภาษาโปรแกรมที่ได้รับความนิยม สามารถสร้างส่วนของโปรแกรมและคอมไพล์ให้เป็น Java Byte Code แล้วนำไปใช้งานทุกที่ที่มีการติดตั้ง Java Virtual Machine Program ได้ กลไกการทำงานของภาษาจาวา เริ่มจากการเขียนโปรแกรม Java Source Code เมื่อนำไปคอมไพล์จะได้เป็นภาษาเฉพาะเรียกว่า Java Byte Code เมื่อนำไปรันโดย Java Virtual Machine Program และให้ Java interpreter แปลแล้ว จะได้เป็นผลการทำงานของโปรแกรม [2]

แนวคิดเชิงวัตถุ

Class & Object เป็นองค์ประกอบที่มีความสัมพันธ์กัน ในหลักการโปรแกรมเชิงวัตถุจะต้องสร้างคลาสเพื่อเป็นต้นแบบให้กับออบเจ็กต์ โดยการสร้างคลาสจะมีการกำหนดกลุ่มข้อมูลแอททริบิวท์และเมธอดเข้าด้วยกัน เมื่อนำคลาสไปใช้งานจะต้องสร้างออบเจ็กต์จากคลาสที่มีอยู่ ออบเจ็กต์ดังกล่าวสามารถใช้ข้อมูลของแอททริบิวท์และเมธอดที่มีอยู่ในคลาสนั้นได้ [3]

Encapsulation เป็นการห่อหุ้มและปกปิด

ข้อมูล [4] เพื่อป้องกันการแก้ไขข้อมูล หมายถึง การป้องกันการกำหนดค่าใหม่แทนที่ค่าเดิมให้กับข้อมูลโดยตรง สามารถทำได้โดยสร้างคลาสที่มีการรวมกลุ่มข้อมูลแอททริบิวท์และเมธอดเข้าด้วยกัน และกำหนดสถานะข้อมูลแอททริบิวท์เป็น Private การแก้ไขข้อมูลจะต้องทำผ่านเมธอดที่มีสถานะเป็น Public เท่านั้น

Inheritance เป็นการสืบทอดแอททริบิวท์และเมธอดของคลาสที่สร้างไว้ในคลาสแม่ไปให้กับคลาสลูกทำให้คลาสลูกไม่จำเป็นต้องสร้างแอททริบิวท์และเมธอดขึ้นเองแต่เรียกใช้จากคลาสแม่ สามารถสร้างกลุ่มแอททริบิวท์และเมธอดเพิ่ม เพื่อใช้งานในคลาสลูกเองได้

Polymorphism เป็นคุณสมบัติการมีได้หลายรูปแบบ คือการอนุญาตให้คลาสใด ๆ มีเมธอดที่ชื่อเหมือนกันได้มากกว่า 1 เมธอด มี 2 ลักษณะ คือ Overloading Method และ Overriding Method โดย Overloading Method หมายถึงการใช้ชื่อเมธอดเหมือนกันมากกว่า 1 เมธอด แต่จะแยกความต่างได้ด้วยชนิดข้อมูลหรือพารามิเตอร์ ส่วน Overriding Method หมายถึงการใช้ชื่อเมธอดที่เหมือนกันในคลาสลูก ๆ ที่มีการสืบทอดคุณสมบัติมาจากคลาสแม่เดียวกัน ทำให้คลาสลูก ๆ มีกลุ่มแอททริบิวท์และเมธอดเหมือนกัน

เครื่องมือช่วยเขียนโปรแกรมเชิงวัตถุ

เป็นชุดพัฒนาภาษาโปรแกรมที่ประกอบด้วยสภาพแวดล้อมที่อำนวยความสะดวกสำหรับการทำงานซึ่งจะแตกต่างกันไป เช่น ภาษาโปรแกรมที่ใช้เขียนโปรแกรมภาษา C++ ได้แก่ Turbo C++ for Dos หรือ Borland C++ เป็นต้น ส่วนภาษาโปรแกรมที่ใช้ในการเขียนโปรแกรมเชิงวัตถุ โปรแกรมเมอร์อาจเลือกใช้ NetBeans ซึ่งเหมาะกับผู้ที่มีความเชี่ยวชาญในภาษาโปรแกรมเชิงวัตถุ และต้องการเขียนโปรแกรมเพื่อการทำงานที่ซับซ้อน ส่วนผู้ที่เริ่มหัดเขียนโปรแกรมสามารถใช้ EditPlus หรือโปรแกรมประเภท Text Editor ซึ่งเป็นโปรแกรมที่ออกแบบมาเพื่อการเขียนไฟล์ที่เป็นข้อความได้

NetBeans เป็นเครื่องมือที่โปรแกรมเมอร์ใช้สำหรับพัฒนาแอพพลิเคชันด้วยภาษาจาวา มีการจัดทำเป็นรูปแบบซอฟต์แวร์ประเภทโอเพ่นซอร์ส ผู้สนใจนำไปใช้งานได้ฟรีและสามารถนำไปดัดแปลงแก้ไขได้ มีคุณสมบัติที่นักพัฒนาโปรแกรมในระดับสูงต้องการ เนื่องจากมีจุดเด่นคือ สามารถสร้างส่วนติดต่อผู้ใช้แบบกราฟิก [5] และรองรับการเขียนโปรแกรมในระดับที่ซับซ้อน จึงไม่เหมาะกับผู้ที่เริ่มเขียนโปรแกรมที่ต้องการเรียนรู้วิธี การเขียนโปรแกรมในระดับพื้นฐาน

EditPlus เป็นเครื่องมือที่โปรแกรมเมอร์ส่วนใหญ่ใช้ในการเขียนโปรแกรม มีการแบ่งสีสันที่ชัดเจนในโปรแกรมที่เขียน ทำให้มองเห็นความแตกต่างของโค้ดได้ง่าย ด้วยความมีขนาดเล็กสามารถทำงานได้หลาย ๆ ไฟล์พร้อมกัน รวมถึงความยืดหยุ่นเรื่องของการกำหนดนามสกุลของไฟล์ จึงทำให้ EditPlus เป็นโปรแกรมหนึ่งที่ได้รับความนิยม แต่อาจมีข้อจำกัดที่เป็นโปรแกรมประเภทเขียนไฟล์ที่เป็นข้อความ ไม่เหมาะกับผู้ที่เริ่มหัดเขียนโปรแกรม เนื่องจากไม่สามารถมองเห็นภาพรวมการทำงานของโปรแกรม และอาจทำให้ไม่เข้าใจในวิธีการเขียนโปรแกรม จึงไม่ประสบความสำเร็จในการเขียนโปรแกรมเชิงวัตถุ

งานวิจัยที่เกี่ยวข้อง

นวลจันทร์ และคณะ [6] ได้ทำวิจัยเรื่องการประเมินระบบแสดงมโนภาพของโปรแกรมที่มีผลต่อการเรียนด้านการเขียนโปรแกรมเชิงวัตถุ โดยใช้โปรแกรม Jeliot3 ซึ่งเป็นซอฟต์แวร์ประเภท Program Visualization ที่ช่วยในการเรียนด้านการเขียนโปรแกรมเชิงวัตถุมาเป็นเครื่องมือในการเรียนภาษาจาวาและทดลองกับนักศึกษาภาค วิชาเทคโนโลยีสารสนเทศ จำนวน 79 คน ซึ่งแบ่งเป็น 2 กลุ่มคือ กลุ่มที่เรียนภาษาจาวาที่ไม่มีการใช้โปรแกรม Jeliot3 และกลุ่มที่เรียนภาษาจาวาโดยใช้โปรแกรม Jeliot3 แสดงการประมวลผลของภาษาจาวา ผลการวิจัยพบว่า โปรแกรม Jeliot3 สามารถช่วยนักศึกษาในการเรียนการเขียนโปรแกรมเชิงวัตถุ แต่โปรแกรม

Jeliot3 ไม่สามารถเพิ่มทัศนคติของนักศึกษาที่มีต่อการเขียนโปรแกรมได้

Teemu Rajala et al. [7] ได้การทำวิจัยเรื่อง "Effectiveness of Program Visualization: A Case Study with the ViLLE Tool" โดยพัฒนาโปรแกรม ViLLE ซึ่งเป็นโปรแกรมประเภท Program Visualization ที่ช่วยในการเรียนรู้การเขียนโปรแกรมโดยภาษาโปรแกรมต่าง ๆ และทำการประเมินประสิทธิผลของโปรแกรม ViLLE โดยทำการทดลองกับนักศึกษาที่ไม่เคยเรียนวิชาการเขียนโปรแกรมมาก่อน 2 กลุ่ม คือกลุ่มที่เรียนโดยไม่มีการใช้โปรแกรม ViLLE และกลุ่มที่เรียนโดยใช้โปรแกรม ViLLE ผลการวิจัยพบว่าโปรแกรม ViLLE สามารถช่วยให้นักศึกษาที่ไม่เคยเรียนวิชาการเขียนโปรแกรมมาก่อนเกิดการเรียนรู้ และพัฒนาการเรียนรู้การเขียนโปรแกรมให้ดีขึ้นได้

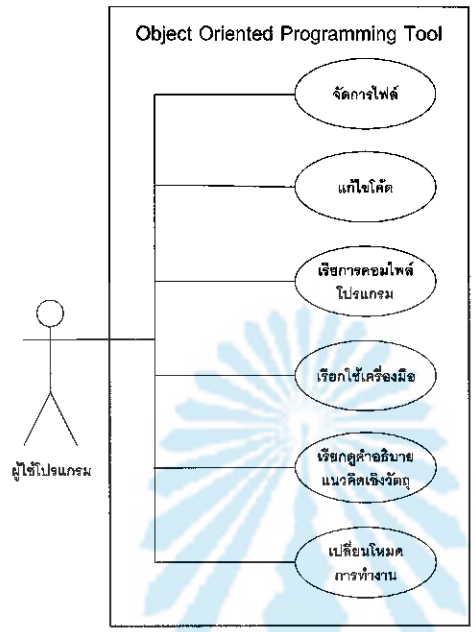
Samy S. and Abu Naser. [8] ได้ทำการวิจัยเรื่อง "JEE-Tutor: An Intelligent Tutoring System For Java Expressions Evaluation" โดยพัฒนาระบบช่วยสอนอัจฉริยะ JEE-Tutor ซึ่งประกอบด้วย ส่วนการออกแบบฐานความรู้ การออกแบบโมดูลผู้เชี่ยวชาญ การออกแบบโมดูลผลตอบกลับ และการออกแบบส่วนของหน้าจอการใช้งาน ซึ่งสอนในเรื่องไวยากรณ์การสร้างนิพจน์ ลำดับตัวดำเนินการ การสร้างประโยคและการประมวลผลภาษาจาวา ให้แก่นักศึกษาชั้นปีที่ 1 คณะวิศวกรรมและเทคโนโลยีสารสนเทศ มหาวิทยาลัย Al-Azhar ผลการวิจัยพบว่าระบบช่วยสอนอัจฉริยะ JEE-Tutor ช่วยให้นักศึกษาเรียนรู้เกี่ยวกับลำดับตัวดำเนินการ ความสัมพันธ์และการประมวลผลนิพจน์ในภาษาจาวาได้รวดเร็วและมีประสิทธิภาพมากกว่านักศึกษาที่เรียนรู้ด้วยวิธีการสอนแบบเดิม

จากการศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องมีการพัฒนาเครื่องมือในรูปแบบที่สามารถนำมาใช้ในการเรียนการสอนและตอบสนองความต้องการการใช้งานโปรแกรมหรือเครื่องมือดังกล่าวล้วนมีจุดเด่นที่อำนวยความสะดวกและส่งเสริมให้ผู้ใช้สามารถเลือกใช้เพื่อให้เกิดประสิทธิภาพกับงานมากที่สุด ส่วนผู้ที่เริ่มหัดเขียนโปรแกรมเชิงวัตถุที่ต้องการเรียนรู้ขั้นต้นนั้น ยังไม่มีเครื่องมือสนับสนุนมากนัก คณะผู้วิจัยจึงมีแนวคิดที่จะพัฒนาซอฟต์แวร์สนับสนุนการเขียนโปรแกรมโดยใช้กรณีศึกษาเรื่องแนวคิดเชิงวัตถุที่ผ่านการตรวจสอบด้านความสอดคล้องกันของผลลัพธ์กับข้อมูลที่ใช้ตรวจสอบและประเมินประสิทธิภาพและประสิทธิผลของซอฟต์แวร์ โดยมีสมมติฐานการวิจัย คือ ซอฟต์แวร์สนับสนุนการเขียนโปรแกรมมีประสิทธิภาพไม่ต่ำกว่าเกณฑ์ 80/80 และมีประสิทธิผลไม่ต่ำกว่า 0.80

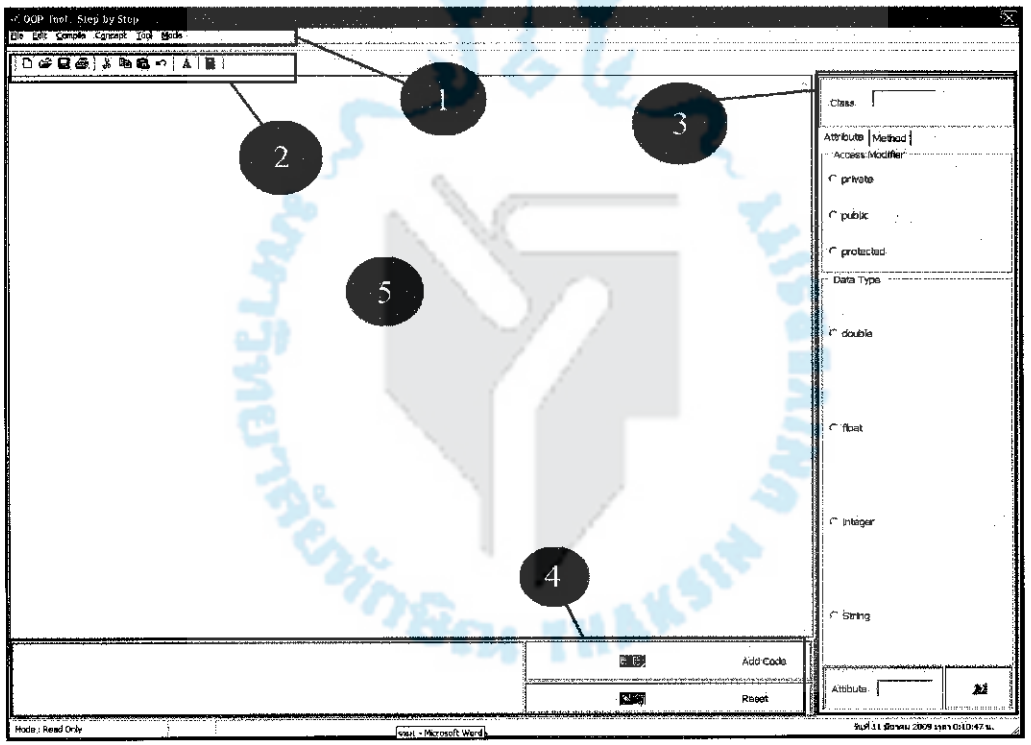
อุปกรณ์และวิธีการ

ผู้วิจัยได้ดำเนินการวิจัยใน 2 ขั้นตอนคือ 1) ออกแบบและพัฒนาซอฟต์แวร์สนับสนุนการเขียนโปรแกรม โดยใช้กรณีศึกษาเรื่องแนวคิดเชิงวัตถุ และ 2) ประเมินประสิทธิภาพและประสิทธิผลของซอฟต์แวร์ วิธีการดำเนินการวิจัยเพื่อพัฒนาซอฟต์แวร์ กลุ่มตัวอย่างที่ใช้ คือ ข้อมูลภาษาจาวาและข้อมูลเรื่องแนวคิดเชิงวัตถุ เครื่องมือที่ใช้ในการวิจัย คือซอฟต์แวร์ใช้โปรแกรม Borland Delphi 6.0 และฮาร์ดแวร์ใช้เครื่องคอมพิวเตอร์ ส่วนวิธีการดำเนินการวิจัยเพื่อประเมินประสิทธิภาพและประสิทธิผลของซอฟต์แวร์ กลุ่มตัวอย่างที่ใช้ คือ นิสิตชั้นปีที่ 3 สาขาวิชาวิทยาการคอมพิวเตอร์คณะวิทยาศาสตร์ มหาวิทยาลัยทักษิณ จำนวน 30 คน เครื่องมือที่ใช้ในการวิจัย คือแบบฝึกหัดและแบบทดสอบที่ผ่านการตรวจสอบคุณภาพมีค่าความยากง่ายรายข้อในช่วง 0.23-0.77 และค่าอำนาจจำแนกรายข้อในช่วง 0.20-0.60 สถิติที่ใช้ในการทดสอบ คือ E1:E2 ตามสูตรการคำนวณประสิทธิภาพ และคำนวณดัชนีประสิทธิผล (Effectiveness Index: E.I.) [9]

ผู้วิจัยได้ทำการวิเคราะห์และออกแบบซอฟต์แวร์ โดยแบ่งเป็นระบบการทำงานย่อยและมีความสัมพันธ์กับผู้ใช้โปรแกรม แสดงได้ดังภาพที่ 1 ในการออกแบบหน้าจอของซอฟต์แวร์นี้ได้แบ่งเป็น 5 ส่วน แสดงดังภาพที่ 2



ภาพที่ 1 Use Case Diagram ซอฟต์แวร์สนับสนุนการเขียนโปรแกรม



ภาพที่ 2 ส่วนประกอบของซอฟต์แวร์

ระบบการทำงานย่อยของซอฟต์แวร์ ประกอบด้วย 1) เมนูบาร์ (Menu Bar) เป็นส่วนที่แสดงเมนูคำสั่งของโปรแกรม ซึ่งใช้ในการทำงาน เช่น File, Edit, Compile, Concept, Tool และ Mode 2) เมนูลัด (Speed Bar) เป็นส่วนของกลุ่มคำสั่งที่ใช้บ่อย ภายใต้เมนูลัด ประกอบด้วยปุ่มแทนรายการต่าง ๆ ของเมนู 3) เครื่องมือเขียนชุดคำสั่ง OOP เป็นเครื่องมือที่ใช้ในการเขียนชุดคำสั่งแบบเชิงวัตถุ ประกอบด้วย 3 ส่วน คือ เป็นการกำหนดชื่อคลาส การกำหนด Access Modifier และชนิดข้อมูลของแอททริบิวท์ และการกำหนด Access Modifier และชนิดข้อมูลของเมธอด 4) ส่วนตรวจสอบโค้ด เป็นส่วนที่ใช้ในการตรวจสอบความถูกต้องของโค้ด เพื่อที่จะสามารถแก้ไขก่อนที่จะนำไปแสดงผลในส่วนแสดงผลต่อไป และ 5) ส่วนแสดงผลของโค้ดเป็นส่วนที่มีไว้สำหรับการเขียนโค้ดคำสั่ง เพื่อสร้างแอพพลิเคชั่นขึ้นมาใช้งาน ส่วนนี้จะแสดงผลการสร้างโค้ดที่ได้มาจากการใช้เครื่องมือ OOP และผู้ใช้สามารถพิมพ์โค้ดเพิ่มเติมได้ โดยต้องกำหนด Mode ให้เป็น Editable

ผลการวิจัยและอภิปรายผล

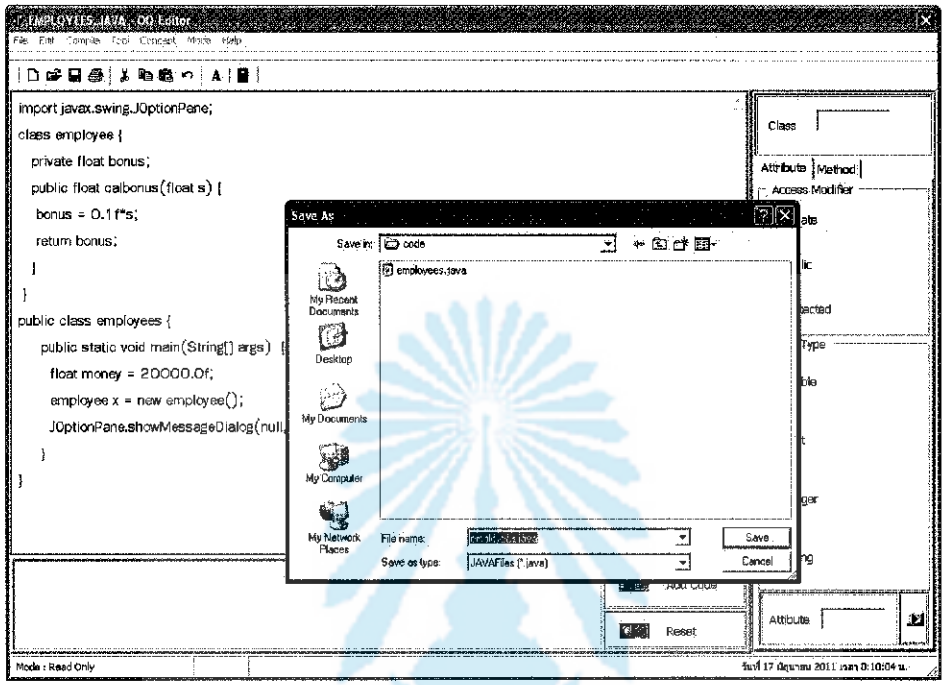
ผลการวิจัยพบว่าซอฟต์แวร์สนับสนุนการเขียนโปรแกรมสามารถสร้างโปรแกรมภาษาจาวา เรื่องแนวคิดเชิงวัตถุได้ โดยการป้อนข้อมูลผ่านส่วนติดต่อผู้ใช้ ผู้ใช้สามารถเปลี่ยนแปลงหรือเพิ่มเติมคำสั่งในส่วนของโปรแกรมได้ นอกจากนี้ผู้ใช้จะได้เรียนรู้ถึงแนวคิดเชิงวัตถุผ่านซอฟต์แวร์ที่มีคำอธิบายอย่างมีลำดับขั้นตอนและเข้าใจง่าย หลัง จากเขียนโปรแกรมแล้ว ผู้ใช้สามารถคอมไพล์และรันโปรแกรมผ่านเมนูคอมไพล์ ซึ่งแสดงผลการทำงานดังภาพที่ 3 และ 4 ตามลำดับ

ซอฟต์แวร์นี้ผ่านการทดสอบตามหลักการ Functional Testing ซึ่งเน้นผลที่เกิดขึ้นจากการทำงานของโปรแกรมโดยผู้เชี่ยวชาญ จำนวน 10 คน ทำการป้อนข้อมูลที่ต้องการทดสอบเข้าสู่กระบวนการทำงานของโปรแกรม [10] ผลการทดสอบพบว่า ส่วนติดต่อผู้ใช้ที่ประกอบด้วยเมนูย่อยสามารถทำงานได้ถูกต้องสอดคล้องกับข้อมูลที่ใช้ในการทดสอบ มีค่าดัชนีความสอดคล้องเฉลี่ยเท่ากับ 0.90

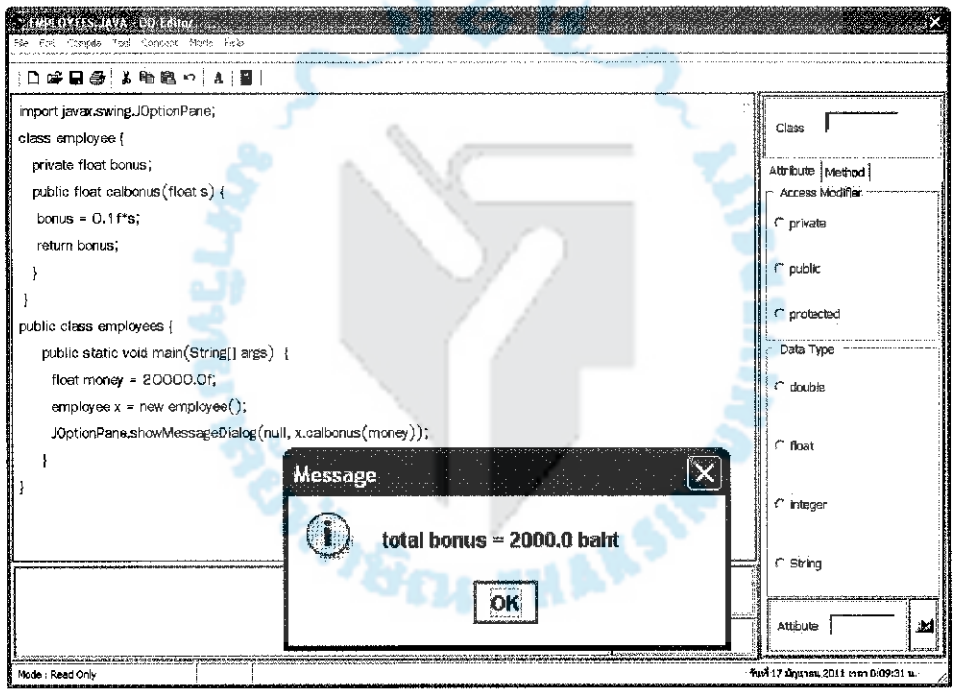
ผู้วิจัยทำการประเมินและประสิทธิผลของซอฟต์แวร์ตามสภาพจริง โดยนำไปทดลองใช้กับผู้เรียนจำนวน 30 คน และเก็บข้อมูลเพื่อคำนวณประสิทธิภาพตามสูตร E1:E2 และคำนวณดัชนีประสิทธิผล (Effectiveness Index: E.I.) ผลการประเมินพบว่าประสิทธิภาพของซอฟต์แวร์ (E1:E2) เท่ากับ 81.11/81.17 และประสิทธิผลของซอฟต์แวร์ (E.I.) เท่ากับ 0.88 เป็นไปตามสมมติฐานการวิจัย คือ ซอฟต์แวร์สนับสนุนการเขียนโปรแกรมมีประสิทธิภาพไม่ต่ำกว่าเกณฑ์ 80/80 และมีประสิทธิผลไม่ต่ำกว่า 0.80 ซึ่งอยู่ในระดับที่ยอมรับได้

สรุปผลการวิจัย

ซอฟต์แวร์สนับสนุนการเขียนโปรแกรมเป็นเครื่องมือที่ช่วยให้ผู้ใช้เขียนโปรแกรมได้ง่าย โดยป้อนข้อมูล ผ่านหน้าจอแบบมีลำดับขั้นตอน ผลการวิจัยพบว่าซอฟต์แวร์นี้สามารถสร้างโปรแกรมภาษาจาวาในเรื่องแนวคิดเชิงวัตถุได้ ผู้ใช้สามารถเปลี่ยนแปลงเพิ่มเติมส่วนโปรแกรมชุดคำสั่ง คอมไพล์ และรันโปรแกรม แล้วซอฟต์แวร์จะแสดงผลการทำงานของโปรแกรม ผลการทดสอบด้านความสอดคล้องกันของผลการทำงานกับข้อมูลที่ใช้ทดสอบ พบว่ามีค่าดัชนีความสอดคล้องเฉลี่ยเท่ากับ 0.90 และผลการประเมินประสิทธิภาพและประสิทธิผลของซอฟต์แวร์พบว่ามีค่าเท่ากับ 81.11/80.17 และ 0.88 ตามลำดับ ซึ่งเป็นไปตามสมมติฐานการวิจัย สามารถนำซอฟต์แวร์ไปใช้ในการเรียนการสอนสภาพแวดล้อมของซอฟต์แวร์ที่มีส่วนติดต่อผู้ใช้ในลักษณะกราฟิกที่ใช้งานง่าย มีความน่าสนใจ จะช่วยกระตุ้นให้ผู้เรียนเกิดกระบวนการเรียนรู้ได้ แนวทางในการพัฒนาต่อไป ทำได้โดยพัฒนาเพิ่มเติมในส่วนเนื้อหาเรื่องคำสั่งควบคุม เมธอดคอนสตรัคเตอร์ และอื่น ๆ เพื่อให้ครอบคลุมเนื้อหาของการเขียนโปรแกรมเชิงวัตถุทั้งหมด จะทำให้เครื่องมือมีความสมบูรณ์และน่าใช้งานมากขึ้น



ภาพที่ 3 ผลการทำงานของซอฟต์แวร์ (1)



ภาพที่ 4 ผลการทำงานของซอฟต์แวร์ (2)

คำขอบคุณ

งานวิจัยนี้เป็นส่วนหนึ่งของงานวิจัยที่ได้รับทุนสนับสนุนการทำวิจัย งบประมาณเงินรายได้ มหาวิทยาลัยทักษิณ ประจำปีงบประมาณ พ.ศ. 2550 และได้รับการสนับสนุนจากคณะวิทยาศาสตร์ มหาวิทยาลัยทักษิณ ที่ให้ความอนุเคราะห์อุปกรณ์คอมพิวเตอร์และฮาร์ดแวร์ต่าง ๆ ทำให้การทำวิจัยครั้งนี้สำเร็จลุล่วงด้วยดี ผู้วิจัยจึงขอขอบ คุณมา ณ โอกาสนี้

เอกสารอ้างอิง

[1] ธวัชชัย งามสันติวงศ์. (2549). **การวิเคราะห์และออกแบบระบบงานเชิงวัตถุ UML2**. กรุงเทพฯ: โรงพิมพ์ 21 เซ็นจูรี่ จำกัด.

[2] Lewis, J. and Loftus, W. (2005). **Java Software Solutions, foundation of programming design**. Fourth Edition, PEARSON Addison Wesley.

[3] ลาภลอย วานิชอังกูร. (2550). **เรียนรู้ด้วยตนเอง OOP C# ASP.NET**. กรุงเทพฯ: บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน).

[4] สุดา เธียรมนตรี. (2549). แนวคิดเชิงวัตถุ: Object – Oriented Concepts. **วารสารวิทยาศาสตร์ทักษิณ. 3**(2), 13-21.

[5] ปราการ ผู้วิบูลย์สุข. (2549). **JAVA สวย ง่าย ด้วย NetBeans และ iReport**. กรุงเทพฯ: บริษัท พิมพ์สวย จำกัด.

[6] นวลจันทร์ เกษมสุขสมบูรณ์ และคณะ. (2549). **การประเมินระบบแสดงมโนภาพของโปรแกรมที่มีผลต่อการเรียน ด้านการเขียนโปรแกรมเชิงวัตถุ**. สืบค้นเมื่อ 10 กรกฎาคม 2553 จาก http://www.cs.tu.ac.th/ file/abstract/I49_12_2549.pdf.

[7] Rajala, T. and Et al. (2008). Effectiveness of Pro gram Visualization: A Case Study with the ViLLE Tool. **Journal of Information Technology Education. 7**, 15-32.

[8] Samy, S.A. (2008). JEE-Tutor: An Intelligent Tutoring System for Java Expressions Eva luation. **Information Technology Journal. 7** (3), 528-532.

[9] บุญชม ศรีสะอาด. (2546). **การวิจัยสำหรับครู**. กรุงเทพฯ: สำนักพิมพ์สุวีริยาสาสน์. ชมรมเด็กจัดจำหน่าย.

[10] อรยา ปรีชาพานิช. (2549). การทดสอบซอฟต์แวร์. **วารสารวิทยาศาสตร์ทักษิณ**. 3(1), 21-31.