สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ภาษาคอมพิวเตอร์สำหรับเด็ก

COMPUTER LANGUAGE FOR KIDS

T 0 6 1 3 8 2

โดย

นายทวีศักดิ์ ปิยภากรวุฒิกุล

นายบวร พิชิตสกุลเดช

อาจารย์ที่ปรึกษา

ดร. อรัญญา วลัยรัชต์

อ.สมเกียรติ วังศิริพิทักษ์

เลขหมู่.....
เลขทะเบียน 61382
วัน,เดือน,ปี 17 ต.ค. 2549

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ภาษาคอมพิวเตอร์สำหรับเด็ก

COMPUTER LANGUAGE FOR KIDS

ผู้จัดทำ

1. นาย ทวีศักดิ์ ปียภากรวุฒิกุล รหัสประจำตัว 44010177
2. นาย บวร พิชิตสกุลเดช รหัสประจำตัว 44010265

อาจารย์ที่ปรึกษา

(ดร. อรัญญา วลัยรัชต์)

อาจารย์ที่ปรึกษา

(อ.สมเกียรติ วังศิริพิทักษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาคอมพิวเตอร์สำหรับเด็ก

นายทวีศักดิ์ ปิยภากรวุฒิกุล 44010177
นายบวร พิชิตสกุลเดช 44010265
อ. สมเกียรติ วังศิริพิทักษ์ อาจารย์ที่ปรึกษา
ดร. อรัญญา วลัยรัชต์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

เนื่องจากในปัจจุบันการเขียนโปรแกรม จะจำกัดอยู่ในกลุ่มของบุคคลที่มีความรู้ ความเข้าใจในภาษาคอมพิวเตอร์มาก่อน ทำให้คนที่ไม่มีความรู้ในภาษานั้นๆ จะต้องมีการศึกษาทั้ง รูปแบบ(syntax) และหลักการในการเขียนก่อนจึงสามารถใช้งานภาษานั้นได้อย่างชำนาญ ทำให้เป็นการลำบากที่จะให้เด็กสามารถเรียนรู้ได้ด้วยตัวเอง และทำให้เด็กเห็นว่าภาษาคอมพิวเตอร์เป็นเรื่องยาก และอาจเกลียดการเขียนโปรแกรมไปเลยก็ได้ ทั้งๆ ที่เด็กคนนั้นอาจมีความสามารถเพียงพอ ซึ่งเป็นการบั่นทอนบุคลากรที่มีความสามารถในอนาคตลงไปนั่นเอง

โปรแกรมภาษาคอมพิวเตอร์สำหรับเด็ก เป็นโครงงานที่มุ่งพัฒนาภาษาคอมพิวเตอร์ซึ่งสามารถเขียนโปรแกรมได้ด้วยการใช้รูปภาพที่เด็กสามารถเข้าใจได้ง่าย โดยการนำรูปไปวางแทนการพิมพ์รูปแบบของภาษาคอมพิวเตอร์ที่เป็นตัวอักษรอักขระ เมื่อวางรูปตามที่ต้องการแล้ว โปรแกรมจะแสดงผลขั้นตอนต่างๆ ของโปรแกรมโดยเรียงตามรูปที่วางไว้ทีละขั้นตอน ทำให้เข้าใจขั้นตอนการทำงานของโปรแกรมที่เขียนได้ง่าย

โปรแกรมภาษาคอมพิวเตอร์สำหรับเด็ก คือโครงงานที่กล่าวถึงการพัฒนาภาษาโปรแกรมคอมพิวเตอร์ โดยใช้ภาษาจาวา(Java) ใน JBuilder ในลักษณะวินด์โดว์ แอปปลิเคชัน(window application) เหมือนโปรแกรมวิชวล(visual) หลังจาก วางรูปเสร็จตามที่ต้องการแล้ว เมื่อต้องการให้แสดงผล ก็จะมีการเอารูปที่วางแปลงเป็นภาษาซี(C) เพื่อทำการประมวลทีละขั้นตอน พร้อมกับส่งผลลัพธ์กลับมาที่โปรแกรมเพื่อให้โปรแกรมแสดงรูปผลลัพธ์บนหน้าจอที่ผู้ใช้วางรูป โดยจะแสดงตามลำดับขั้นตอนของการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Computer Language For Kids

Thaveesak Piyapakornvutikul

Bovon Pichitsakuldej

Somkiet Wangsiripitak Advisor

Dr. Arunya Walairach Advisor

ABSTRACT

Nowadays, programming is only limited in a small group of people who have a computer background. In order to learn programming, the syntax and structure of programming is necessary and required. For beginner, these structure and command seem to be very difficult for them. As a result of complex structure and syntax, some of beginner might give up for this and leave this behind.

As a consequence, this project aims to develop a program providing a friendly interface with beginner in order to help them to understand programming easily. Rather than dealing with a very complex structure, the program will illustrate those complex commands in the form of picture which is very easy to understand and friendly with a new programmer.

"Computer Language For Kids" is implemented in Java language by using J-Builder. This technique is similar to the one in visual program because the output is illustrated in window application format. The user can do programming much more easily because all complex source-codes are transformed and converted into pictures displaying in window application format. Moreover, all of pictures codes in this program can be converted in C language as well. This makes the program be more general and world wide because "C" is one of the most important and useful language in the world.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ สมเกียรติ วังศิริพิทักษ์ และอาจารย์ อรัญญา วลัยรัชต์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

ทวีศักดิ์ ปิยภากรวุฒิกุล

บวร พิชิตสกุลเดช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญประกอบภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	2
1.5 ประโยชน์ที่ได้รับ	2
บทที่ 2 การออกแบบสำหรับเด็ก	3
2.1 การออกแบบสำหรับเด็ก	3
2.2 เด็กๆ มีความแตกต่างกันอย่างไร	3
บทที่ 3 การออกแบบโปรแกรมด้วยวิธีเชิงวัตถุ	5
3.1 หลักการของวิธีเชิงวัตถุ	5
3.2 คุณสมบัติของหลักการเชิงวัตถุ	5
3.2.1 การซ่อนข้อมูล	5
3.2.2 การถ่ายทอดคุณสมบัติ	6
3.2.3 โพลิมอร์ฟีซึม	6
3.3 ภาษาที่ใช้ในการเขียนโปรแกรมเชิงวัตถุ	6
บทที่ 4 พื้นฐานภาษาจาวา	8
4.1 ภาษาจาวา	8
4.2 หลักการทำงานของภาษาจาวา	8
4.3 การใช้คลาส Vector	9
4.4 การใช้คลาส JFileChooser	9
4.5 การใช้ File Reader/Writer	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ทฤษฎีต่างๆ ในการสร้างส่วนติดต่อกับผู้ใช้ 10
4.6.1 Scroll Pane 10
4.6.2 ปุ่ม (Button) 11
4.6.3 อิมเมจไอคอน (ImageIcon) 11
4.6.4 Even Listener Interfaces 11
4.6.4.1 KeyListener 11
4.6.4.2 MouseListener 12

บทที่ 5 การออกแบบไดอะแกรม 13
5.1 คลาสไดอะแกรม 13
5.1.1 คลาส GUI 14
5.1.2 คลาส BlackBox 14
5.1.3 คลาส CheckError 14
5.1.4 คลาส Translate 14
5.1.5 คลาส ReadFile 14
5.2 ท็อปดาวน์ไดอะแกรม 15
5.3 โฟล์วชาร์ทไดอะแกรม 16

บทที่ 6 การพัฒนาโปรแกรม 17
6.1 ส่วนที่ติดต่อกับผู้ใช้ 17
6.1.1 ส่วนสำหรับเลือกรูป 17
6.1.2 ส่วนสำหรับวางรูป 20
6.1.3 ส่วนปุ่มต่างๆ 20
6.2 การแปลงจากรูปภาพเป็นซอร์สโค้ด 21
6.3 การเก็บค่าตัวแปรบนส่วนวางรูปเพื่อแสดงผล 22
6.3.1 การเก็บตำแหน่งรูป 22
6.3.2 การเก็บค่าจำนวนตัวแปร 22
6.3.3 การเก็บค่าตำแหน่งลูป 23
6.3.4 การเก็บชื่อไฟล์ที่เป็นฟังก์ชัน 23
6.4 การบันทึกและการอ่านไฟล์ 23
6.4.1 การบันทึกลงไฟล์ 23
6.4.2 การอ่านค่าจากไฟล์ 23
6.5 การตรวจสอบรูปแบบคำสั่ง 23
6.6 การเรียกใช้ฟังก์ชันที่มองไม่เห็นการทำงาน 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6.1 การเตรียมนำฟังก์ชันที่มองไม่เห็นการทำงานขึ้นมาเพื่อใช้งาน 25
6.6.2 การเรียกใช้ฟังก์ชันที่มองไม่เห็นการทำงาน 25
6.7 การรันโปรแกรม 25
6.7.1 การแสดงผลทีละขั้นตอนการทำงานของโปรแกรม 26
6.7.2 การรันแล้วแสดงผลลัพธ์ของโปรแกรมโดยอัตโนมัติ 26

บทที่ 7 ผลการทดลอง 27
7.1 การทดลองประกาศตัวแปร และการบวก 27
7.2 การทดลองการลบ คูณ และหาร 29
7.3 การทดลองการตรวจสอบเงื่อนไขและการวนทำซ้ำ 32
7.4 การทดลองฟังก์ชันที่มองไม่เห็นการทำงาน และการเรียกตัวเอง 41

บทที่ 8 บทสรุปและวิจารณ์ 46
8.1 สรุปผลการดำเนินงาน 46
8.1.1 ส่วนการศึกษาโปรแกรมที่มีอยู่แล้ว 46
8.1.2 ส่วนการศึกษาพัฒนาการของเด็ก 46
8.1.3 ส่วนการพัฒนาการออกแบบส่วนติดต่อกับผู้ใช้ 46
8.1.4 ส่วนการทดลองการทำงานของโปรแกรมภาษาคอมพิวเตอร์สำหรับเด็ก 46
8.2 แนวทางการพัฒนาต่อ 46

บรรณานุกรม 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้าที่

รูปที่ 2-1 ตัวอย่างเด็กระดับ Sensori-motor กำลังเล่นกล่องดนตรี 4

รูปที่ 4-1 แสดงการสร้างสำหรับ FileReader และ FileWriter 10

รูปที่ 5-1 คลาสไดอะแกรมของโครงงานภาษาคอมพิวเตอร์สำหรับเด็ก 13

รูปที่ 5-2 ท็อปดาวน์ไดอะแกรมของโครงงานภาษาคอมพิวเตอร์สำหรับเด็ก 15

รูปที่ 5-3 โฟล์วชาร์ทไดอะแกรมของโครงงานภาษาคอมพิวเตอร์สำหรับเด็ก 16

รูปที่ 6-1 ตัวอย่างส่วนสำหรับเลือกรูป 17

รูปที่ 6-2 จุดเริ่มต้นโปรแกรม 17

รูปที่ 6-3 จุดสิ้นสุดโปรแกรม 17

รูปที่ 6-4 ตัวแปร 17

รูปที่ 6-5 ใช้ร่วมกับการบวก 18

รูปที่ 6-6 สัญลักษณ์บวก 18

รูปที่ 6-7 ใช้ร่วมกับการลบ 18

รูปที่ 6-8 สัญลักษณ์ลบ 18

รูปที่ 6-9 ใช้ร่วมกับการคูณ 18

รูปที่ 6-10 สัญลักษณ์คูณ 18

รูปที่ 6-11 ใช้ร่วมกับการหาร 18

รูปที่ 6-12 สัญลักษณ์หาร 18

รูปที่ 6-13 เงื่อนไข 18

รูปที่ 6-14 จบเงื่อนไข 18

รูปที่ 6-15 คำสั่งทำซ้ำ 19

รูปที่ 6-16 คำสั่งทำซ้ำหลังจากเลือกแล้ว 19

รูปที่ 6-17 ฟังก์ชันที่มองไม่เห็นการทำงาน 19

รูปที่ 6-18 ส้มที่ใช้เพิ่มค่า 19

รูปที่ 6-19 เส้นทางการทำงานทั้งหมด 7 แบบ 19

รูปที่ 6-20 แสดงภาพภายในส่วนสำหรับวางรูป 20

รูปที่ 6-21 ปุ่มต่างๆ 6 ปุ่มภายในโปรแกรม 20

รูปที่ 6-22 ตัวอย่างซอร์สโค้ดหลังจากแปลงเสร็จแล้ว 22

รูปที่ 7-1 ตัวอย่างโปรแกรม การทดลองประกาศตัวแปร และการบวก 27

รูปที่ 7-2 ตัวอย่างโปรแกรม ขณะทำการประกาศตัวแปร 27

รูปที่ 7-3 ตัวอย่างโปรแกรม ที่แสดงผลลัพธ์หลังทำการบวก 28

รูปที่ 7-4 ไฟล์ภาษาซีจากตัวอย่างโปรแกรม การทดลองประกาศตัวแปร และการบวก 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7-5 ตัวอย่างโปรแกรม การทดลองการลบ คูณ และหาร 29
รูปที่ 7-6 ตัวอย่างโปรแกรม ที่แสดงผลลัพธ์หลังทำการลบ 29
รูปที่ 7-7 ตัวอย่างโปรแกรม ที่แสดงผลลัพธ์หลังทำการคูณ 30
รูปที่ 7-8 ตัวอย่างโปรแกรม ที่แสดงผลลัพธ์หลังทำการหาร 30
รูปที่ 7-9 ไฟล์ภาษาซีจากตัวอย่างโปรแกรม การทดลองการลบ คูณ และหาร 31
รูปที่ 7-10 ตัวอย่างโปรแกรม การทดลองการตรวจสอบเงื่อนไขและการวนทำซ้ำ 32
รูปที่ 7-11 ตัวอย่างโปรแกรม ที่แสดงการเลือกเส้นทางจากเงื่อนไขครั้งแรก 33
รูปที่ 7-12 ตัวอย่างโปรแกรม ที่แสดงการวนกลับไปตรวจสอบจำนวนครั้งการวนทำซ้ำรอบแรก 34
รูปที่ 7-13 ตัวอย่างโปรแกรม ที่แสดงการเลือกเส้นทางจากเงื่อนไขครั้งที่สอง 35
รูปที่ 7-14 ตัวอย่างโปรแกรม หลังจากทำการบวกและวนกลับไปตรวจสอบ
จำนวนครั้งการวนทำซ้ำรอบที่สอง 36
รูปที่ 7-15 ตัวอย่างโปรแกรม หลังการลบจากการทำซ้ำในลูปรอบที่สาม 37
รูปที่ 7-16 ตัวอย่างโปรแกรม ที่แสดงการวนกลับไปตรวจสอบจำนวนครั้งการวนทำซ้ำรอบที่สาม 38
รูปที่ 7-17 ตัวอย่างโปรแกรม หลังจากการตรวจสอบจำนวนครั้งการวนทำซ้ำ
ที่ครบรอบแล้วออกจากลูป 39
รูปที่ 7-18 ไฟล์ภาษาซีจากตัวอย่างโปรแกรม ทดลองการตรวจสอบเงื่อนไขและการวนทำซ้ำ 40
รูปที่ 7-19 โปรแกรมหลักของการทดลองฟังก์ชันที่มองไม่เห็นการทำงาน และการเรียกตัวเอง 41
รูปที่ 7-20 โปรแกรมหลักของการทดลองฟังก์ชันที่มองไม่เห็นการทำงาน และการเรียกตัวเอง
ก่อนการเรียกใช้ฟังก์ชัน a 41
รูปที่ 7-21 ฟังก์ชัน a ซึ่งจะมองไม่เห็นการทำงานภายใน ในโปรแกรมหลัก 42
รูปที่ 7-22 โปรแกรมหลักของการทดลอง หลังผ่านการเรียกใช้ฟังก์ชัน a 42
รูปที่ 7-23 ฟังก์ชัน b ซึ่งจะมองไม่เห็นการทำงานภายใน ในโปรแกรมหลัก 43
รูปที่ 7-24 โปรแกรมหลักของการทำงาน หลังผ่านการเรียกใช้ฟังก์ชัน b 44
รูปที่ 7-25 ไฟล์ภาษาซีจากตัวอย่างโปรแกรม การทดลองฟังก์ชัน
ที่มองไม่เห็นการทำงาน และการเรียกตัวเอง 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากในปัจจุบันการเขียนโปรแกรม เริ่มมีการแพร่หลายมากขึ้น เนื่องจากโปรแกรมคอมพิวเตอร์ (Application) เกี่ยวข้องกับชีวิตประจำวันในแทบทุกๆ อย่าง เช่น โทรศัพท์มือถือ รถยนต์ เป็นต้น ทำให้การเขียนโปรแกรมไม่จำกัดอยู่ในเฉพาะกลุ่มของบุคคลที่ทำงานหรือศึกษาทางด้านคอมพิวเตอร์โดยตรงเท่านั้น แต่ยังมีบุคคลทั่วไป เด็กและเยาวชน หันมาเขียนโปรแกรมกันมากขึ้น ทำให้สามารถพัฒนาโปรแกรมที่ตนเองต้องการ เหมาะสมกับงานเฉพาะได้ แต่ภาษาที่ใช้ในการเขียนโปรแกรมยังคงเป็นอุปสรรคต่อการเขียนโปรแกรมของผู้ที่ไม่มีความรู้มาก่อน ถึงแม้ว่าจะเป็นภาษาที่ใช้กันอยู่ทั่วไป เช่น ภาษาปาสคาล (Pascal) ภาษาซี และภาษาจาวา เป็นต้น ก็ตาม

ดังนั้น ผู้พัฒนาจึงมีแนวคิดที่จะสร้างภาษาคอมพิวเตอร์ในอีกรูปแบบหนึ่งขึ้นมา เพื่อให้ง่ายต่อความเข้าใจและการใช้งานของผู้ใช้ อีกทั้งยังเป็นพื้นฐานในการเรียนรู้ทางด้านภาษาคอมพิวเตอร์ของเด็กต่อไปด้วย

โปรแกรมภาษาคอมพิวเตอร์สำหรับเด็ก เป็นโครงงานที่มุ่งเน้นผู้ใช้ที่เป็นเด็กอายุ 10 ปีขึ้นไป สามารถพัฒนาภาษาคอมพิวเตอร์ได้โดยการใช้รูปภาพที่เด็กสามารถเข้าใจได้ง่าย โดยการนำรูปไปวางแทนการพิมพ์รูปแบบของภาษาคอมพิวเตอร์ที่เป็นตัวอักษรอักขระ เมื่อวางรูปตามที่ต้องการแล้ว โปรแกรมสามารถแสดงขั้นตอนต่างๆ ของโปรแกรมโดยเรียงตามรูปที่วางไว้ทีละขั้นตอน ทำให้เข้าใจขั้นตอนการทำงานของโปรแกรมที่เขียนได้ง่าย

โครงงานนี้พัฒนาโดยใช้ภาษาจาวา ในโปรแกรม JBuilder ในลักษณะวินด์โดว์ แอปปลิเคชัน (window application) เหมือนโปรแกรมวิชวล (visual) โดยเมื่อเขียนโปรแกรมเสร็จแล้ว สามารถแปลงจากโปรแกรมที่เป็นรูปภาพเป็นภาษาซีได้ เนื่องจากภาษาซีเป็นภาษาที่ค่อนข้างง่ายและยังใช้ค่อนข้างแพร่หลายในปัจจุบัน

1.2 วัตถุประสงค์ของงานวิจัย

- เพื่อเป็นแนวทางในการศึกษาการเขียนโปรแกรม สามารถเข้าใจพื้นฐานของการเขียนโปรแกรม และหลักการด้านภาษาคอมพิวเตอร์ทั่วๆ ไป ว่ามีการทำงานเป็นขั้นตอน มีเงื่อนไขการทำงานอย่างไรบ้าง และเพื่อเป็นพื้นฐานในการศึกษาการเขียนโปรแกรมขั้นสูงต่อต่อไป
- เพื่อให้เด็กได้รับความเพลิดเพลินในการใช้โปรแกรม พร้อมกับเรียนรู้หลักการเขียนโปรแกรม
- เพื่อพัฒนาภาษาคอมพิวเตอร์สำหรับเด็ก
- เพื่อให้บุคคลทั่วๆ ไป หันมาศึกษาการเขียนโปรแกรม ภาษาคอมพิวเตอร์กันมากขึ้น เพื่อการพัฒนาของ software ของประเทศไทยต่อไป
- เพื่อพัฒนาภาษาคอมพิวเตอร์สำหรับเด็ก ที่ใช้รูปภาพให้สามารถใช้งานได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของงานวิจัย

โปรแกรมภาษาคอมพิวเตอร์สำหรับเด็กนี้พัฒนาขึ้นเพื่อเป็นพื้นฐานในการเขียนโปรแกรม ดังนั้นจึงมีเพียงคำสั่งการทำงานพื้นฐานเท่านั้น ประกอบด้วย คำสั่งคำนวณพื้นฐาน (บวก, ลบ, คูณ, หาร) เงื่อนไข (condition) การวนทำซ้ำ (loop) และฟังก์ชัน (function) จึงไม่สามารถนำไปใช้เขียนโปรแกรมที่มีขั้นตอนซับซ้อนและตัวแปรหลากหลายชนิดได้

ซึ่งในอนาคตสามารถพัฒนาคำสั่งต่างๆ ในการเขียนโปรแกรมที่ซับซ้อนเพิ่มขึ้นเพื่อให้สามารถพัฒนาเป็นโปรแกรมใหญ่ๆ ได้อย่างครบถ้วนต่อไป

1.4 วิธีการดำเนินงาน

งานวิจัยในโครงงานนี้จะเริ่มต้นด้วยการศึกษาทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลักๆ อยู่ 4 เรื่องด้วยกัน คือ ทฤษฎีพัฒนาการของเด็ก การเขียนโปรแกรมเชิงวัตถุ (Objcct Oricnted Programming) การเขียนโปรแกรมภาษาจาวา และโปรแกรม ROBOLAB ซึ่งจะมีรายละเอียดในบทที่ 2, 3, 4 และ 5 จากนั้นก็จะนำเอาความรู้ที่ได้มาออกแบบคลาสต่างๆ ซึ่งมีรายละเอียดในบทที่ 6

จากนั้นก็จะเริ่มเข้าสู่ขั้นตอนของการพัฒนาโปรแกรม โดยบทที่ 7 จะกล่าวถึงองค์ประกอบโดยรวมของโปรแกรมที่พัฒนาขึ้นมา ซึ่งมีอยู่ 5 ส่วนหลักๆ คือ การพัฒนา GUI (Graphic User Interface) การแปลงเป็นซอร์สโค้ด (source code) การอ่านไฟล์ การตรวจสอบรูปแบบคำสั่ง และการเรียกใช้ฟังก์ชัน

สำหรับบทที่ 8 ก็จะเป็นการทดสอบระบบรวมทั้งหมด และบทที่ 9 ซึ่งเป็นบทสุดท้ายก็จะเป็นการสรุปการทำงาน ผลที่ได้รับจากงานวิจัยชิ้นนี้ และแนวทางในการพัฒนางานวิจัยนี้เพิ่มเติม และแนวทางในการนำไปประยุกต์ใช้

1.5 ประโยชน์ที่ได้รับ

- ได้รับความรู้เกี่ยวกับพัฒนาการของเด็ก
- สามารถออกแบบคลาสการทำงานได้ดีขึ้น
- ได้รับความรู้ในการเขียนภาษาจาวามากขึ้น
- ได้ภาษาคอมพิวเตอร์สำหรับเด็กที่ใช้รูปภาพแทนภาษาคอมพิวเตอร์ทั่วไป เพื่อเป็นแนวทางในการศึกษา และพัฒนาภาษาคอมพิวเตอร์สำหรับเด็กต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การออกแบบสำหรับเด็ก

2.1 การออกแบบสำหรับเด็ก

การออกแบบซอฟต์แวร์และฮาร์ดแวร์สำหรับเด็กนั้นแตกต่างจากการออกแบบให้ผู้ใหญ่อย่างไร? ในขณะที่กำลังเขียนอยู่นี้ มีเพียงงานวิจัยที่น้อยมากที่ศึกษาเกี่ยวกับด้านนี้ ส่วนใหญ่งานวิจัยจะมุ่งไปที่การออกแบบซอฟต์แวร์เพื่อการศึกษา และประเมินผลจากการเรียนรู้เบื้องต้น ซึ่งคงไม่มีประโยชน์ หากเด็กๆ ไม่สามารถใช้งานได้อย่างมีประสิทธิภาพ

ในการออกแบบสำหรับเด็กนั้น ผู้คนมักสมมติว่าเด็กนั้นสร้างสรรค์ ฉลาดและสามารถทำสิ่งที่ดีๆ ได้ ถ้าได้รับเครื่องมือและการการสนับสนุนที่ดี แต่ถ้าเด็กไม่สามารถใช้เทคโนโลยีที่พวกเราออกแบบได้ ก็เท่ากับว่าเป็นความล้มเหลวของผู้ออกแบบ สมมติฐานเหล่านี้ถูกตั้งขึ้นเพราะว่าผู้ใช้จะขึ้นอยู่กับความคาดหวังของผู้ออกแบบ แต่ในความเป็นจริง สมมติฐานเดียวกันจะเป็นประโยชน์ในการออกแบบให้กับผู้ใหญ่ ผู้ออกแบบซอฟต์แวร์สำหรับเด็กเริ่มต้นด้วยดี เพราะพวกเค้าเชื่อในตัวเด็ก อย่างไรก็ตาม ก็อาจจะมีข้อเสียเพราะว่ามันผ่านช่วงเวลาที่ร่างกายและความเข้าใจเมื่อครั้งเป็นเด็กมานาน ทำให้นึกไม่ออก

2.2 เด็กๆ มีความแตกต่างกันอย่างไร

เนื่องจากมนุษย์เรามีการพัฒนาการจากเด็กทารกจนเป็นผู้ใหญ่ ทั้งทางด้านกายภาพและความสามารถในการเข้าใจเพิ่มขึ้นตลอดเวลา ทำให้นักจิตวิทยาชาวสวิสเซอร์แลนด์ จีน เพียเกท (Jean Piaget) ได้นำมาวิเคราะห์พัฒนาการความเข้าใจของเด็กว่าเป็นอย่างไร โดย เพียเกทแสดงให้เห็นว่า เด็กๆ ไม่ได้ขาดความรู้และประสบการณ์แต่ยังขาดประสบการณ์พื้นฐานด้วยและเข้าใจโลกแตกต่างกว่าผู้ใหญ่ เขาแบ่งพัฒนาการของเด็กออกเป็น 4 ระดับ

- ระดับ Sensori-motor (ตั้งแต่แรกเกิด - 2 ปี)

  เด็กในระดับนี้ ยากที่จะเข้าใจได้ทันทีว่าได้พบหรือเจอกับอะไร การออกแบบซอร์ฟแวร์สำหรับเด็กในระดับนี้ยากมาก สามารถคาดหวังการตอบรับเล็กๆ น้อยๆ ได้ ทุกๆคำสั่งต้องให้เป็นออดิโอ (audio) วิดีโอ (vdo) หรือแอนิเมชัน (animation) เนื่องจากเด็กในวัยนี้อ่านยังไม่ออก ยิ่งไปกว่านั้นเด็กยังไม่สามารถใช้เม้าส์ (mouse) ได้อย่างเชี่ยวชาญ จะใช้ได้ก็แต่รูปใหญ่ๆ เท่านั้น

- ระดับ Pre-operational (2 – 7 ปี)

  เด็กในระดับนี้ สามารถจดจำสิ่งๆ หนึ่งได้ในระยะเวลาหนึ่งเท่านั้น พวกเค้าไม่สามารถเข้าใจมุมมองของผู้อื่นได้ ในระดับนี้เด็กเริ่มอ่านได้ ทำให้การออกแบบสำหรับเด็กในระดับนี้สมมติได้ว่าเด็กพอจะอ่านได้ และเด็กในระดับนี้สามารถใช้และคลิกเม้าส์บนเป้าหมายได้ดีขึ้น สำหรับการใช้คีย์บอร์ดยังควรหลีกเลี่ยงอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระดับ Concrete Opearational (7 - 11 ปี)

เราจะเห็นได้ว่าเด็กในระดับนี้มีความเข้าใจที่เป็นผู้ใหญ่มากขึ้น แม้ว่าพวกเค้าไม่สามารถตั้งสมมติฐานและหลักการได้ เช่น ช่วงกว้างของตัวเลข ยังคงยากไปสำหรับเค้า แต่พวกเค้าสามารถแบ่งกลุ่มของสิ่งต่างๆได้

ที่เห็นได้ชัดเจนคือ เด็กโตเพียงพอที่จะใช้ซอฟต์แวร์ที่ซับซ้อนได้ แต่ด้วยความที่อายุน้อย ทำให้เด็กยังคงชอบสิ่งที่สนุกอยู่ เราสามารถคาดหวังการใช้งานคีย์บอร์ดเล็กๆ น้อยๆ ได้ ความสามารถในการพิมพ์ของเด็กมีมากขึ้นตามอายุ และยังคาดหวังการควบคุมเม้าส์ได้เป็นอย่างดีด้วย

- ระดับ Formal Operational (11 ปีขึ้นไป)

สุดท้ายนี้ ผู้ออกแบบสามารถคิดได้ว่า ความคิดของเด็กในระดับนี้โดยทั่วๆไปแล้วจะคล้ายกับของผู้ใหญ่ ความสนใจ ความรู้สึก แม้แต่ว่าความแตกต่างด้วย การออกแบบสำหรับเด็กในระดับนี้มีความท้าทายที่น้อยมาก เพราะว่าผู้ใหญ่ที่ออกแบบนี้สามารถใช้ความคิด ความเข้าใจของตัวเองได้

รูปที่ 2-1 ตัวอย่างเด็กระดับ Sensori-motor กำลังเล่นกล่องดนตรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบโปรแกรมด้วยวิธีเชิงวัตถุ

ในการออกแบบโปรแกรมด้วยวิธีเชิงวัตถุเป็นหลักการซึ่งมีองค์ประกอบต่างๆ ที่เรียกว่าโมเดลของวัตถุ (Object Model) ในโมเดลของวัตถุนั้นแบ่งออกเป็นการแยกแยะเอกลักษณ์ (Abstraction) การซ่อนข้อมูล (Encapsulation) การรวมกลุ่มความสัมพันธ์ (Modularity) ลำดับชั้น (Hicrarchy) ชนิด (Typing) การทำงานพร้อมกัน (Concurrency) และการรักษาสถานะ (Persistence) สำหรับลักษณะการวิเคราะห์และออกแบบระบบด้วยวิธีเชิงวัตถุ จะแตกต่างจากการออกแบบระบบด้วยวิธีโครงสร้าง เนื่องจากรูปแบบในการคิดและการมองระบบจะต่างกัน และภาษาที่ใช้สำหรับการเขียนโปรแกรมเชิงวัตถุแต่ละภาษาก็มีรายละเอียดที่ต่างกันด้วย

3.1 หลักการของวิธีเชิงวัตถุ

การพัฒนาระบบแบบโครงสร้างนั้นจะพยายามให้นักพัฒนาระบบแตกปัญหาออกเป็นหลายๆ ฟังก์ชัน (Function) หรือ โพรซีเยอร์ (Procedure) แล้วแก้ปัญหาในแต่ละส่วนด้วยอัลกอริทึม (Algorithm) และโครงสร้างข้อมูล (Data Structure) เฉพาะเพื่อช่วยให้การพัฒนาโปรแกรมเป็นไปได้ง่ายโดยเฉพาะโปรแกรมขนาดเล็ก

สำหรับการพัฒนาระบบด้วยวิธีเชิงวัตถุก็เช่นเดียวกัน แต่จะแก้ไขปัญหาโดยการแตกปัญหาที่กำลังพิจารณาออกเป็นส่วนย่อยๆ ซึ่งมีความซับซ้อนน้อยลงและเรียกแต่ละส่วนเหล่านี้ว่า "วัตถุ"

วัตถุต่างๆ เหล่านี้จะถูกประกอบขึ้นมาเป็นระบบที่สมบูรณ์ที่สุด ทำให้สามารถใช้ได้กับโปรแกรมขนาดใหญ่และมีความซับซ้อนเพิ่มมากขึ้นได้ การพัฒนาระบบด้วยวิธีเชิงวัตถุไม่เพียงจะเป็นแนวความคิดใหม่ในการเขียนโปรแกรมเท่านั้นแต่ยังมีข้อดีจากโมเดลของวัตถุ ซึ่งเป็นแนวความคิดใหม่ในวงการคอมพิวเตอร์อีกด้วย สามารถนำหลักการในการพัฒนาระบบด้วยวิธีเชิงวัตถุไปประยุกต์ใช้กับการออกแบบ ส่วนติดต่อกับผู้ใช้ (User Interface) ระบบฐานข้อมูลหรือแม้แต่การออกแบบสถาปัตยกรรมฮาร์ดแวร์ของคอมพิวเตอร์ เนื่องจากหลักการของการพัฒนาระบบด้วยวิธีเชิงวัตถุนั้นสามารถนำไปประยุกต์ใช้ในงานต่างๆ ได้อย่างกว้างขวาง

3.2 คุณสมบัติของหลักการเชิงวัตถุ

3.2.1 การซ่อนข้อมูล

Encapsulation หมายถึง การจัดกลุ่มองค์ความคิดที่คล้ายคลึงกันเข้าเป็นหน่วยเดียวกันเพื่อทำการอ้างถึงด้วยชื่อเดียวกัน ในเชิงซอฟต์แวร์ โปรแกรมเมอร์มักจะพบว่าชุดคำสั่งที่คล้ายคลึงกันปรากฏอยู่หลายครั้งในตัวโปรแกรม ดังนั้นจึงเกิดแนวความคิดในการจับรูปแบบการซ้ำดังกล่าวเข้ามาอยู่ในสิ่งเดียวกันอันเป็นที่มาของเอนแคปซุเลชันนั่นเอง สิ่งที่ตามมานอกจากความง่ายในการทำความเข้าใจตัวโปรแกรมแล้วยังเป็นการช่วยประหยัดหน่วยความจำอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแนวความคิดเชิงวัตถุ เอนแคปซูเลชัน หมายถึงการรวบรวมโอเปอเรชัน (operation) และแอตทริบิวต์ (attribute) เข้าเป็นหน่วยเดียวกันเพื่อที่ว่าแอตทริบิวต์สามารถถูกเปลี่ยนแปลงได้อย่างเหมาะสมโดยผ่านโอเปอเรชัน และเราเรียกผลที่เกิดจากการใช้งานแอนแคปซูเลชันว่า การซ่อนข้อมูล (information hiding)

3.2.2 การถ่ายทอดคุณสมบัติ

Inheritance คือ วิธีในการสร้างคลาสใหม่จากคลาสเดิมที่มีอยู่ ทั้งนี้คลาสที่สร้างขึ้นใหม่จะมีวัตถุประสงค์ในการทำงานที่เฉพาะเจาะจงมากยิ่งขึ้น

การถ่ายทอดคุณสมบัติและความสามารถของคลาสหนึ่งไปยังอีกคลาสหนึ่ง ลักษณะความสัมพันธ์ของคลาสที่ใช้ข้อมูลและพฤติกรรมร่วมกัน โดยมีความสัมพันธ์ระหว่างคลาสในลักษณะความสัมพันธ์แบบ "is-a" ในการถ่ายทอดคุณสมบัติของคลาสอันประกอบด้วยแอตทริบิวต์ (Attribute) และเมธอด (Method) ในที่นี้คลาสที่ถูกถ่ายทอดคุณสมบัติจะถูกเรียกว่าซูเปอร์คลาส (Superclass, Parent class) และเรียกคลาสที่ได้รับการถ่ายทอดคุณสมบัติว่าซับคลาส (Subclass, Child class, Derived class) ซึ่งทำให้ซอฟต์แวร์มีความสามารถนำกลับมาใช้ได้ (Reusable)

3.2.3 โพลิมอร์ฟิซึม

Polymorphism หมายถึง การเปลี่ยนรูปร่างของออบเจกต์หนึ่งๆ ซึ่งในเชิงโปรแกรมมิ่งจะเป็นการที่ตัวแปรออบเจกต์ของคลาสใดคลาสหนึ่งสามารถเปลี่ยนรูปแบบไปจากคลาสเดิมได้ เช่น

การสร้างออบเจกต์ของซับคลาสและกำหนดตัวแปรออบเจกต์ดังกล่าวให้เป็นประเภทซูเปอร์คลาสแทน นอกจากนี้ยังรวมถึงการที่โอเปอเรชันเดียวกันที่มีพฤติกรรมที่แตกต่างกันเมื่อถูกเรียกใช้จากออบเจกต์ที่เกิดจากคนละคลาสกัน

การส่งแมสเซจเดียวกันให้วัตถุที่ต่างกันทำให้วัตถุสามารถแสดงการทำงานที่ต่างกันออกมาได้ คุณสมบัตินี้ทำให้ซอฟต์แวร์มีความยืดหยุ่น (Flexible)

3.3 ภาษาที่ใช้ในการเขียนโปรแกรมเชิงวัตถุ

การเขียนโปรแกรมด้วยภาษาเชิงวัตถุ เป็นวิธีการในการสร้างโปรแกรมให้มีโครงสร้างในการทำงานร่วมกันของวัตถุ ซึ่งวัตถุนั้นจะเป็นตัวตนหรือเป็นอินสแทนซ์ (Instance) ของคลาส และคลาสก็เป็นส่วนหนึ่งของลำดับชั้น (Hierarchy) ของคลาสที่มีความสัมพันธ์ในการสืบทอดคุณสมบัติ (Inheritance) โดยคุณสมบัติเด่นของภาษาในการเขียนโปรแกรมเชิงวัตถุได้นั้นจะต้องมีคุณสมบัติ 3 ประการเรียกว่า PIE ซึ่งย่อมาจาก Polymorphism, Inheritance และ Encapsulation

ถ้าภาษาใดขาดคุณสมบัติหนึ่งในสามข้อนี้ไป ก็จะไม่เรียกว่าเป็นภาษาในการเขียนโปรแกรมเชิงวัตถุ ตัวอย่างเช่น ถ้าภาษาใดสนับสนุนการสร้างคลาสและวัตถุ แต่ไม่สนับสนุนการสืบทอดคุณสมบัติของคลาส เราจะเรียกภาษานั้นว่า ภาษาที่ทำงานบน Abstract Data Type หรือเรียกว่าภาษาในการเขียนโปรแกรมบนพื้นฐานของวัตถุ (Object Based) มิใช่ภาษาในการเขียนโปรแกรมเชิงวัตถุ (Object Oriented)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของภาษาในการเขียนโปรแกรมเชิงวัตถุเช่น Smalltalk, Object Pascal, C++, Eiffel, CLOS และ Java ส่วนภาษาในการเขียนโปรแกรมบนพื้นฐานของวัตถุก็เช่น Ada เป็นต้น ซึ่งในการที่จะเขียนโปรแกรมด้วยภาษาเชิงวัตถุให้ได้อย่างมีประสิทธิภาพ ผู้เขียนโปรแกรมควรจะมีการออกแบบด้วยหลักการพัฒนาระบบด้วยวิธีเชิงวัตถุเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
พื้นฐานภาษาจาวา

4.1 ภาษาจาวา

จาวาเป็นภาษาโปรแกรมขั้นสูงที่พัฒนาขึ้นมาโดยบริษัท Sun Microsystems โดยมีวัตถุประสงค์เพื่อให้เป็นภาษาระดับสูงที่ในการพัฒนาซอฟต์แวร์สำหรับอุปกรณ์ที่ควบคุมโดยไมโครโพรเซสเซอร์ (Embedded System) ที่สามารถนำไปรันได้โดยไม่ขึ้นกับแพลตฟอร์ม (Platform)

ภาษาจาวาได้รับการยอมรับว่าเป็นภาษาที่ดีที่สุดในการพัฒนาซอฟต์แวร์ที่ต้องการความปลอดภัยสูงแบบกระจายผ่านระบบเครือข่าย โดยสามารถใช้พัฒนาได้ตั้งแต่โปรแกรมของอุปกรณ์เน็ตเวิร์กที่ควบคุมโดยไมโครโพรเซสเซอร์ โปรแกรมในเว็บบราวเซอร์ จนถึงแอพพลิเคชันทั่วไป ตัวภาษาจาวามีลักษณะเด่นดังนี้

- เป็นภาษาพัฒนาโปรแกรมเชิงวัตถุ
- สามารถนำไปใช้งานได้โดยไม่ขึ้นกับแพลตฟอร์ม
- สามารถประมวลผลแบบกระจาย โดยดึงคลาสไลบรารีจากที่ต่างๆ ผ่านทาง HTTP และ FTP
- สามารถทำงานได้หลายงานพร้อมกันในเวลาเดียวกัน
- ไม่มีตัวแปรแบบพอยเตอร์ (Pointer) เพื่อขจัดปัญหาในการจัดการหน่วยความจำ
- สามารถตรวจสอบและจัดการกับความผิดพลาด (Exception) ที่เกิดขึ้น
- มีการรองรับมาตรฐานความปลอดภัยในการใช้งานรูปแบบต่างๆ

4.2 หลักการทำงานของภาษาจาวา

จาวาเป็นภาษาระดับสูงเชิงวัตถุซึ่งมีกลไกการแปลภาษาและการประมวลผลในลักษณะผสมผสานระหว่างแบบคอมไพเลอร์ (Compiler) และแบบอินเทอร์พรีเตอร์ (Interpreter) เพื่อจุดประสงค์ที่จะรับข้อดีของทั้งสองแบบ

แบบคอมไพเลอร์นั้น ตัวคอมไพเลอร์จะทำหน้าที่วิเคราะห์โค้ดภาษาระดับสูงและแปลเป็นภาษาเครื่อง (Machine code) ซึ่งจะมีข้อดีคือ จะได้โปรแกรมภาษาเครื่องที่มีประสิทธิภาพ เนื่องจากตัวคอมไพเลอร์ได้ทำการวิเคราะห์และปรับแต่งตัวโค้ดให้ได้ภาษาเครื่องที่มีประสิทธิภาพที่สุด แต่ข้อเสียของแบบนี้คือภาษาเครื่องที่ได้จะไม่สามารถนำไปใช้กับแพลตฟอร์มอื่นได้

ส่วนแบบอินเทอร์พรีเตอร์นั้น ตัวอินเทอร์พรีเตอร์จะทำการอ่านโค้ดภาษาระดับสูงทีละบรรทัด แล้วแปลเป็นภาษาเครื่องและทำงานทันที แล้วก็จะกลับไปอ่านโค้ดบรรทัดถัดไป ทำอย่างนี้ไปจนจบโปรแกรม วิธีนี้จะต้องทำการอ่านโค้ดและแปลเป็นภาษาเครื่องสลับกันไป ทำให้ทำงานได้ช้ากว่าแบบคอมไพล์เลอร์ แต่ก็มีข้อดีคือตัวอินเทอร์พรีเตอร์สามารถสร้างได้ง่ายกว่าตัวคอมไพเลอร์

ภาษาจาวาได้นำข้อดีของทั้งสองแบบมารวมกัน โดยเมื่อทำการแปลงโค้ด ตัวจาวาคอมไพเลอร์ (Java Compiler) จะทำการแปลโค้ดภาษาจาวาเป็นไบต์โค้ด (Byte Code) ที่ไม่ขึ้นกับแพลตฟอร์ม และเมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำการประมวลผล ตัวจาวาเวอร์ชวลแมชชีน (Java Virtual Machine) จะทำการแปลไบต์โค้ดทีละคำสั่งเป็นภาษาเครื่องของแพลตฟอร์มที่กำลังทำงานแล้วส่งให้ตัวประมวลผลทำงานตามคำสั่ง

วิธีการนี้ทำให้โปรแกรมที่เขียนในภาษาจาวาสามารถทำงานได้ในลักษณะ "เขียนครั้งเดียวใช้งานได้ทุกที่" คือไม่ว่าจะทำการคอมไพล์โค้ดในแพลตฟอร์มใด ก็สามารถนำไบต์โค้ดที่ได้ไปใช้กับแพลตฟอร์มอื่นที่มีตัวจาวาเวอร์ชวลแมชชีนสำหรับใช้ประมวลผล

4.3 การใช้คลาส Vector

อาร์เรย์ (Array) ในภาษาจาวา มีขนาดคงที่ คือเมื่อสร้างขึ้นแล้วจะเปลี่ยนแปลงให้มีพื้นที่สำหรับเก็บสมาชิกมากขึ้นหรือน้อยลงไม่ได้ เราจึงต้องเตรียมอาร์เรย์ ไว้ใหญ่กว่าที่ใช้จริง เป็นการสิ้นเปลืองหน่วยความจำและในบางกรณีก็อาจจะเกิดความผิดพลาดขึ้นได้ ถ้าข้อมูลมีจำนวนมากกว่าที่เตรียมไว้ ภาษาจาวามีคลาส Vector สำหรับเก็บกลุ่มของข้อมูลที่เราไม่ทราบล่วงหน้าว่ามีจำนวนเท่าใด หรือกล่าวอีกอย่างได้ว่าเป็นอาร์เรย์ที่เพิ่มหรือลดขนาดได้อย่างอัตโนมัติขณะที่โปรแกรมทำงาน คลาส Vector มี constructors สามตัว คือ

- Public Vector();

  จะสร้าง vector ที่มีขนาดเริ่มต้นเป็น 10

- Public Vector(int initialCapacity);

  จะสร้าง vector ที่มีขนาดเริ่มต้นเป็น initialCapacity

- Public Vector(int initialCapacity, int capacityIncrement);

  จะสร้าง vector ที่มีขนาดเริ่มต้นเป็น innitialCapacity และจะเปลี่ยนขนาดครั้งละ capacityIncrement

4.4 การใช้คลาส JFileChooser

สามารถแสดงไดอะล็อก (Dialog) เพื่อใช้ในการเลือกไฟล์ได้ โดยประกาศดังนี้

```
JFileChooser fileChooser = new JFileChooser();
fileChooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
int result = fileChooser.showSaveDialog(this);
```

จากโค้ดข้างบนนี้เป็นไดอะล็อกสำหรับบันทึก (save) ไฟล์ โดยแสดงไดอะล็อกเพื่อให้ผู้ใช้สามารถบันทึกไฟล์ได้อย่างเดียว (FILES_ONLY) แต่สามารถเปลี่ยนให้จากบันทึกไฟล์อย่างเดียว ให้บันทึกได้ทั้งไดเรกทอรีอย่างดียว (DIRECTORIES_ONLY) หรือทั้งไฟล์และไดเรกทอรี (FILES_AND_DIRECTORIES)

ถ้าต้องการใช้ไดอะล็อกเพื่อเปิดไฟล์ ให้เปลี่ยนจากเมธอด (Method) showSaveDialog(this); ไปเป็น showOpenDialog(this); แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในไดอะล็อกจะมีปุ่ม OK และ Cancel โดยเราสามารถตรวจสอบการกดปุ่มว่ากดปุ่มไหนได้โดยดูจากตัวแปร (paramcters) ของออบเจกของ JFilcChooscr เช่น ถ้าปุ่ม Cancel ถูกกด จะทำให้ค่าของ JFileChooser.CANCEL_OPTION เป็นจริง (TRUE)

ออบเจกนี้ยังสามารถเอาชื่อไฟล์ที่ถูกเลือกมาได้ด้วย จากเมธอด gctSclcctcdFilc() ซึ่งจะส่งค่ากลับมาเป็นชนิดไฟล์

4.5 การใช้ File Reader/Writer

ภาษาจาวามีคลาส FileReader ซึ่งขยายจากคลาส InputStreamReader สำหรับสร้างสตรีมของการอ่านข้อมูล (input char strcam) และมีคลาส FileWritcr ซึ่งขยายจากคลาส OutputStreamWritcr สำหรับสร้างสตรีมของการเขียนข้อมูล (output char strcam) ซึ่งแสดงตัวอย่างดังรูปที่ 5-1

```
FileReader fin = new FileReader("FileRW2.java");
BufferedReader bin = new BufferedReader(fin);

FileWriter fout = new FileWriter("tmp");
BufferedWriter bout = new BufferedWriter(fout);
PrintWriter pout = new PrintWriter(bout);

String s;
While ((s = bin.readLine()) != null)
  pout.println(s);
bin.close();
pout.close();
```

รูปที่ 4-1 แสดงการสร้างสำหรับ FileReader และ FileWriter

4.6 ทฤษฎีต่างๆ ในการสร้างส่วนติดต่อกับผู้ใช้

เป็นการใช้ Swing Sct ในภาษาจาวา ซึ่งอยู่ในแพ็คเกจ javax.swing ซึ่งมีความสามารถติดต่อกับผู้ใช้ได้อย่างดีในเรื่องส่วนกราฟฟิค 2 มิติ ซึ่งสามารถลบข้อเสียของ Awt sct ได้ ที่ไม่สามารถรองรับการสร้างส่วนติดต่อกับผู้ใช้ที่ซับซ้อนได้

4.6.1 Scroll Pane

ใช้ คลาส JScrollPane สำหรับการสร้าง Scroll Pane คือเป็นคอนเทนเนอร์ ที่สามารถวางคอมโพเนนท์อื่นลงบนตัวมันได้ และถ้าหากคอมโพเนนท์นั้นมีขนาดใหญ่กว่าพื้นที่ของคอนเทนเนอร์ ก็จะมีการสร้าง scroll bars ให้อย่างอัตโนมัติ ทั้งแนวนอนและแนวตั้งเมื่อจำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการใช้งาน Scroll Pane

```
JLabel L = new JLabel(new Image Icon("image/picture.gif"))
JScrollPane ScrollPane1 = new JScrollPane(L);
```

4.6.2 ปุ่ม (Button)

ใช้คลาส JButton สำหรับการสร้างปุ่มขึ้นมาโดยสามารถนำไอคอนรูปภาพหรือข้อความไปใส่ในปุ่มได้ เช่น

```
JButton jButton1 = new JButton(createImageIcon("images/start.gif"));
```

โดยจะมีการเรียก addActionListener เพื่อทำการตรวจสอบว่ามีการกดปุ่มนั้น

```
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        jButton1_actionPerformed(e);
    }
});
```

4.6.3 อิมเมจไอคอน (ImageIcon)

ใน Swing Component สามารถโหลดรูปได้ทุกที่เช่น แอพเพล็ต (applet) เฟรม(Frame) ไดอะล็อก (Dialog) ซึ่งต่างจาก Awt Component ที่โหลดได้ในแอพเพล็ต เท่านั้น มิฉะนั้นจะต้องไปเรียกใช้ฟังก์ชันในคลาส Toolkit โดยอิมเมจไอคอนสามารถโหลดรูปจากชื่อไฟล์เลยได้

```
ImageIcon Start = createImageIcon("images/start.gif");
```

4.6.4 Even Listener Interfaces

ทุกคลาสสำหรับ เหตุการณ์ (event) แต่ละประเภทจะมี event listener interface ซึ่งกำหนดฟังก์ชันที่โปรแกรม listener จะต้องทำการประกาศไว้เพื่อรับเหตุการณ์ประเภทนั้น ทุก event listener interface อยู่ในแพ็คเกจ java.awt.event

4.6.4.1 KeyListener

เป็นการติดต่อที่ listener ต้องทำการ implements เพื่อจัดการกับ เหตุการณ์กดคีย์บอร์ด (key event) เมื่อ listener ทำการ KeyListener และ addKeyListener() แล้ว หากมีเหตุการณ์กดคีย์บอร์ดเกิดขึ้น ถ้ามีการกดคีย์บอร์ดลงไป จะส่งไปที่ฟังก์ชัน keyPressed()

```
public void keyPressed( KeyEvent event ){}
```

จะเก็บค่าตัวที่ถูกกดลงใน e.getKeyChar( ) ตามรหัสแอสกี้(ascii) เช่นกดปุ่มdelete จะมีค่าเป็น 127 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.4.2 MouseListener

เป็นคลาสที่รับเหตุการณ์ที่เกิดจากการใช้เมาส์(mouse events) จะมีการเก็บว่าเป็นเหตุการณ์ประเภทใด เกิดที่ตำแหน่งใด เวลาใด มีการกดเม้าส์กี่ครั้ง และเหตุการณ์ทำให้เกิดเมนูป๊อปอัพ(pop-up menu) หรือไม่ มีเหตุการณ์ที่เกิดจากการใช้เมาส์อยู่ทั้งหมด 7 แบบ คือ

- MOUSE_CLICKED เมื่อปุ่มเม้าส์ถูกกดและถูกปล่อย
- MOUSE_DRAGGED เมื่อปุ่มเม้าส์ถูกกดและถูกลาก
- MOUSE_ENTERED เมื่อเม้าส์เข้าไปในขอบเขตของคอมโพเนนท์
- MOUSE_EXITED เมื่อเม้าส์ออกไปนอกขอบเขตของคอมโพเนนท์
- MOUSE_MOVED เมื่อเม้าส์เคลื่อนที่
- MOUSE_PRESSED เมื่อปุ่มเม้าส์ถูกกด
- MOUSE_RELEASED เมื่อปุ่มเม้าส์ถูกปล่อย

ซึ่งแต่ละเหตุการณ์ก็จะถูกส่งไปยังแต่ละฟังก์ชัน เช่น

```
public void mouseClicked( MouseEvent event ){}
```

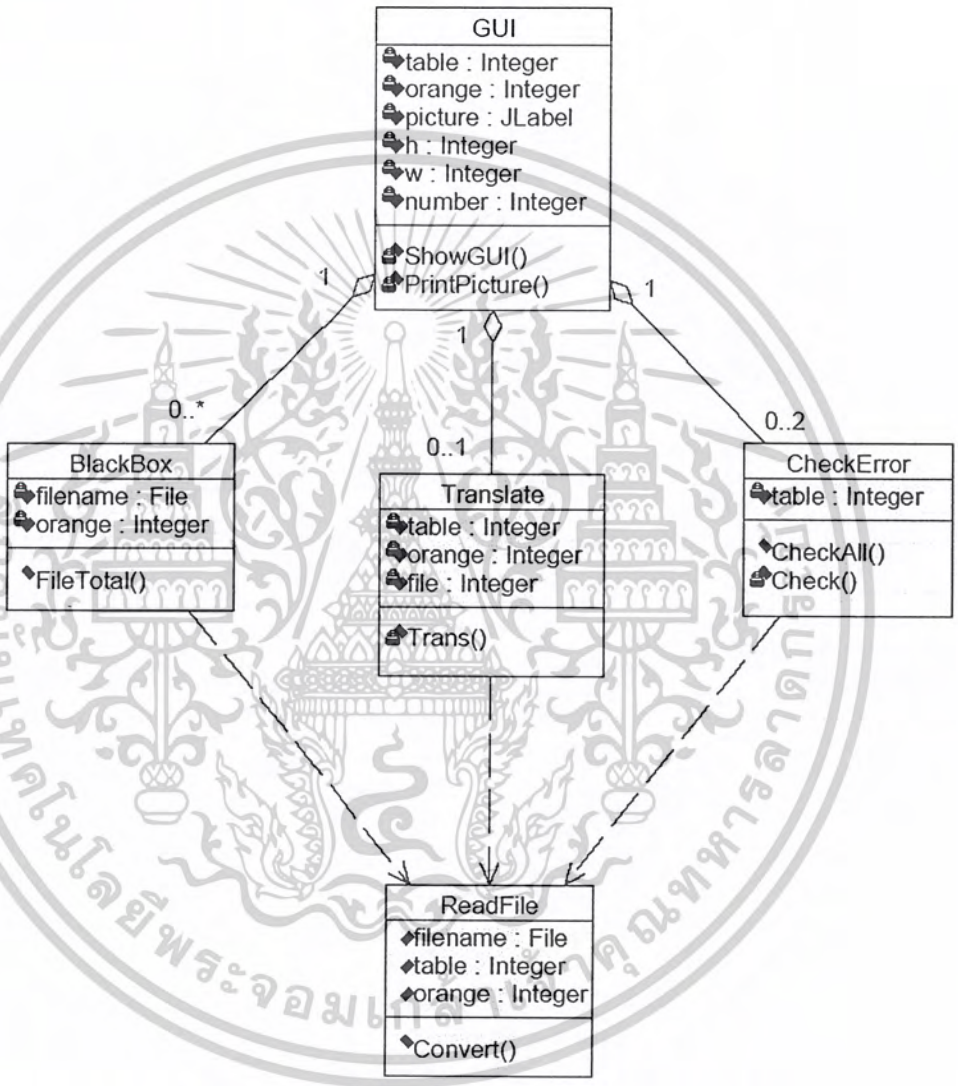
โดยมีการเก็บตำแหน่งที่เม้าส์มีการกระทำขณะนั้นโดยมีการเรียกขึ้นมาโดยใช้คำสั่ง event.getX( ) และ event.getY( )

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การออกแบบไดอะแกรม

5.1 คลาสไดอะแกรม



รูปที่ 5-1 คลาสไดอะแกรมของโครงงานภาษาคอมพิวเตอร์สำหรับเด็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5-1 ประกอบด้วย คลาสต่างๆดังนี้

5.1.1 คลาส GUI

ทำหน้าที่ติดต่อกับผู้ใช้ผ่านทางหน้าจอคอมพิวเตอร์ และเม้าส์เป็นหลัก เก็บค่าตารางชนิดของรูปและเก็บวัตถุรูปภาพเพื่อแสดงผลให้ถูกตำแหน่ง และรับคำสั่งของผู้ใช้จากเม้าส์ในการเลือกรูปและเคลื่อนย้ายรูป และทำหน้าที่แสดงผลการทำงานของภาษาโปรแกรมที่ได้เขียน

5.1.2 คลาส BlackBox

คือคลาสของฟังก์ชันที่ไม่เห็นการทำงาน ทำหน้าที่คำนวณเพื่อหาผลลัพธ์ในฟังก์ชันที่ไม่เห็นการทำงาน โดยมีการรับค่าจำนวนค่าตัวแปรก่อนทำฟังก์ชัน และมีการเรียกคลาส ReadFile เพื่ออ่านค่าไฟล์ที่ต้องการหาค่า เพื่อทำการคำนวณ โดยฟังก์ชัน FindTotal() แล้วทำการส่งค่าผลลัพธ์กลับออกมา

5.1.3 คลาส CheckError

ทำหน้าที่รับค่าไฟล์ตารางเข้ามา เพื่อที่จะทำการแปลงไฟล์เป็นตารางด้วยคลาส ReadFile แล้วเก็บค่าเป็นตารางลงในคลาสนี้ จะมีฟังก์ชันทำการตรวจสอบรูปแบบเพื่อนำตารางมาหาว่าวางรูปแบบถูกต้องหรือไม่ ถ้าไม่ถูกจะมีการส่งตำแหน่งช่องที่วางไม่ถูกต้องกลับออกมา

5.1.4 คลาส Translate

ทำหน้าที่เก็บค่าแปลงรูปแบบคำสั่งจากภาษารูปภาพ เป็นภาษาซี ในฟังก์ชัน Tran() โดยส่งค่าตัวแปรต่างๆที่จำเป็นจากคลาส GUI เพื่อแปลงเป็นภาษาซี

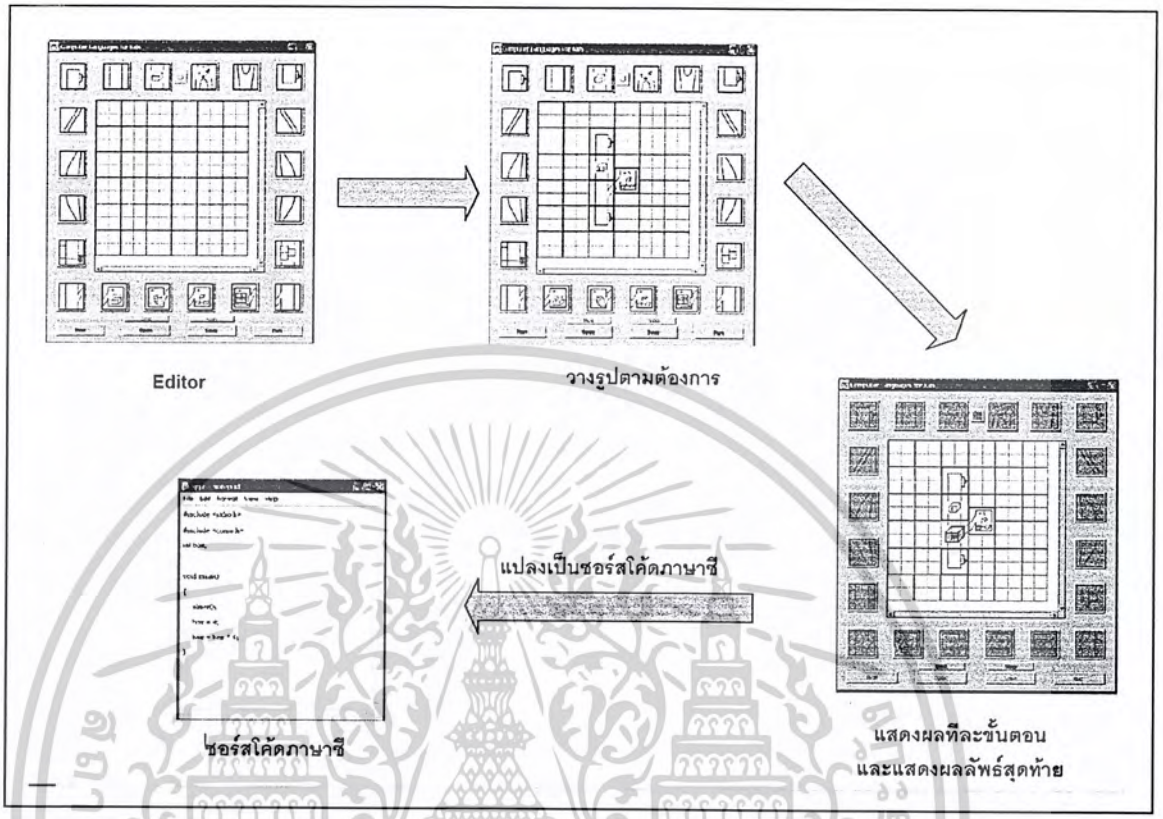
5.1.5 คลาส ReadFile

ทำหน้าที่หน้าที่อ่านไฟล์จากไฟล์ที่ได้ทำการบันทึกไว้ในโปรแกรมภาษาคอมพิวเตอร์สำหรับเด็ก แล้วมีฟังก์ชันแปลงเป็นค่าตัวแปรต่างๆ ที่มีการเก็บตำแหน่งรูปภาพและตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ท็อปดาวน์ไดอะแกรม (Top Down Diagram)

ท็อปดาวน์ไดอะแกรมของโครงงานภาษาคอมพิวเตอร์สำหรับเด็กเป็นดังรูปที่ 5-2

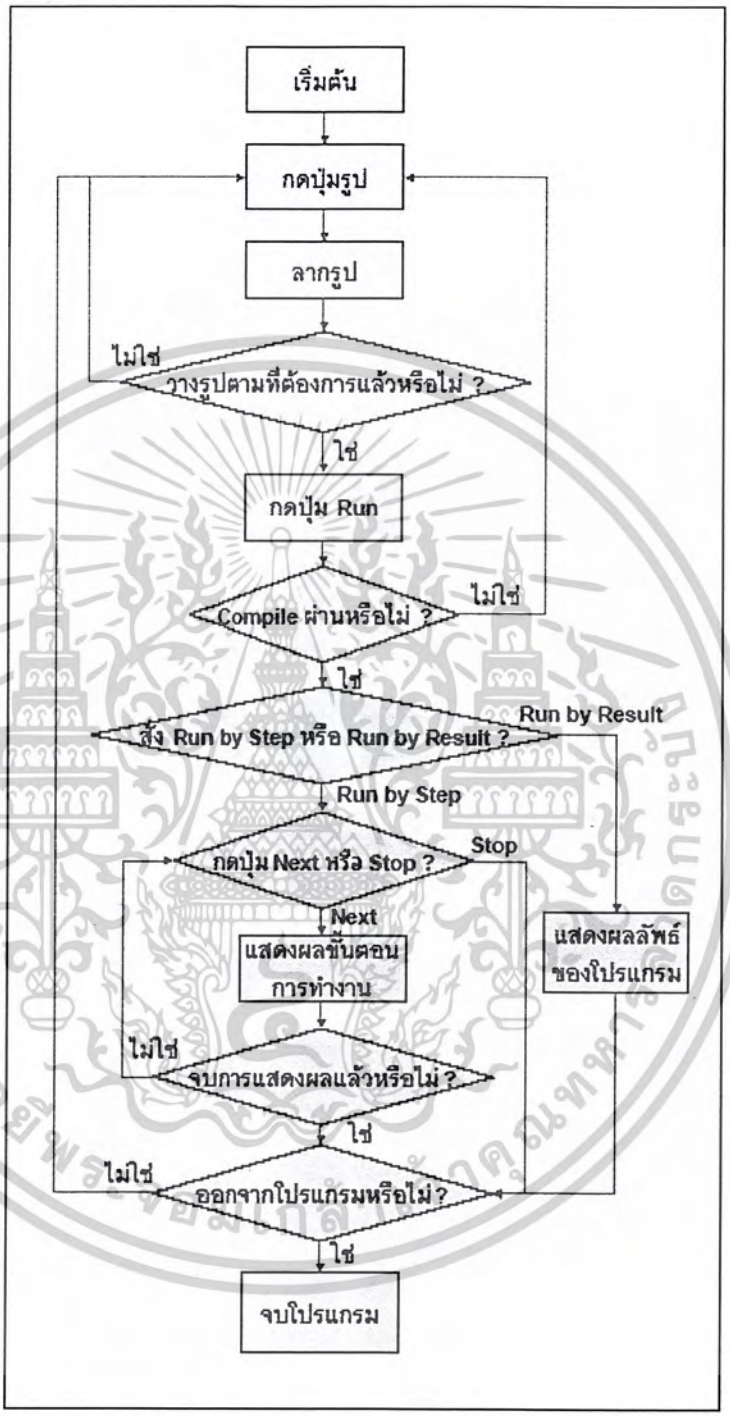


รูปที่ 5-2 ท็อปดาวน์ไดอะแกรมของโครงงานภาษาคอมพิวเตอร์สำหรับเด็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 โฟล์วชาร์ทไดอะแกรม (Flow Chart Diagram)

เมื่อเริ่มเปิดโปรแกรมภาษาคอมพิวเตอร์สำหรับเด็ก จะมีลำดับการทำงานดังรูปที่ 5-3



รูปที่ 5-3 โฟล์วชาร์ทไดอะแกรมของโครงงานภาษาคอมพิวเตอร์สำหรับเด็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

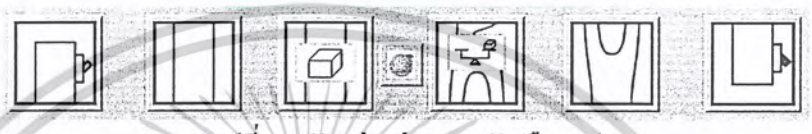
บทที่ 6
การพัฒนาโปรแกรม

6.1 ส่วนที่ติดต่อกับผู้ใช้ (Graphic User Interface, GUI)

ในส่วนนี้จะแบ่งออกเป็น 3 ส่วนใหญ่ๆ คือ

6.1.1 ส่วนสำหรับเลือกรูป

ในส่วนนี้จะมีทั้งหมด 21 รูป เพื่อใช้ในการเขียนโปรแกรมต่างๆ ขึ้นมา ดังตัวอย่างในรูปที่ 6-1



รูปที่ 6-1 ตัวอย่างส่วนสำหรับเลือกรูป

การออกแบบ Object นี้จะออกแบบตามพื้นฐานของการเขียนโปรแกรม ซึ่งจะแบ่งออกได้ดังนี้

- จุดเริ่มต้นและจุดสิ้นสุด จะใช้สัญลักษณ์รูปสวิตช์เปิด(สีเขียว) และสวิตช์ปิด(สีแดง) ดังรูปที่ 6-2 และรูปที่ 7-3

รูปที่ 6-2 จุดเริ่มต้นโปรแกรม

รูปที่ 6-3 จุดสิ้นสุดโปรแกรม

- ตัวแปร ในที่นี้จะให้ตัวแปรทุกตัวเป็นชนิด Integer โดยจะแทนด้วยรูปกล่องเปล่าบนสายพาน โดยมีค่าเริ่มต้นเป็นศูนย์ ดังรูปที่ 6-4

รูปที่ 6-4 ตัวแปร

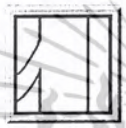
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61382

- คำสั่งคำนวณพื้นฐาน จะใช้สัญลักษณ์ดังนี้
  - การบวก จะใช้ร่วมกันสองรูปเสมอคือ รูปที่ 6-5 และรูปที่ 6-6

รูปที่ 6-5 ใช้ร่วมกับการบวก

รูปที่ 6-6 สัญลักษณ์บวก

  - การลบ จะใช้ร่วมกันสองรูปเสมอคือ รูปที่ 6-7 และรูปที่ 6-8



รูปที่ 6-7 ใช้ร่วมกับการลบ

รูปที่ 6-8 สัญลักษณ์ลบ

  - การคูณ จะใช้ร่วมกันสองรูปเสมอคือ รูปที่ 6-9 และรูปที่ 6-10

รูปที่ 6-9 ใช้ร่วมกับการคูณ

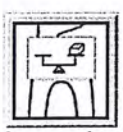
รูปที่ 6-10 สัญลักษณ์คูณ

  - การหาร จะใช้ร่วมกันสองรูปเสมอคือ รูปที่ 6-11 และรูปที่ 6-12

รูปที่ 6-11 ใช้ร่วมกับการหาร

รูปที่ 6-12 สัญลักษณ์หาร

- คำสั่งเงื่อนไข จะเปรียบเทียบกับค่าคงที่ค่าหนึ่ง โดยถ้ามีค่ามากกว่าหรือเท่ากัน จะไปทางซ้าย และถ้ามีค่าน้อยกว่าไปทางขวา ดังรูปที่ 6-13 และจบเงื่อนไขโดยใช้รูปที่ 6-14

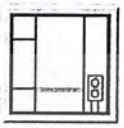


รูปที่ 6-13 เงื่อนไข

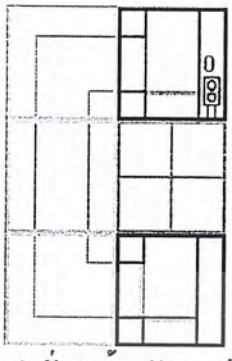
รูปที่ 6-14 จบเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คำสั่งทำซ้ำ เป็นการกำหนดการวนทำซ้ำ โดยกำหนดจำนวนรอบในการทำ ดังรูปที่ 6-15 และเมื่อเลือกแล้วจะเป็นดังรูปที่ 6-16 ซึ่งสามารถขยายความยาวเพิ่มได้



รูปที่ 6-15 คำสั่งทำซ้ำ



รูปที่ 6-16 คำสั่งทำซ้ำหลังจากเลือกแล้ว

- ฟังก์ชันที่มองไม่เห็นการทำงาน (Blackbox Function) ดังรูปที่ 6-17 โดยสามารถเลือกฟังก์ชันที่เคยเขียนไว้แล้ว เข้ามาใช้ในโปรแกรมได้ ซึ่งในการทำงานจริง จะไม่เห็นกระบวนการทำงานภายในของฟังก์ชันนี้

รูปที่ 6-17 ฟังก์ชันที่มองไม่เห็นการทำงาน

- การเพิ่มค่า โดยมีลักษณะเป็นรูปส้ม ดังรูปที่ 6-18 โดยนำส้มไปใส่ในรูปที่ต้องการเพิ่มค่า

รูปที่ 6-18 ส้มที่ใช้เพิ่มค่า

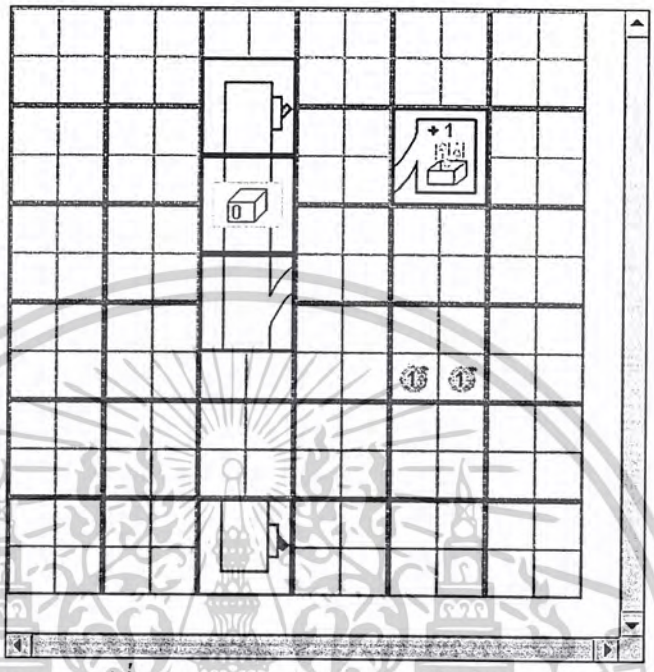
- เส้นทางของการทำงาน จะมีทั้งหมด 7 แบบ คือ เส้นทางตรงธรรมดา เส้นทางจากขวาไปซ้าย เส้นทางจากซ้ายไปขวา เส้นทางจากขวามาตรงกลาง เส้นทางจากซ้ายมาตรงกลาง เส้นทางจากตรงกลางไปขวา และเส้นทางจากตรงกลางไปซ้าย ดังรูปที่ 6-19 ตามลำดับ

รูปที่ 6-19 เส้นทางการทำงานทั้งหมด 7 แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.2 ส่วนสำหรับวางรูป

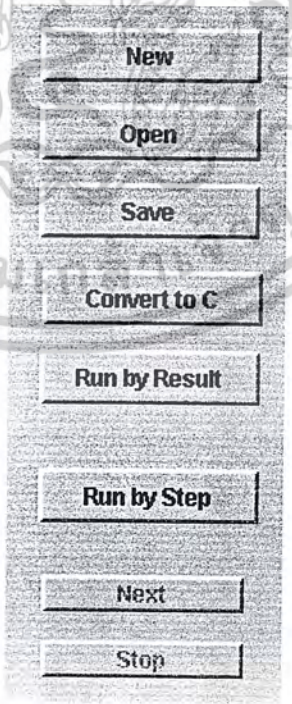
เมื่อกดเลือกรูปในหัวข้อที่ผ่านมาแล้ว รูปนั้นจะปรากฏอยู่ในส่วนสำหรับวางรูป ดังรูปที่ 6-20 โดยจะกำหนดค่าปกติที่ตำแหน่งล่าสุดที่เคยกดเม้าส์ หรือตำแหน่งใกล้เคียง อีกทั้งยังสามารถเคลื่อนย้ายรูปได้ตามต้องการด้วย



รูปที่ 6-20 แสดงภาพภายในส่วนสำหรับวางรูป

6.1.3 ส่วนปุ่มต่างๆ

มีทั้งหมด 8 ปุ่มดังรูปที่ 6-21 ได้แก่



รูปที่ 6-21 ปุ่มต่างๆ 8 ปุ่มภายในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปุ่ม New เป็นปุ่มที่ใช้เปิดส่วนสำหรับวางรูปใหม่
- ปุ่ม Open เป็นปุ่มที่ใช้เพื่อเปิดไฟล์เดิมเพื่อใช้งาน หรือแก้ไข
- ปุ่ม Save เป็นปุ่มที่ใช้เพื่อเก็บบันทึกไฟล์ตารางของโปรแกรม
- ปุ่ม Concert to C เป็นปุ่มที่ใช้แปลงรูปภาพในส่วนวางรูปเป็นภาษาซี
- ปุ่ม Run by Result เป็นปุ่มที่ทำการคอมไพล์เพื่อตรวจสอบรูปแบบของคำสั่ง และทำการแสดงผลลัพธ์ของโปรแกรมทันที่
- ปุ่ม Run by Step เป็นปุ่มที่ทำการคอมไพล์เพื่อตรวจสอบรูปแบบของคำสั่ง และสั่งทำการรันโปรแกรมทีละขั้นตอน
- ปุ่ม Next เป็นปุ่มที่ใช้ได้เมื่อผ่านการตรวจสอบรูปแบบของคำสั่งว่าถูกต้องแล้วด้วยการกดปุ่ม Run by Step ก่อน โดยใช้ในการแสดงผลทีละขั้นตอน
- ปุ่ม Stop เป็นปุ่มที่ใช้ได้เมื่อผ่านการตรวจสอบรูปแบบของคำสั่งว่าถูกต้องแล้วด้วยการกดปุ่ม Run by Step ก่อน โดยใช้ในการสิ้นสุดการแสดงผลทีละขั้นตอน

6.2 การแปลงจากรูปภาพเป็นซอร์สโค้ด

ในการแปลงจากรูปภาพเป็นซอร์สโค้ดนั้น จะต้องอาศัยคลาส Translate ในการแปลง ซึ่งจะต้องส่งตารางที่เก็บค่าของรูปภาพทั้งหมด ค่าของตัวแปร ค่าของไฟล์ต่างๆ ให้กับออบเจกของคลาส Translate

สำหรับคลาส Translate จะประกอบด้วย 2 ส่วนคือ การแปลงไฟล์ธรรมดา และการแปลงไฟล์ของฟังก์ชัน ซึ่งทั้งสองส่วนนี้อาศัยอัลกอริทึม ฟังก์ชันเรียกตัวเอง (Recursion) ซึ่งมีลักษณะการทำงานดังนี้

เมื่อรับค่าต่างๆ จากคลาส GUI แล้ว จะเปรียบเทียบค่าในตารางที่ได้มากับรหัสของรูปว่าแต่ละช่องมีรูปอะไรอยู่บ้าง และรูปนั้นจะแปลงเป็นโค้ดอะไร โดยจะเก็บโค้ดที่แปลงแล้วลงในตัวแปรที่เป็น vector ทีละตัวๆ ตามลำดับของการเขียนโปรแกรม และเมื่อแปลงจนครบแล้ว จึงทำการเขียนลงไฟล์เป็นนามสกุล .c

ในกรณีที่มีฟังก์ชัน คลาส Translate จะเปิดไฟล์ของฟังก์ชันเพื่อเปรียบเทียบค่าในตารางและแปลงเป็นโค้ดเป็นไฟล์ปกติ แต่จะทำแปลงไฟล์ของฟังก์ชันให้เสร็จก่อนจะทำส่วนไฟล์ปกติต่อไป ดังรูปที่ 6-22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <conio.h>
int box;

void c();

void main()
{
    clrscr();
    box = 1;
    box = box + 2;
    c();
}

void c()
{
    box = box + 6;
    box = box - 2;
}
```

รูปที่ 6-22 ตัวอย่างซอร์สโค้ดหลังจากแปลงเสร็จแล้ว

6.3 การเก็บค่าตัวแปรบนส่วนวางรูปเพื่อแสดงผล

เมื่อมีการกดปุ่มเพื่อเพิ่มรูป จะมีรูปมาวางเพิ่มในส่วนการวางรูป โดยจะมีการเก็บค่าต่างๆ เพื่อจะได้รู้ตำแหน่งที่แน่นอนของรูปนั้นได้ ว่ารูปอยู่ตำแหน่งใดและมีค่าตัวแปรเท่าไหร่ ทำให้สามารถทำการย้ายตำแหน่ง หรือลบรูปที่ไม่ต้องการออกไปได้ และสามารถตรวจสอบตำแหน่งและความผิดพลาดของรูปแบบคำสั่ง อีกทั้งยังบันทึกลงไฟล์ และอ่านจากไฟล์ที่บันทึกได้อย่างถูกต้อง โดยมีการเก็บค่าตัวแปรที่สำคัญๆ ดังนี้

6.3.1 การเก็บตำแหน่งรูป

จะเก็บค่าเป็นชนิดตัวเลข เป็นตาราง (array) 2 มิติ ตามความกว้างและความสูงตามหน้าจอวางรูป โดยจะเก็บเป็นชนิดของรูป ซึ่งมีการกำหนดก่อนไว้แล้วว่าหมายเลขใดเป็นรูปชนิดอะไร เมื่อมีการย้ายตำแหน่งรูปจะมีการย้ายค่าหมายเลขจากตารางช่องเดิมไปไว้ในตารางช่องที่ย้ายไป

6.3.2 การเก็บค่าจำนวนตัวแปร

จะเก็บค่าเป็นชนิดตัวเลข เป็นตาราง 2 มิติ เช่นเดียวกับการเก็บตำแหน่งรูป แต่จะเก็บค่าเป็นจำนวนตัวแปรที่ได้ใส่เข้าไปในรูปที่สามารถใส่ค่าตัวแปรได้ เพื่อจะได้รู้ว่าช่องนั้นจะทำการบวกด้วยค่าเท่าไหร่ เป็นต้น โดยช่องอื่นๆ จะถูกกำหนดค่าไว้เป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.3 การเก็บค่าตำแหน่งการทำซ้ำ

โดยเก็บตำแหน่งเริ่มต้นการทำซ้ำ และสิ้นสุดการทำซ้ำแยกกันเก็บลงไปในตารางเก็บตำแหน่งรูปในข้อ 7.3.1 แต่จะมีการเก็บตำแหน่งรูปเริ่มต้นการทำซ้ำกับตำแหน่งรูปสิ้นสุดการทำซ้ำที่เป็นคู่กัน เมื่อทำการย้ายหรือลบรูป จะสามารถทำการย้ายหรือลบทั้งสองตัวได้พร้อมกัน

6.3.4 การเก็บชื่อไฟล์ที่เป็นฟังก์ชัน

เมื่อมีการวางรูปฟังก์ชันเพื่อเรียกใช้ฟังก์ชันที่เคยเขียนมาก่อน โดยจะเก็บค่าเป็นชนิดไฟล์ซึ่งเป็นตาราง 2 มิติ เช่นเดียวกัน ช่องใดที่เป็นรูปฟังก์ชัน จะเก็บค่าตำแหน่งและชื่อรูปเอาไว้ แต่ส่วนอื่นๆ จะไม่มีการเก็บชื่อไฟล์เอาไว้

เมื่อมีการวางรูปเพิ่มเข้าไป ตารางจะเก็บชนิดรูปที่เพิ่มเข้าไปในตาราง ตามตำแหน่งที่วางไว้ในส่วนวางรูป

6.4 การบันทึกและการอ่านไฟล์

6.4.1 การบันทึกลงไฟล์

เมื่อต้องการเก็บค่ารูปที่แสดงอยู่บนส่วนวางรูป จะมีการเขียนลงไฟล์ เป็นชนิดไฟล์ข้อความ (Text File) ซึ่งเก็บความกว้าง ความสูงของหน้าจอวางรูป เก็บตารางตำแหน่งรูปว่าตำแหน่งใดเก็บรูปอะไร และเก็บค่าตัวแปรว่าในตำแหน่งนั้นมีค่าเท่าไหร่ เก็บค่าว่าตำแหน่งเริ่มต้นลูป (loop) และตำแหน่งสิ้นสุดลูปตำแหน่งใดที่คู่กัน และเก็บชื่อไฟล์ ที่ทำการเรียกใช้ฟังก์ชันในไฟล์นั้น โดยที่สามารถกำหนดตำแหน่งและชื่อไฟล์ที่ต้องการบันทึกได้

6.4.2 การอ่านค่าจากไฟล์

เมื่อต้องการเปิดไฟล์ที่ได้ทำการบันทึกไว้ก่อนหน้านี้ จะมีการเลือกว่าต้องการเปิดไฟล์ใดขึ้นมา หลังจากเลือกแล้ว เริ่มต้นจะมีการอ่านไฟล์กำหนดความกว้างยาวของตารางของส่วนวางรูปเพื่อนำมาสร้างเป็นตารางชนิดต่างๆ ตาม 6.3 และนำค่าที่อ่าน มาเก็บลงในตารางต่างๆ แล้วทำการวางตำแหน่งรูปต่างๆ ให้ตรงกับไฟล์ที่อ่านมา นับจำนวนรูปทั้งหมด แล้วมีการแปลงเป็นตัวแปรอื่นๆ ให้ตรงกับที่ในคลาสหลักต้องการ

6.5 การตรวจสอบรูปแบบคำสั่ง

ก่อนที่จะทำการรันจะมีการตรวจสอบรูปแบบคำสั่งก่อน ว่ามีการวางรูปแบบคำสั่งถูกต้องหรือไม่ รูปแบบคำสั่งที่ถูกต้อง ต้องมีเงื่อนไขดังต่อไปนี้

- รูปที่วางต้องติดต่อกันทั้งหมด ไม่มีบิดเบี้ยว ไม่มีส่วนขาด และไม่มีส่วนเกิน
- มีรูปจุดเริ่มต้นโปรแกรม และรูปจุดสิ้นสุดโปรแกรม อย่างละหนึ่งเท่านั้นในแต่ละโปรแกรม
- หากมีการคำนวณหรือเปรียบเทียบ ต้องมีการประกาศตัวแปรก่อนเสมอ
- ไม่มีการหารด้วยศูนย์
- ไม่มีการเรียกฟังก์ชันซ้ำไปซ้ำมาแบบไม่รู้จบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไฟล์ที่เรียกใช้เป็นฟังก์ชันนั้นต้องมีอยู่จริง และสามารถอ่านไฟล์นั้นได้
- เมื่อมีรูปเริ่มต้นเงื่อนไขต้องมีรูปสิ้นสุดเงื่อนไขด้วยควบคู่กันโดยที่ไม่มีการรวมเงื่อนไขไม่ถูกคู่

การตรวจสอบรูปแบบคำสั่ง เป็นการส่งตารางรูปภาพเข้าไป ถ้ามีการวางรูปแบบคำสั่งไม่ตรงตามรูปแบบตามเงื่อนไขข้างต้น จะมีการส่งค่าตำแหน่งที่ไม่ถูกต้องนั้นออกมาเพื่อแสดงผล โดยถ้าวางถูกตำแหน่งจะตรวจสอบไปเรื่อยๆ เป็นการเรียกตัวเอง โดยเมื่อพบรูปต่างๆจะทำการตรวจสอบดังนี้

- เริ่มแรกจะทำการไล่ตารางทีละบรรทัดตั้งแต่จุดซ้ายบนสุดของตารางไปทางขวาจนจบบรรทัดจึงขึ้นบรรทัดใหม่ จนกระทั่งพบจุดเริ่มต้นโปรแกรม ถ้าพบรูปอื่นก่อนจะส่งค่าตำแหน่งรูปนั้นกลับออกมานั้น
- เมื่อพบจุดเริ่มต้นโปรแกรมแล้วจึงทำการตรวจสอบตำแหน่งล่างของจุดเริ่มต้นโปรแกรมหนึ่งรูป ถ้ามีรูป และรูปตรงตามชนิดที่กำหนดจะทำการตรวจสอบต่อไปเรื่อยๆ เป็นฟังก์ชันเรียกตัวเอง
- ถ้ามีการวางรูปไม่ถูกต้อง จะมีการส่งค่าตำแหน่งที่ไม่ถูกกลับออกมาและแจ้งว่ารูปตำแหน่งนั้นที่กำลังตรวจสอบอยู่วางรูปไม่ถูกต้อง
- เมื่อพบรูปที่ใช้คู่กับการบวกหรือการคูณ จะตรวจสอบก่อนว่ามีรูปประกาศตัวแปรก่อนหน้านี้หรือไม่แล้วตรวจสอบตำแหน่งขวาบนของตำแหน่งนั้นว่าเป็นรูปคำสั่งบวกหรือคำสั่งคูณหรือไม่ ถ้าไม่ใช่ส่งค่าตำแหน่งรูปที่ใช้คู่กับการบวกหรือการคูณนั้นกลับเพื่อแจ้งข้อผิดพลาด ถ้าใช่จะตรวจสอบตำแหน่งข้างล่างตำแหน่งรูปที่ใช้คู่กับการบวกหรือการคูณต่อไป
- ถ้าพบรูปที่ใช้คู่กับการหารหรือการลบ จะทำเช่นเดียวกับเมื่อพบรูปที่ใช้คู่กับการบวกหรือการคูณ แต่จะตรวจสอบว่าตำแหน่งซ้ายล่างของตำแหน่งนั้นแทน
- ถ้าพบคำสั่งเงื่อนไข จะทำการตรวจสอบก่อนว่ามีรูปประกาศตัวแปรก่อนหน้านี้หรือไม่ จากนั้นทำการตรวจสอบด้านซ้ายของเงื่อนไขก่อน เมื่อพบคำสั่งสิ้นสุดเงื่อนไขจะทำการส่งค่าตำแหน่งสิ้นสุดเงื่อนไขกลับ แล้วทำการตรวจสอบทางด้านขวาของเงื่อนไขต่อจนถึงตำแหน่งสิ้นสุดเงื่อนไขก็จะทำการส่งค่าตำแหน่งสิ้นสุดเงื่อนไขกลับเช่นเดียวกัน นำค่าตำแหน่งสิ้นสุดเงื่อนไขทั้งสองมาเปรียบเทียบกัน ถ้าจุดตำแหน่งสิ้นสุดเงื่อนไขเป็นตำแหน่งเดียวกันแสดงว่าถูกต้องจึงทำการตรวจสอบตำแหน่งข้างล่างตำแหน่งสิ้นสุดเงื่อนไขต่อไป แต่ถ้าไม่ใช่ตำแหน่งเดียวกันแสดงว่าเกิดข้อผิดพลาด ในกรณีนี้จะส่งค่าตำแหน่งจุดเริ่มต้นเงื่อนไขที่ไม่ถูกต้องกลับไป
- ถ้าพบคำสั่งเริ่มการทำซ้ำ จะมีการเพิ่มค่าจำนวนการทำซ้ำ ถ้ามีการพบจุดสิ้นสุดการทำซ้ำจะลดค่าจำนวนการทำซ้ำลง ถ้าลดแล้วได้ค่าลบ แสดงว่ามีคำสั่งสิ้นสุดการทำซ้ำแต่ไม่มีคำสั่งเริ่มต้นการทำซ้ำ จึงส่งค่าตำแหน่งสิ้นสุดการทำซ้ำนั้นกลับ
- ถ้าพบรูปการเรียกใช้ฟังก์ชันจะทำการอ่านไฟล์นั้นแปลงเป็นตารางแล้วทำการตรวจสอบตารางตำแหน่งรูปนั้น ถ้าถูกจะส่งค่าบอกกลับมาว่าฟังก์ชันข้างในนั้นทำงานถูกต้องแล้วจึงตรวจสอบตำแหน่งข้างล่างรูปการเรียกใช้ฟังก์ชันต่อไป แต่ถ้าฟังก์ชันนั้นรูปแบบคำสั่งไม่ถูกต้องจะมีการส่งกลับออกมาเป็นตำแหน่งที่ผิด แล้วจึงทำการส่งค่าตำแหน่งรูปการเรียกใช้ฟังก์ชันที่ไม่ถูกต้องกลับออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้าพบตำแหน่งสิ้นสุดโปรแกรม โปรแกรมจะทำการตรวจสอบดูก่อนว่ามีค่าจำนวนการทำซ้ำเป็นศูนย์หรือไม่ ถ้าเป็นศูนย์แสดงว่าจำนวนเริ่มต้นและสิ้นสุดการทำซ้ำเท่ากัน ถ้าไม่เท่าจะทำการส่งค่าตำแหน่งสิ้นสุดโปรแกรมกลับว่าไม่ถูกต้อง
- สุดท้ายจะตรวจสอบว่ามีตัวอื่นนอกเหนือจากตัวที่เรียกตรวจสอบไปแล้วหรือไม่ ถ้าพบจะทำการส่งค่าตำแหน่งที่เกินมา กลับไปว่าไม่ถูกต้อง แต่ถ้าไม่มีแสดงว่า รูปแบบคำสั่งนั้นถูกต้องทั้งหมด

ถ้ารูปแบบคำสั่งนั้นถูกต้องหมด จึงจะอนุญาตให้ทำการรันโปรแกรม หรือทำการแปลงไฟล์เป็นภาษาซีได้

6.6 การเรียกใช้ฟังก์ชันที่มองไม่เห็นการทำงาน

6.6.1 การเตรียมนำฟังก์ชันที่มองไม่เห็นการทำงานขึ้นมาเพื่อใช้งาน

ทำได้โดยต้องมีการสร้างและบันทึกฟังก์ชันที่มองไม่เห็นการทำงานให้เป็นไฟล์ภาษารูปภาพตามปกติ เมื่อต้องการใช้งานฟังก์ชันที่มองไม่เห็นการทำงานนั้น ให้เลือกปุ่มสร้างฟังก์ชันแล้วเลือกไฟล์ที่ได้ทำการบันทึกนั้น ผลลัพธ์จะได้รูปฟังก์ชันที่มองไม่เห็นการทำงานซึ่งมีชื่อเขียนไว้ที่รูป เป็นชื่อเดียวกับไฟล์ ผู้ใช้สามารถนำรูปไปต่อกับรูปอื่นได้ตามปกติ

6.6.2 การเรียกใช้ฟังก์ชันที่มองไม่เห็นการทำงาน

เป็นฟังก์ชันที่ถูกเรียกใช้โดยฟังก์ชันหลัก หรืออาจถูกเรียกใช้ด้วยตัวฟังก์ชันอื่นก็ได้ ฟังก์ชันเหล่านี้จะถูกเรียกใช้ 2 กรณี คือ เมื่อมีการตรวจรูปแบบคำสั่งจะมีการตรวจไปถึงในฟังก์ชันนี้ด้วย และอีกกรณีคือ เมื่อมีการรันโปรแกรม จะมีการหาค่าผลลัพธ์ของโปรแกรมที่มีฟังก์ชันนี้อยู่

- การเรียกใช้ฟังก์ชันขณะตรวจสอบรูปแบบคำสั่ง

  จะมีหลักการทำงานคือ เมื่อตรวจสอบรูปแบบคำสั่งจนถึงรูปฟังก์ชัน จะทำการหาว่าฟังก์ชันนั้นคือไฟล์ใด แล้วทำการอ่านไฟล์นั้นแปลงเป็นตาราง จากนั้นทำการตรวจสอบตารางตำแหน่งรูปนั้นตามการตรวจสอบรูปแบบคำสั่งต่อไป
- การเรียกใช้ฟังก์ชันขณะรันโปรแกรม

  จะคล้ายกับการเรียกใช้ฟังก์ชันขณะตรวจสอบรูปแบบคำสั่ง แต่จะต่างกันตรงที่การเรียกใช้ฟังก์ชันขณะรันโปรแกรม จะรับค่าตัวแปรเริ่มต้นของฟังก์ชันนี้เข้ามา เพื่อให้ฟังก์ชันนั้นทำงาน แล้วส่งค่าผลลัพธ์ของการทำงานนั้นกลับออกมา โดยไม่เห็นการทำงานข้างในฟังก์ชัน ผู้ใช้สามารถนำค่าผลลัพธ์นั้นไปรันโปรแกรมทีละขั้นตอนต่อได้

6.7 การรันโปรแกรม

การรันโปรแกรมจะเกิดขึ้นหลังจากเมื่อมีการตรวจสอบรูปแบบคำสั่งเรียบร้อยแล้ว เพื่อจะให้เด็กได้รู้ว่าโปรแกรมที่เขียนได้ผลลัพธ์ตามที่ต้องการหรือไม่ สามารถเลือกที่จะแสดงผลได้ 2 แบบ คือ ให้รันแล้วแสดงผลลัพธ์ของโปรแกรมโดยอัตโนมัติ หรือ แสดงผลทีละขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7.1 การแสดงผลทีละขั้นตอนการทำงานของโปรแกรม

มีหลักการทำงานคือ เริ่มต้นจะทำการหาจุดเริ่มต้นโปรแกรมและไล่ไปทีละขั้นตอน โดยจะต้องกดปุ่มให้แสดงผลทีละรูปที่โปรแกรมได้มีการทำงานผ่าน และสามารถจะสั่งให้ออกจากการแสดงผลทีละขั้นตอนได้ตามที่ต้องการ มีขั้นตอนการทำงานโดยมีการแสดงว่าขณะนี้ทำงานอยู่ที่ส่วนใด และค่าตัวแปรขณะนั้นมีค่าเท่าไหร่ ซึ่งทำแต่ละขั้นตอนดังนี้

- หารูปจุดเริ่มต้นของโปรแกรม แล้วทำการตรวจสอบรูปข้างล่างที่ติดกันต่อจากรูปนี้แล้วตรวจสอบไปเรื่อยๆ จนถึงรูปจุดสิ้นสุดโปรแกรม
- เมื่อพบรูปประกาศค่าตัวแปร จะทำการประกาศตัวแปร และเป็นการเก็บค่าเริ่มต้นของตัวแปรด้วยค่าที่ได้กำหนดไว้ ถ้าพบรูปมีการประกาศค่าตัวแปรซ้ำอีกที จะเป็นการเก็บค่าตัวแปรใหม่ทับตัวเก่า
- เมื่อพบรูปที่ใช้คู่กับการบวก ลบ คูณ หรือ หาร ก็จะทำการนำค่าตัวแปรที่มีอยู่ขณะนั้น ไปบวก ลบ คูณ หรือ หาร ด้วยค่าที่ได้กำหนดไว้ก่อน แล้วนำผลลัพธ์เก็บลงไปแทนตัวแปรตัวเดิม
- เมื่อพบรูปเงื่อนไข จะทำการตรวจสอบว่าค่าตัวแปรขณะนั้น มากกว่าหรือเท่ากับค่าที่กำหนดไว้ในเงื่อนไขหรือไม่ ถ้ามากกว่าหรือเท่ากัน จะไปทำงานที่ด้านซ้ายของเงื่อนไข แต่ถ้าน้อยกว่า จะไปทำงานที่ด้านขวาของเงื่อนไขแทน จนบรรจบกันที่ตัวปิดเงื่อนไข
- เมื่อพบรูปการวนทำซ้ำ ค่าของรูปการวนทำซ้ำคือจำนวนรอบที่ต้องการให้วนในลูปนั้นๆ จะมีการนับเพิ่มไปเรื่อยๆ ว่าจำนวนครั้งที่ได้วนไปแล้วเท่ากับจำนวนครั้งที่ต้องการให้วนแล้วหรือไม่ เมื่อวนครบตามรอบที่กำหนดแล้วจึงทำการออกจากการวนซ้ำนั้น
- เมื่อพบรูปฟังก์ชันที่มองไม่เห็นการทำงาน จะทำการหาว่ารูปนี้คือไฟล์ใด แล้วทำการอ่านไฟล์นั้นเป็นตาราง จากนั้นนำตารางมารันหาผลลัพธ์แบบอัตโนมัติ คือไม่เห็นการทำงานในฟังก์ชันนั้น แล้วนำค่าตัวแปรที่เป็นผลลัพธ์ทำงานต่อไป
- เมื่อพบรูปสิ้นสุดการทำงาน จะถือว่าสิ้นสุดการรันโปรแกรม ค่าของตัวแปรที่ได้ในขั้นตอนสุดท้ายคือผลลัพธ์ของโปรแกรมนี้

6.7.2 การรันแล้วแสดงผลลัพธ์ของโปรแกรมโดยอัตโนมัติ

จะเหมือนกับการแสดงผลทีละขั้นตอนการทำงานของโปรแกรม แต่ทำการแสดงผลโดยอัตโนมัติ ซึ่งจะได้คำตอบออกมาอย่างเดียว

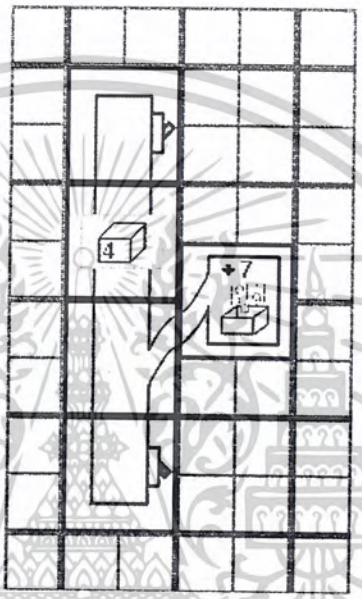
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทดลอง

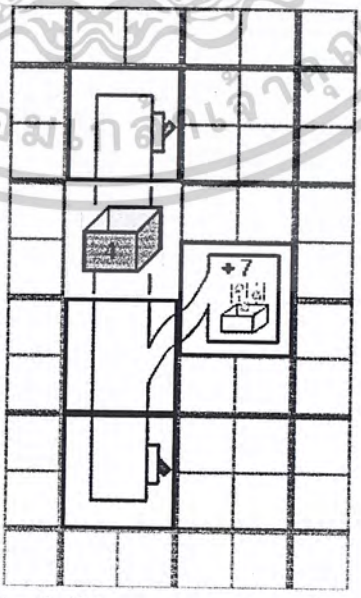
7.1 การทดลองประกาศตัวแปร และการบวก

เริ่มแรก คลิกเลือกรูปมาวาง และกำหนดค่า ดังรูปที่ 7-1 โดยการกดปุ่ม Run by Step และ กด Next เพื่อทำงานทีละขั้นตอน



รูปที่ 7-1 ตัวอย่างโปรแกรม การทดลองประกาศตัวแปร และการบวก

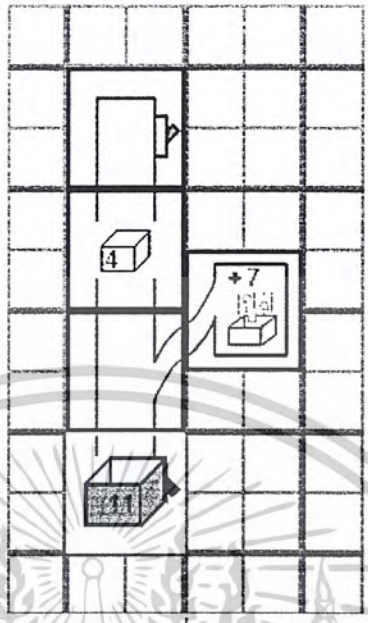
เมื่อทำงานมาถึงรูปกล่องจะมีกล่องสีฟ้ากรอบสีเหลืองแสดงขึ้นมาว่าในกล่องมีค่าตัวแปรเท่ากับ 4 ดังรูปที่ 7-2



รูปที่ 7-2 ตัวอย่างโปรแกรม ขณะทำการประกาศตัวแปร

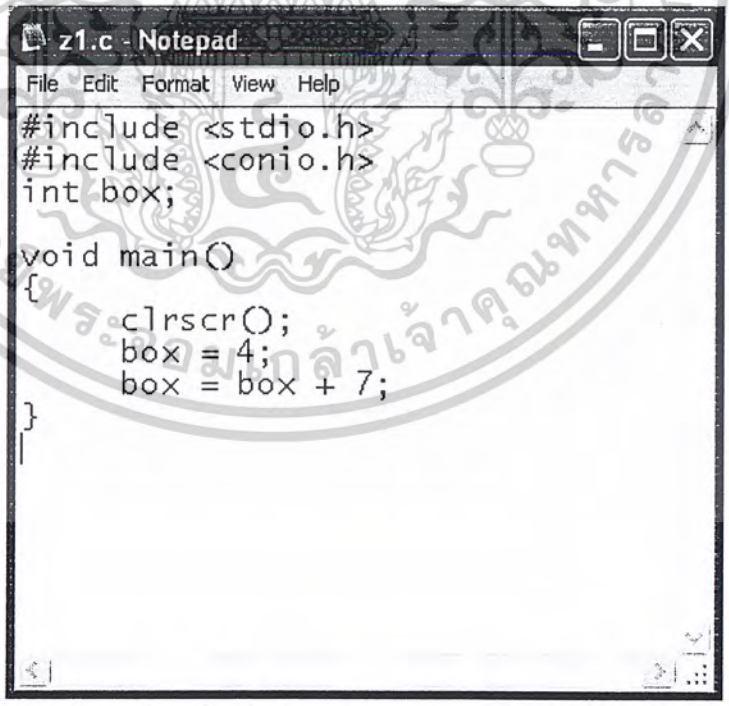
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อไปคือ ค่าในกล่อง (มีค่า 4) ถูกบวกด้วย 7 จะได้ค่าตัวแปรเป็น 11 ดังรูปที่ 7-3



รูปที่ 7-3 ตัวอย่างโปรแกรม ที่แสดงผลลัพธ์หลังทำการบวก

หลังจากจบโปรแกรม เมื่อกดปุ่ม Save จะแปลงจากโปรแกรมรูปภาพที่เขียนเป็นซอร์สโค้ดภาษาซี ดังรูปที่ 7-4

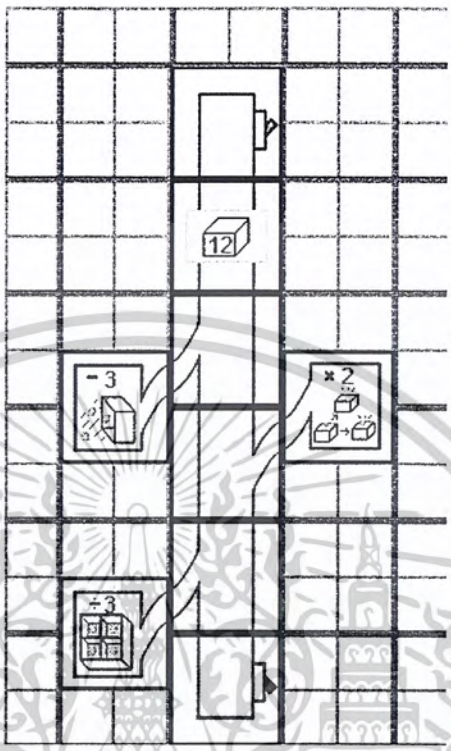


รูปที่ 7-4 ไฟล์ภาษาซีจากตัวอย่างโปรแกรม การทดลองประกาศตัวแปร และการบวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

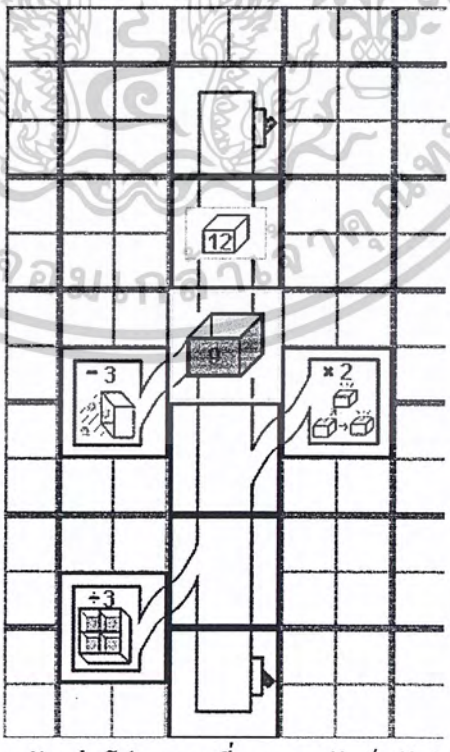
7.2 การทดลองการลบ คูณ และหาร

เริ่มแรก คลิกเลือกรูปมาวาง และกำหนดค่า ดังรูปที่ 7-5 และกดปุ่ม Run by Step และ กด Next เพื่อทำงานทีละขั้นตอน



รูปที่ 7-5 ตัวอย่างโปรแกรม การทดลองการลบ คูณ และหาร

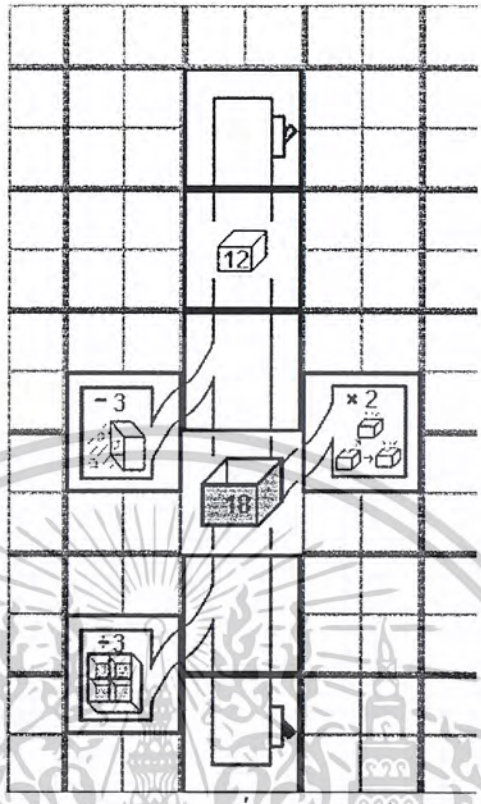
ขั้นต่อไป ค่าในกล่อง (มีค่าเป็น 12) ถูกลบด้วย 3 จะได้ค่าเป็น 9 ดังรูปที่ 7-6



รูปที่ 7-6 ตัวอย่างโปรแกรม ที่แสดงผลลัพธ์หลังทำการลบ

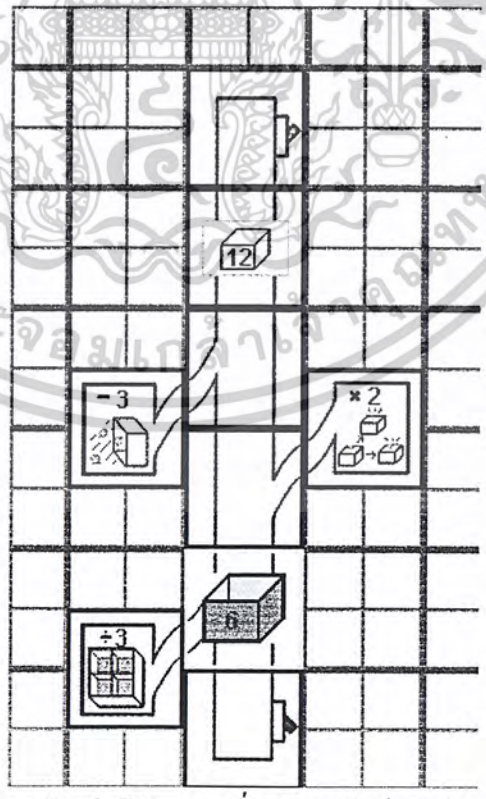
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อไปคือ ค่าในกล่อง (มีค่า 9) ถูกคูณด้วย 2 จะได้ค่าเป็น 18 ดังรูปที่ 7-7



รูปที่ 7-7 ตัวอย่างโปรแกรม ที่แสดงผลลัพธ์หลังทำการคูณ

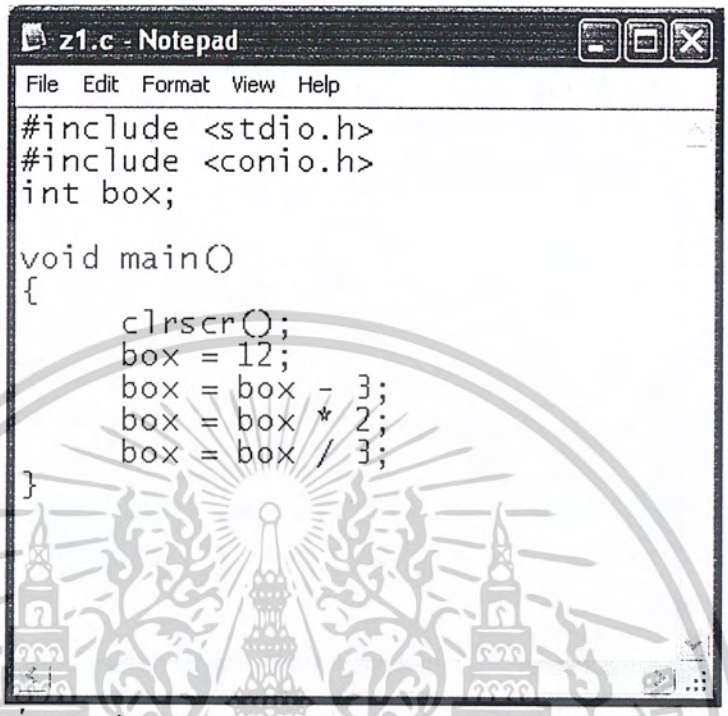
ขั้นตอนต่อไปคือ ค่าในกล่อง (มีค่า 18) ถูกหารด้วย 3 จะได้ค่าเป็น 6 ดังรูปที่ 7-8



รูปที่ 7-8 ตัวอย่างโปรแกรม ที่แสดงผลลัพธ์หลังทำการหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากจบโปรแกรม เมื่อกดปุ่ม Save จะแปลงจากโปรแกรมรูปภาพเป็นซอร์สโค้ดภาษาซี ดังรูปที่ 7-9

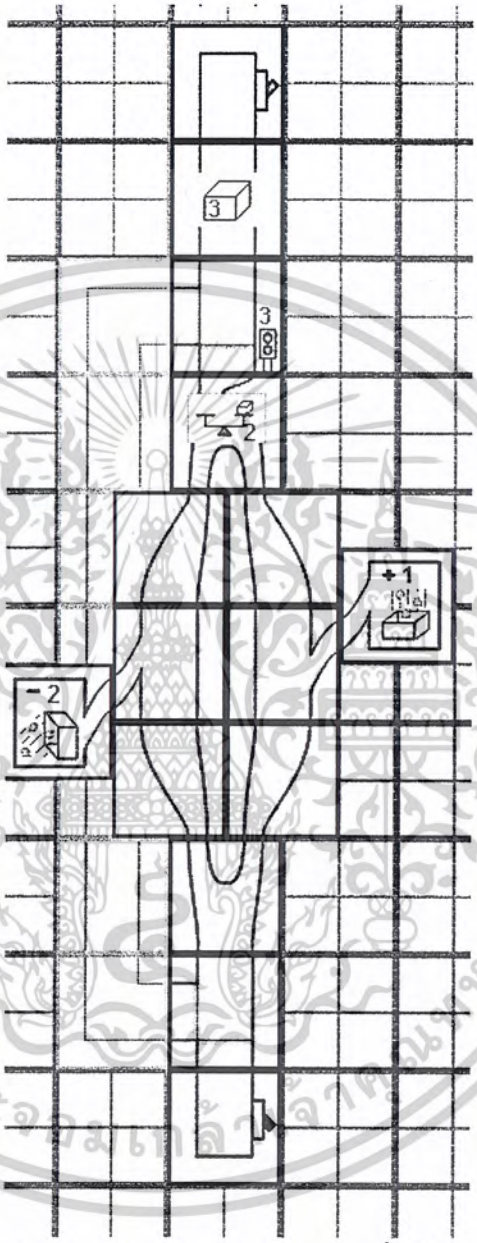


รูปที่ 7-9 ไฟล์ภาษาซีจากตัวอย่างโปรแกรม การทดลองการลบ คูณ และหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 การทดลองการตรวจสอบเงื่อนไขและการวนทำซ้ำ

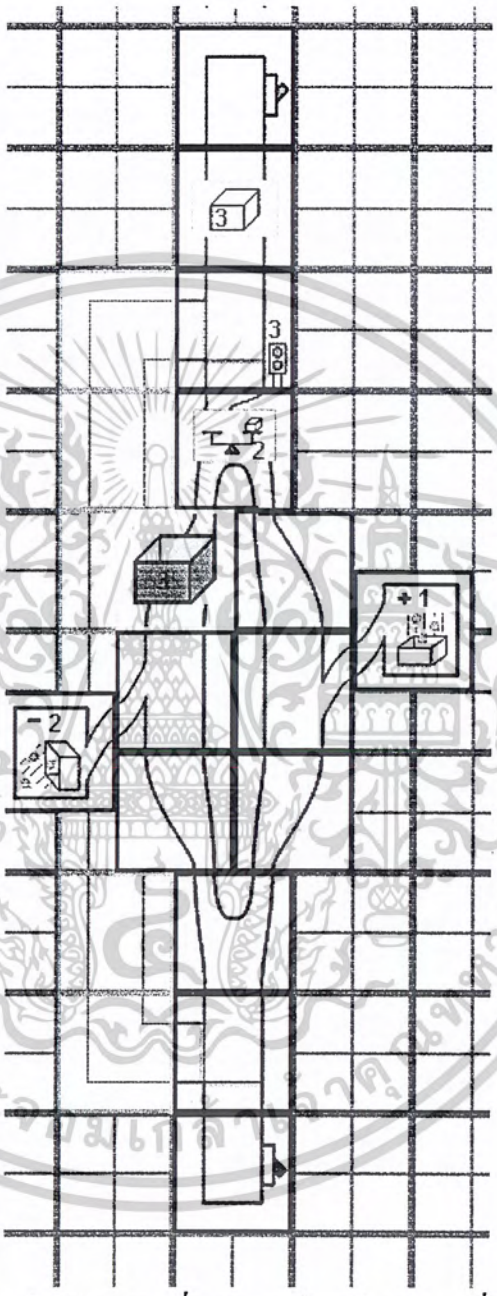
เริ่มแรก คลิกเลือกรูปมาวาง และกำหนดค่า ดังรูปที่ 7-10 โดยการกดปุ่ม Run by Step และ กด Next เพื่อทำงานทีละขั้นตอน



รูปที่ 7-10 ตัวอย่างโปรแกรม การทดลองการตรวจเงื่อนไขและการวนทำซ้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

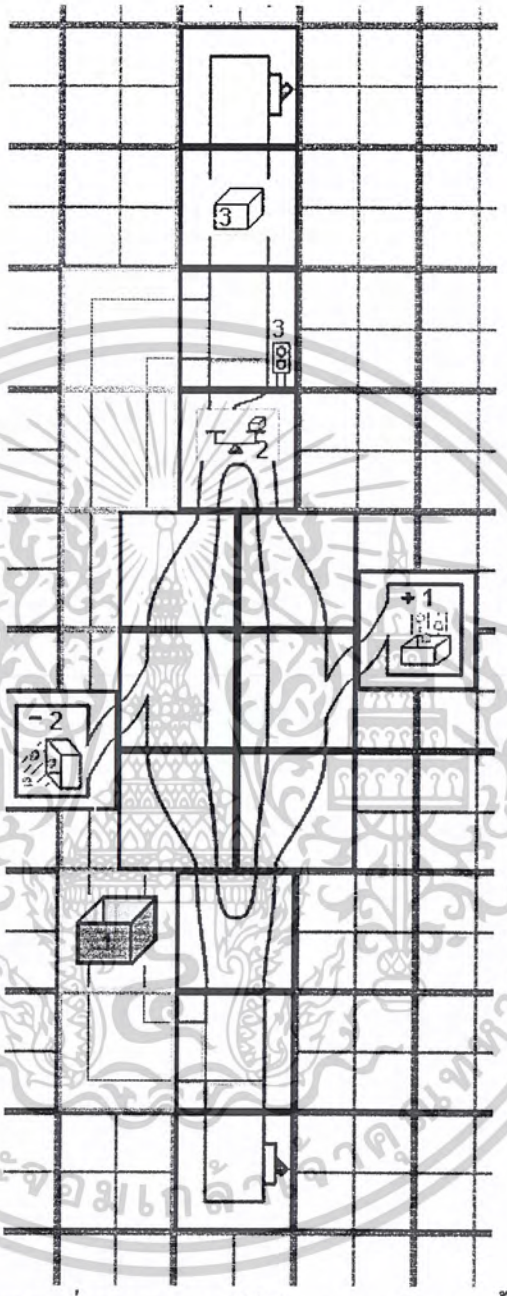
เมื่อทำงานผ่านรูปการวนซ้ำ ระบบจะตรวจสอบเปรียบเทียบค่าตัวแปร (มีค่าเป็น 3) กับค่าที่กำหนด (มีค่าเป็น 2) ค่าตัวแปรมีค่ามากกว่า ดังนั้นกล่องจึงมาทางซ้าย ผลลัพธ์คือระบบจะแสดงกล่องสีฟ้ากรอบสีเหลืองมีค่าตัวแปรเท่ากับ 3 ดังรูปที่ 7-11



รูปที่ 7-11 ตัวอย่างโปรแกรม ที่แสดงการเลือกเส้นทางจากเงื่อนไขครั้งแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

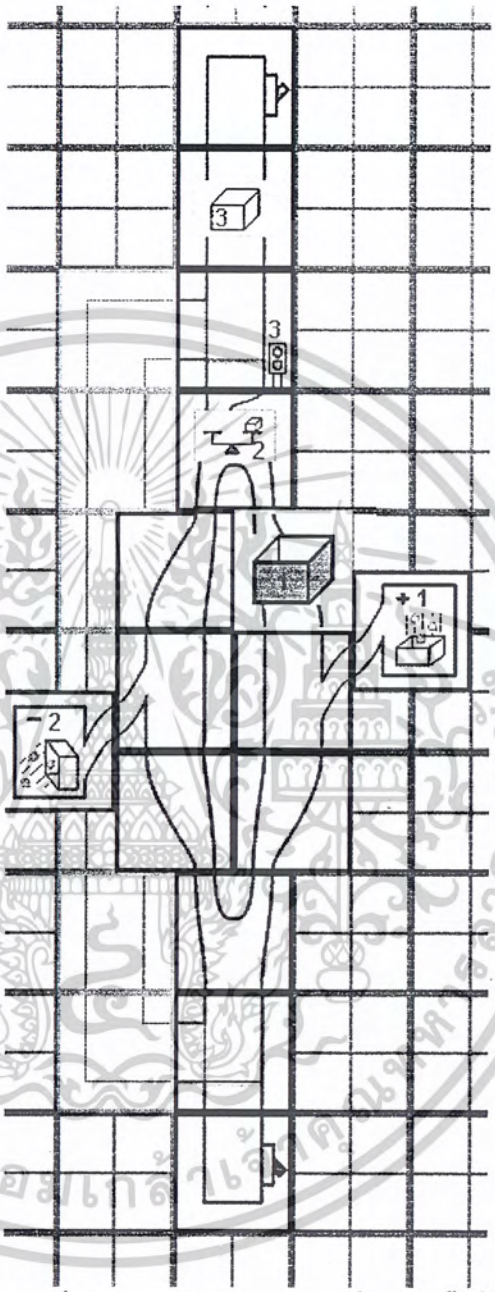
ขั้นต่อไป ค่าในกล่อง (มีค่าเป็น 3) ถูกลบด้วย 2 จะมีค่าเป็น 1 และต้องวนกลับมาขึ้นไปตรวจสอบว่าครบ 3 รอบหรือไม่ ดังรูปที่ 7-12



รูปที่ 7-12 ตัวอย่างโปรแกรม ที่แสดงการวนกลับไปตรวจสอบจำนวนครั้งการวนทำซ้ำรอบแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

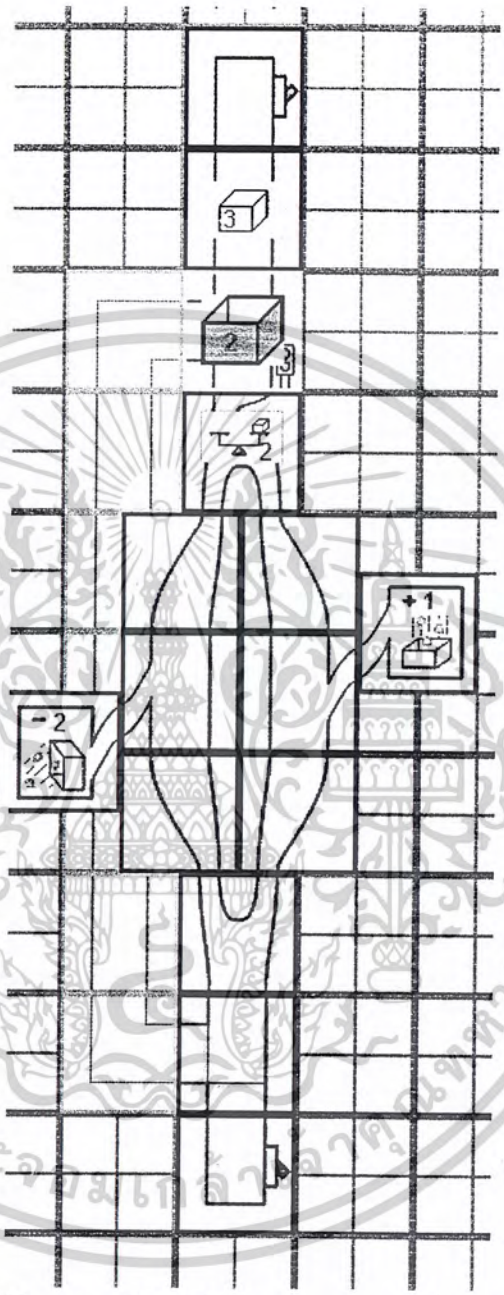
ซึ่งทำไปเพียงแค่ 1 รอบ จึงเข้ามาตรวจสอบค่าในกล่อง (มีค่าเป็น 1) กับค่าที่กำหนด (มีค่าเป็น 3) ในรอบนี้ค่าของกล่องมีค่าน้อยกว่าค่าที่กำหนด ดังนั้นกล่องจึงไปทางขวา ดังรูปที่ 7-13



รูปที่ 7-13 ตัวอย่างโปรแกรม ที่แสดงการเลือกเส้นทางจากเงื่อนไขครั้งที่สอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

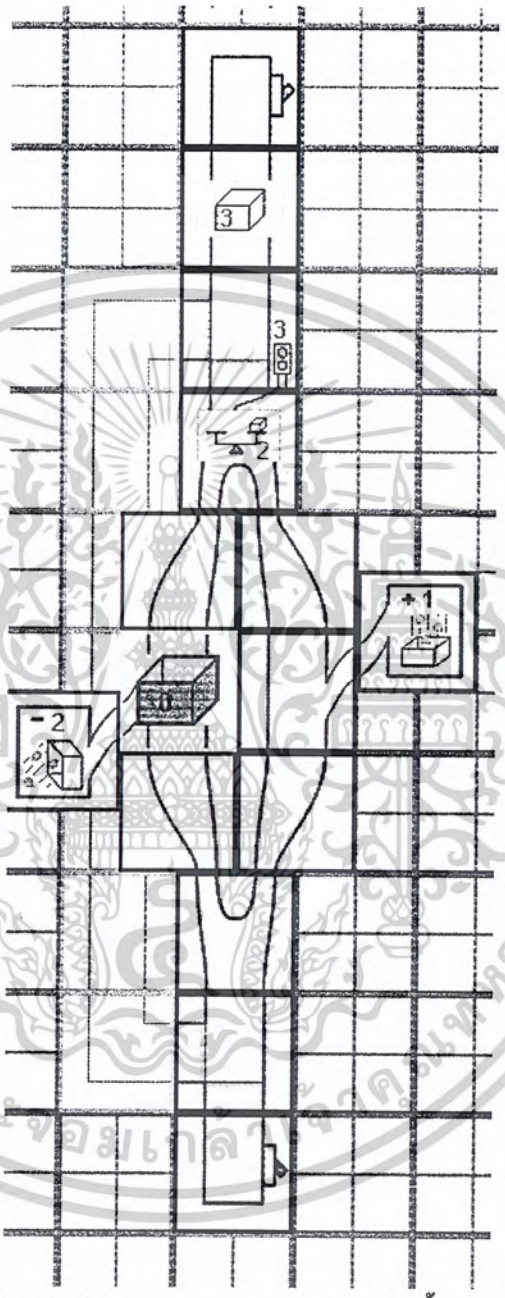
ขั้นตอนต่อไป ค่าในกล่อง (มีค่าเป็น 1) ถูกบวกด้วย 1 จะได้ค่าเป็น 2 แล้ววนกลับขึ้นไปตรวจสอบจำนวนรอบว่าครบแล้วหรือไม่ ดังรูปที่ 7-14



รูปที่ 7-14 ตัวอย่างโปรแกรม หลังจากทำการบวกและวนกลับไปตรวจสอบจำนวนครั้งการวนทำซ้ำรอบที่สอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

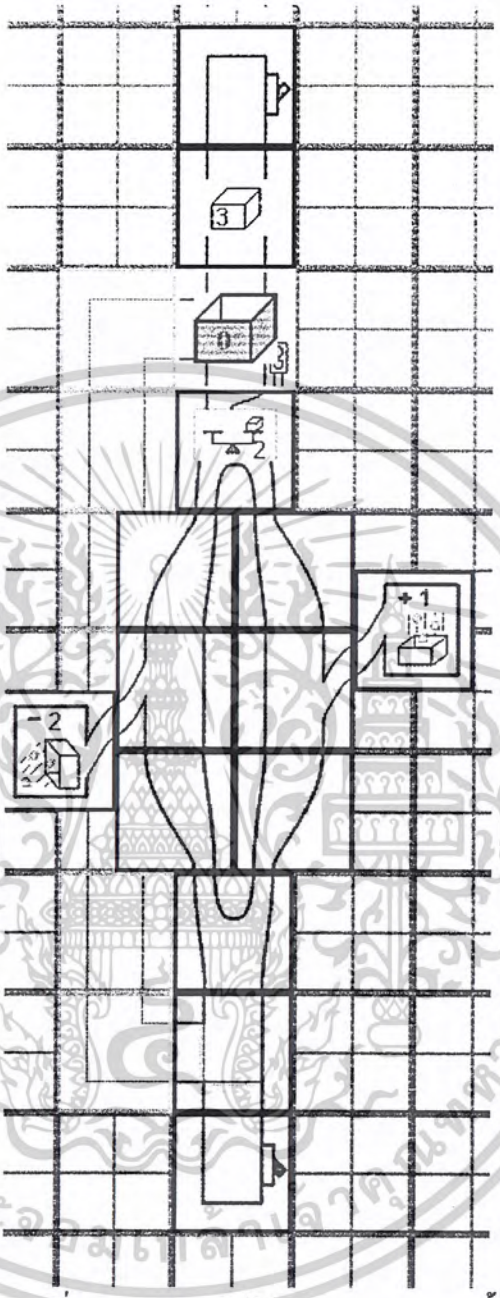
ซึ่งทำไปเพียง 2 รอบ จึงมีการทำรอบที่ 3 ซึ่งเป็นรอบสุดท้าย โดยเปรียบเทียบค่าในกล่อง (มีค่าเป็น 2) กับค่าที่กำหนด (มีค่าเป็น 2) ซึ่งมีค่าเท่ากัน กล่องจึงมาทางซ้าย จากนั้นค่าในกล่อง ถูกลบด้วย 2 จะได้ค่าเป็น 0 ดังรูปที่ 7-15



รูปที่ 7-15 ตัวอย่างโปรแกรม หลังการลบจากการทำซ้ำในลูปรอบที่สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

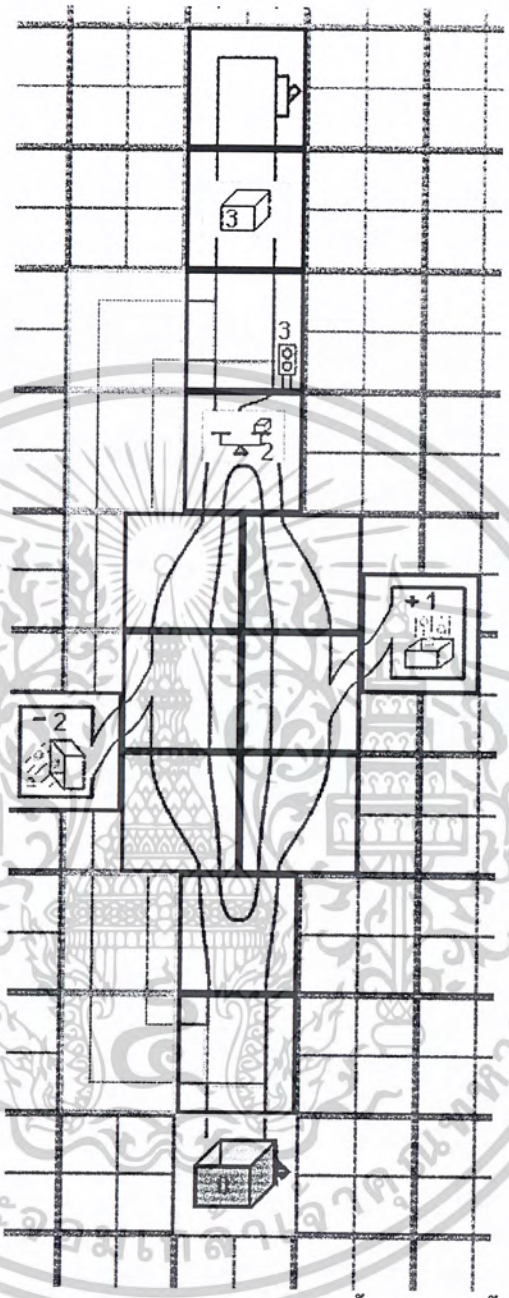
ขั้นต่อไป จะทำการตรวจสอบจำนวนรอบว่าครบหรือไม่ ซึ่งครบ 3 รอบแล้ว ดังรูปที่ 7-16



รูปที่ 7-16 ตัวอย่างโปรแกรม ที่แสดงการวนกลับไปตรวจสอบจำนวนครั้งการวนทำซ้ำรอบที่สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

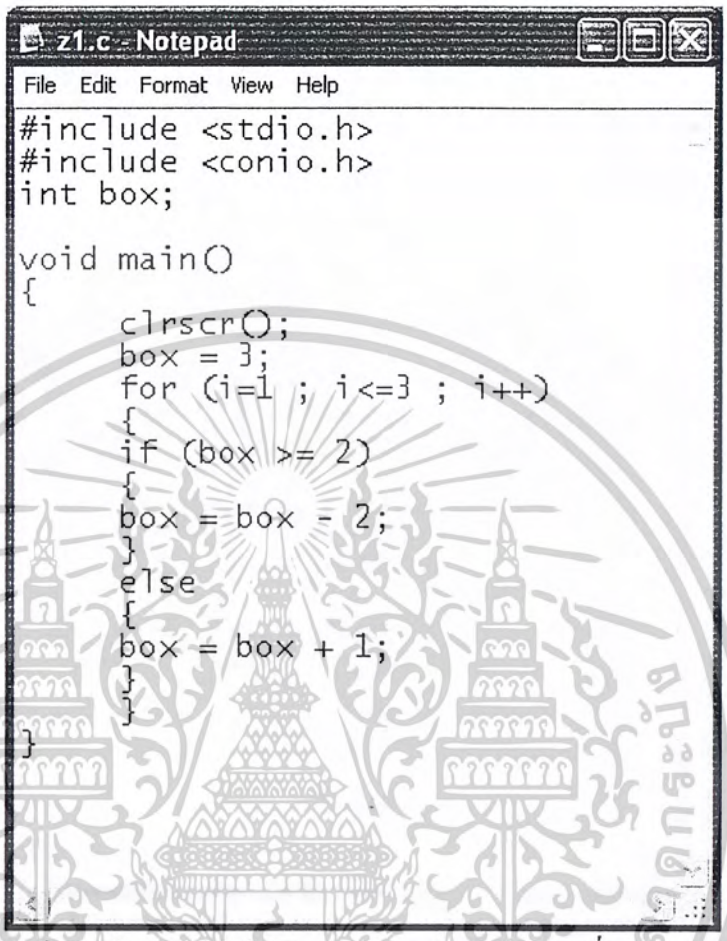
เมื่อครบ 3 รอบ กล่องจะไปอยู่ในตำแหน่งหลังจากจบการทำซ้ำ ดังรูปที่ 7-17



รูปที่ 7-17 ตัวอย่างโปรแกรม หลังจากการตรวจสอบจำนวนครั้งการวนทำซ้ำที่ครบรอบแล้วออกจากลูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากจบโปรแกรม เมื่อกดปุ่ม Save จะแปลงจากโปรแกรมรูปภาพเป็นซอร์สโค้ดภาษาซี ดังรูปที่ 7-18

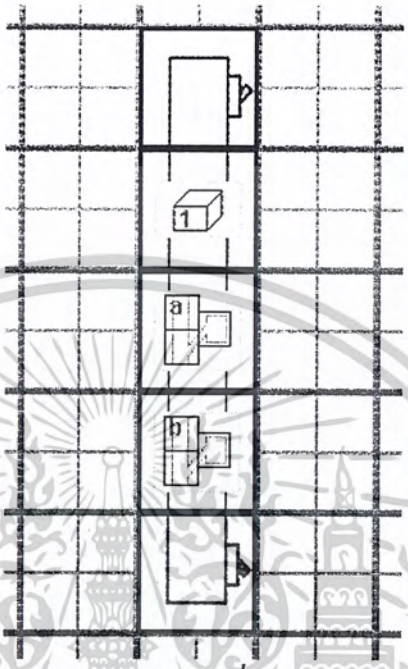


รูปที่ 7-18 ไฟล์ภาษาซีจากตัวอย่างโปรแกรม ทดลองการตรวจสอบเงื่อนไขและการวนทำซ้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

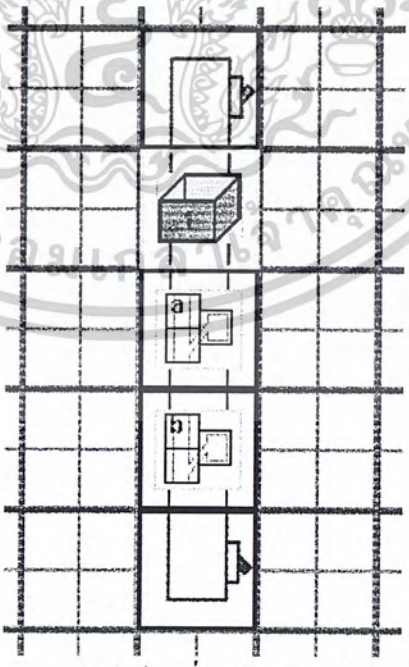
7.4 การทดลองฟังก์ชันที่มองไม่เห็นการทำงาน และการเรียกตัวเอง

เริ่มแรก คลิกเลือกรูปมาวาง และกำหนดค่า ดังรูปที่ 7-19 โดยการกดปุ่ม Run by Step และ กด Next เพื่อทำงานทีละขั้นตอน



รูปที่ 7-19 โปรแกรมหลักของการทดลองฟังก์ชันที่มองไม่เห็นการทำงาน และการเรียกตัวเอง

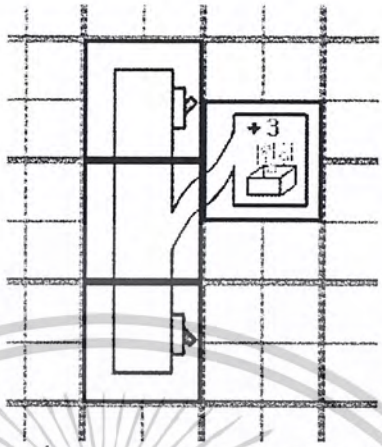
ขั้นแรก กล่องสีฟ้ากรอบสีเหลือง แสดงค่าในกล่องเป็น 1 ดังรูปที่ 7-20



รูปที่ 7-20 โปรแกรมหลักของการทดลองฟังก์ชันที่มองไม่เห็นการทำงาน และการเรียกตัวเอง ก่อนการเรียกใช้ฟังก์ชัน a

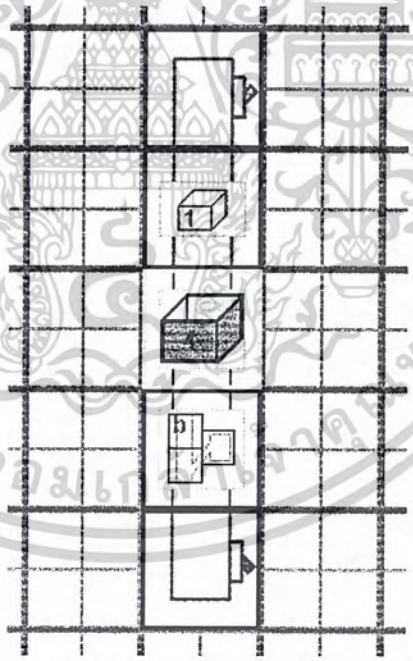
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าฟังก์ชัน a มีการทำงานภายในดังรูปที่ 7-21



รูปที่ 7-21 ฟังก์ชัน a ซึ่งจะมองไม่เห็นการทำงานภายใน ในโปรแกรมหลัก

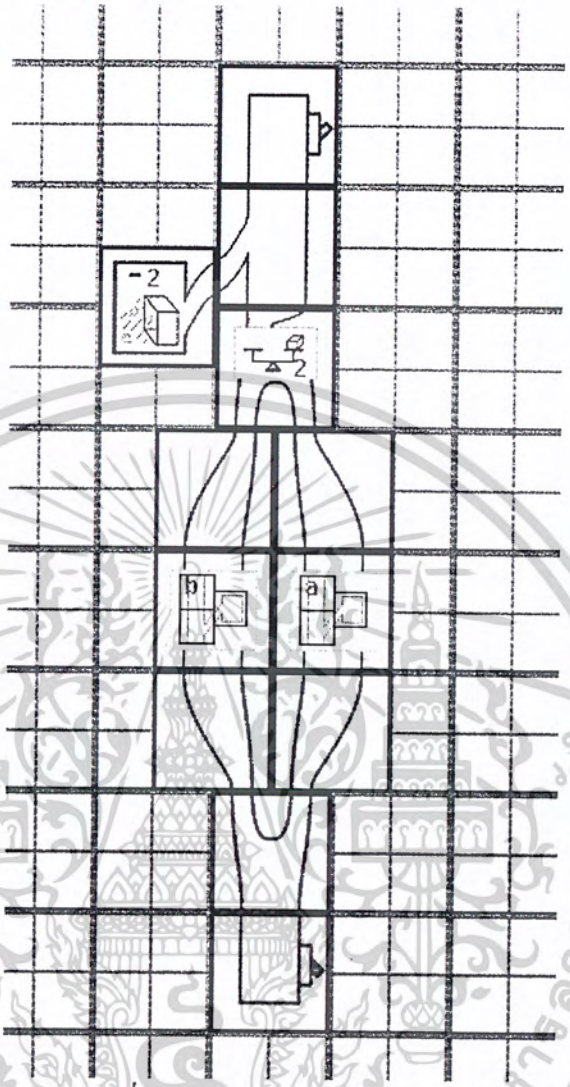
ค่าในกล่อง (มีค่าเป็น 1) ที่ถูกส่งผ่านมาในฟังก์ชัน a จะถูกบวกกับ 3 ได้ค่าเป็น 4 ดังรูปที่ 7-22



รูปที่ 7-22 โปรแกรมหลักของการทดลอง หลังผ่านการเรียกใช้ฟังก์ชัน a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน b มีการทำงานภายในดังรูปที่ 7-23



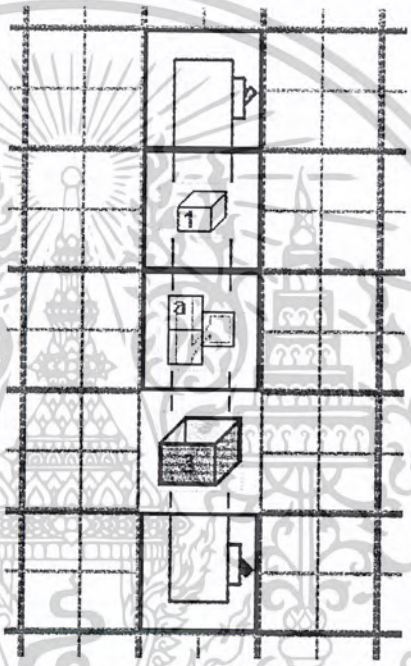
รูปที่ 7-23 ฟังก์ชัน b ซึ่งจะมองไม่เห็นการทำงานภายใน ในโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าในกล่อง (มีค่าเป็น 4) ที่ถูกส่งผ่านเข้ามาในฟังก์ชัน b จะถูกลบด้วย 2 ทำให้มีค่าเป็น 2 หลังจากนั้นเปรียบเทียบกับค่าที่กำหนด (มีค่าเป็น 2) ซึ่งทั้งสองค่ามีค่าเท่ากัน จึงมาทางซ้าย และเรียกใช้ฟังก์ชัน b ซึ่งเป็นการเรียกตัวเอง

ผลลัพธ์คือค่าในกล่อง (มีค่าเป็น 2) จะถูกส่งผ่านเข้ามาในฟังก์ชัน b อีกครั้ง และจะถูกลบด้วย 2 ทำให้มีค่าเป็น 0 หลังจากนั้นจะถูกนำไปเปรียบเทียบกับค่าที่กำหนด (มีค่าเป็น 2) ซึ่งค่าในกล่องมีค่าน้อยกว่าค่าที่กำหนด จึงมาทางขวา และเรียกใช้ฟังก์ชัน a

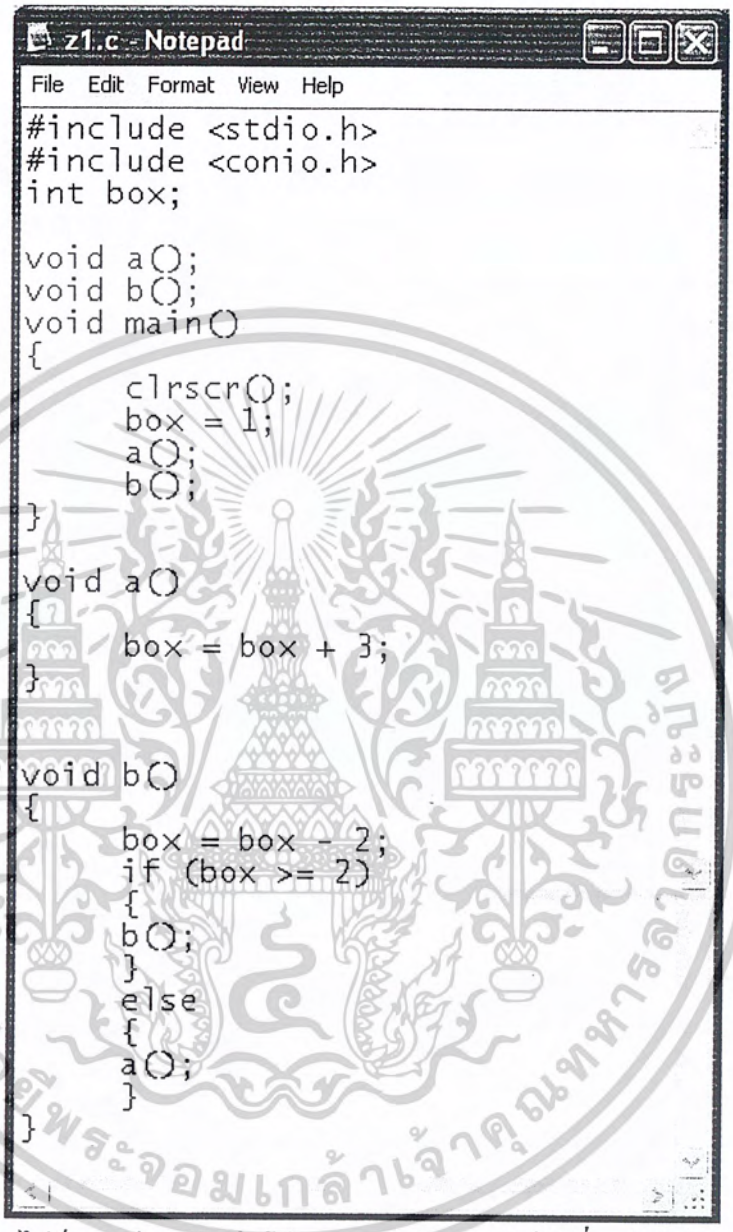
ก่อนที่จะทำงานในฟังก์ชัน a ค่าในกล่อง (มีค่าเป็น 0) และเมื่อถูกส่งผ่านเข้ามาในฟังก์ชัน a จะถูกบวกด้วย 3 ทำให้มีค่าเป็น 3 ซึ่งค่านี้จะเป็นค่าที่ถูกส่งกลับไปยังโปรแกรมหลัก ดังรูปที่ 7-24



รูปที่ 7-24 โปรแกรมหลักของการทำงาน หลังผ่านการเรียกใช้ฟังก์ชัน b

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากจบโปรแกรม เมื่อกดปุ่ม Save จะแปลงจากโปรแกรมรูปภาพเป็นซอร์สโค้ดภาษาซี ดังรูปที่ 7-25



รูปที่ 7-25 ไฟล์ภาษาซีจากตัวอย่างโปรแกรม การทดลองฟังก์ชันที่มองไม่เห็นการทำงานและการเรียกตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
บทสรุปและวิจารณ์

8.1 สรุปผลการดำเนินงาน

สำหรับส่วนสรุปผลการดำเนินการ จะเป็นการสรุปการทำงานในขั้นตอนต่างๆ โดยกล่าวเป็นส่วนๆ พร้อมทั้งกล่าวถึงปัญหาและวิธีการแก้ไข รวมทั้งได้สรุปการทำงานโดยรวมของระบบไว้ด้วย

8.1.1 ส่วนการศึกษาพัฒนาการของเด็ก

จากการศึกษาทฤษฎีต่างๆ ทำให้เข้าในความสามารถของเด็กในแต่ละวัย รวมถึงสามารถออกแบบได้ตรงตามความสามารถ และเหมาะสมกับการใช้งานของเด็กมากขึ้น

8.1.2 ส่วนการพัฒนาการออกแบบส่วนติดต่อกับผู้ใช้

ได้ออกแบบให้สามารถใช้งานด้วยเม้าส์ โดยหลีกเลี่ยงการใช้คีย์บอร์ดให้มากที่สุด รวมถึงออกแบบรูปภาพต่างๆ ให้สอดคล้องกับความเป็นจริงโดยจำลองการทำงานในโรงงานมาใช้ ซึ่งง่ายต่อความเข้าใจ

8.1.3 ส่วนการทดลองการทำงานของโปรแกรมภาษาคอมพิวเตอร์สำหรับเด็ก

หลังจากทดลองใช้งานโปรแกรมจริงๆ โปรแกรมสามารถทำงานได้อย่างถูกต้อง และมีประสิทธิภาพ ทั้งในด้านการออกแบบ การบันทึกไฟล์ การเปิดไฟล์ การใช้คำสั่งต่างๆ และการแปลงไฟล์เป็นภาษาซี

8.2 แนวทางการพัฒนาต่อ

เนื่องจากจุดประสงค์ในการพัฒนาโปรแกรมภาษาคอมพิวเตอร์สำหรับเด็กนั้นเน้นที่ความเข้าใจง่าย และสามารถให้เด็กได้เรียนรู้ภาษาคอมพิวเตอร์ได้อย่างสนุกสนาน และเพิ่มประสิทธิภาพของโปรแกรมให้ดียิ่งขึ้นต่อไป โดยมีแนวทางในการพัฒนาดังนี้

- เส้นทางตอนวนกลับของลูป ควรจะสามารถขยายหรือย่อได้ตามที่ต้องการ เพื่อจะได้ไม่มีภาพที่ซ้อนทับกัน
- ทุกเส้นทางของรูปควรมีการระบุทิศทางในการทำงาน เพื่อจะได้ไม่เกิดความสับสนในความเข้าใจและการวางภาพได้
- อาจมีการเพิ่มตัวแปรให้สามารถรองรับตัวแปรได้หลายชนิดและได้หลายตัว ทำให้สามารถเขียนโปรแกรมขนาดใหญ่ และหลากหลายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หลังจากทำการตรวจสอบความถูกต้อง แล้วควรจะมีการบอกตำแหน่งที่ไม่ถูกต้องทั้งหมด และบอกด้วยว่าไม่ถูกต้องเพราะเหตุใด เพราะในโปรแกรมสามารถหาได้เพียงตำแหน่งที่ผิดแห่งแรกที่พบเท่านั้น
- ในการบันทึกไฟล์ ควรจะสามารถระบุนามสกุลของไฟล์ว่าเป็นไฟล์ของโปรแกรมภาษาคอมพิวเตอร์สำหรับเด็ก หรือเป็นนามสกุลชนิดที่ต้องการได้
- การแสดงชื่อไฟล์บนฟังก์ชันที่มองไม่เห็นการทำงานควรจะแสดงชื่อไฟล์เป็นแบบป๊อป-อัพหรือกล่องข้อความช่วยแทน เพราะเนื่องจากถ้าชื่อฟังก์ชันยาวเกินไปก็จะเกินออกนอกรูปฟังก์ชันนี้ ซึ่งอาจทำให้เกิดการทับรูปอื่นได้
- การแสดงผล ควรแสดงผลเป็นรูปภาพเคลื่อนไหวต่อเนื่องเองอัตโนมัติ โดยไม่ต้องมีการกดทีละขั้นตอน เพื่อให้เด็กสามารถดูแล้วเข้าใจได้ง่ายขึ้นกว่าเดิม
- สามารถแปลงจากภาษาคอมพิวเตอร์สำหรับเด็กเป็นภาษาคอมพิวเตอร์ชนิดอื่น นอกจากภาษาซี เช่น ภาษาจาวา เป็นต้น
- สามารถอ่านไฟล์จากภาษาซีอย่างง่ายหรือภาษาอื่นที่กำหนดไว้ แปลงกลับมาเป็นภาษาคอมพิวเตอร์สำหรับเด็ก และแสดงผลแบบรูปภาพในโปรแกรมได้
- อาจมีการเปลี่ยนรูปหรือรูปแบบหน้าจอที่ติดต่อกับผู้ใช้ ให้ดูแล้วง่ายขึ้น ทำให้เด็กดูแล้วเข้าใจได้ง่ายขึ้นและสนุกกับการใช้โปรแกรมมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ดร.วีระศักดิ์ ซึงถาวร (2545) : "Java Programming Volume I", ซีเอ็ดยูเคชั่น, พ.ศ. 2545

[2] H.M. DEITEL, P.J. DEITEL : "JAVA HOW TO PROGRAM",fifth Edition, Prentice Hall,

[3] http://www.ceeo.tufts.edu/graphics/robolab.html

[4] http://www.cc.gatech.edu/~asb/papers/hci-for-kids.pdf

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้