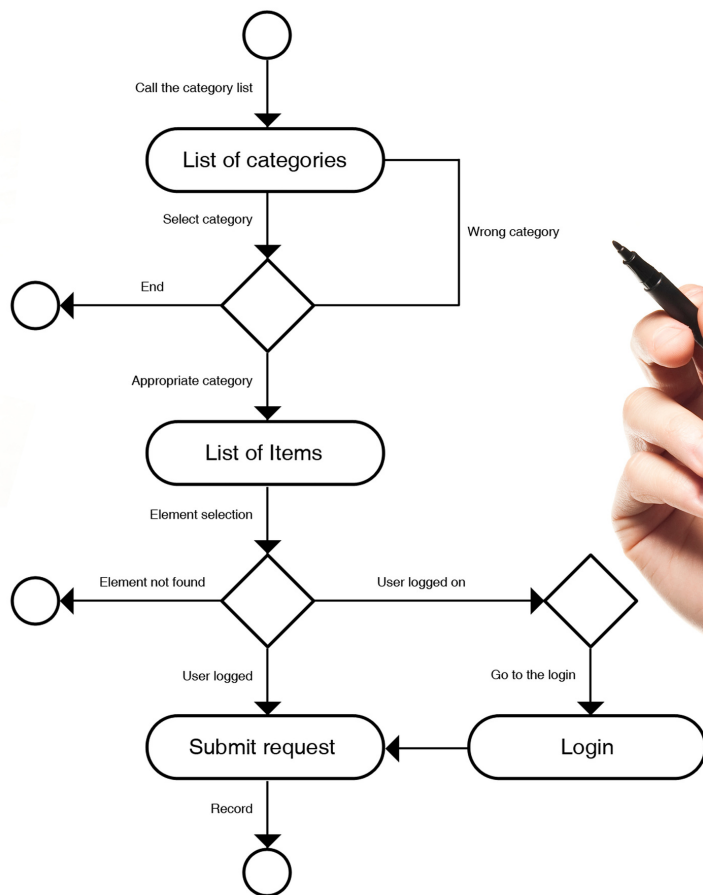




มโนทัศน์ที่คลาดเคลื่อนของนักเรียนระดับประถมศึกษา ในเขตภาคเหนือตอนบนเกี่ยวกับแนวคิดอัลกอริทึมใน การออกแบบผังงานโปรแกรม

THE UPPER NORTHERN REGION ELEMENTARY SCHOOL STUDENTS' MISCONCEPTIONS IN ALGORITHMIC CONCEPTS FOR PROGRAM FLOWCHART DESIGNING

กริธา แก้วคง*, สุรศักดิ์ เมาเทือก, ภาคิน บุรณเครือ,
พงษ์ระวี พันธุ์บุญปลูก และสุนีย์ เงินยวง

คณะศึกษาศาสตร์ มหาวิทยาลัยเชียงใหม่

*Corresponding email: kreetha.ka@cmu.ac.th

Receives: 26 December 2022, Revised: 14 June 2023, Accepted: 15 June 2023

บทคัดย่อ

บทความวิจัยนี้เป็นงานวิจัยเชิงคุณภาพมีวัตถุประสงค์เพื่อศึกษามโนทัศน์ที่คลาดเคลื่อนของนักเรียนระดับประถมศึกษาเกี่ยวกับแนวคิดอัลกอริทึมในการออกแบบผังงานโปรแกรม (program flowchart) ของนักเรียนระดับประถมศึกษาเพื่อใช้ในการแก้ปัญหาในชีวิตประจำวันผ่านการเขียนโปรแกรมร่วมกับการใช้บอร์ดไมโครคอนโทรลเลอร์ Kidbright กลุ่มเป้าหมายคือนักเรียนระดับประถมศึกษาจำนวน 76 คน จาก 3 โรงเรียน ในเขตภาคเหนือตอนบน ที่เข้าร่วมโครงการยกระดับทักษะโค้ดดิ้งสู่การประยุกต์ใช้ในท้องถิ่น สำหรับนักเรียนในโรงเรียนระดับการศึกษาขั้นพื้นฐานในพื้นที่จังหวัดภาคเหนือ การระดมมโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับแนวคิดอัลกอริทึมพื้นฐานสามารถวิเคราะห์ได้จากผังงานโปรแกรม ที่นักเรียนสร้างขึ้นจำนวน 44 ผังงาน

ผลการวิจัยพบว่า นักเรียนมีมโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการเขียนผังงานโปรแกรมที่หลากหลายซึ่งสามารถแบ่งออกเป็นสองลักษณะคือ มโนทัศน์ที่คลาดเคลื่อนที่พิจารณาจากปัญหาและอุปสรรคเกี่ยวกับการแก้ปัญหาจากการนำผังงานโปรแกรมที่สร้างขึ้นมาประกอบเข้าด้วยกัน (plan-composition problems) และมโนทัศน์ที่คลาดเคลื่อนที่พิจารณาจากปัญหาและอุปสรรคเกี่ยวกับโครงสร้างภาษาและไวยากรณ์ของการเขียนโปรแกรม (construct-based problems) นอกจากนี้ มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการตั้งเงื่อนไข มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการวนซ้ำ และ มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับอัลกอริทึมย่อย เป็นมโนทัศน์ที่คลาดเคลื่อนซึ่งปรากฏมากที่สุด กล่าวคือ ร้อยละ 84.00 ของ ผังงานโปรแกรมทั้งหมด

คำสำคัญ: มโนทัศน์ที่คลาดเคลื่อน, แนวคิดอัลกอริทึม, ผังงานโปรแกรม

Abstract

This research paper is qualitative research aiming to study the misconceptions about basic algorithm concepts in program flow design of elementary school students for solving problems in daily life through programming of the Kid bright microcontroller board. The samples are 76 elementary school students from 3 schools in the upper northern region of Thailand who participated in the Empowering Coding Skill to Local Application for Students from Basic Education Schools in Northern Area project. Identifying misconceptions about basic algorithmic concepts can be analyzed from 44 program flowcharts created by students.

The results showed that the students have a wide range of misconceptions about programming flowcharts that can be divided into two categories: The misconception that is considered from problems and obstacles about solving problems from the integration of the generated program flowchart. (plan-composition problems) and inconsistencies in concepts considered from problems and obstacles related to language structure and syntax (construct-based problems). In addition, condition misconceptions, loop misconceptions and the sub-algorithm misconception appeared the most, 84 percents of all flowchart programs.

Keywords: Misconceptions, algorithm, flowchart program

บทนำ

ปัจจุบันนโยบายทางการศึกษา การวิจัย และการพัฒนาเทคโนโลยี ให้ความสำคัญและส่งเสริมให้มีการจัดการเรียนการสอนที่เกี่ยวข้องกับวิทยาการคอมพิวเตอร์กันอย่างแพร่หลาย โดยเฉพาะอย่างยิ่งการเขียนโปรแกรมคอมพิวเตอร์ซึ่งปัจจุบันเป็นส่วนสำคัญของการจัดการเรียนการสอนวิทยาศาสตร์และเทคโนโลยีในระดับการศึกษาขั้นพื้นฐานจนถึงระดับอุดมศึกษา(Barendsen, et al., 2016; Clancy, 2004; Elo & Kyngäs, 2008) ในศตวรรษที่ 21 การเรียนรู้แนวคิดพื้นฐานด้านวิทยาการคอมพิวเตอร์และการเขียนโปรแกรมได้กลายเป็นสิ่งที่จำเป็นสำหรับทุกคน เนื่องจากลักษณะการทำงานและการใช้ชีวิตที่ขับเคลื่อนด้วยเทคโนโลยีและขึ้นอยู่กับกำรับส่งข้อมูลต่าง ๆ อย่างรวดเร็ว วิทยาการคอมพิวเตอร์และการเขียนโปรแกรมเป็นสิ่งจำเป็นซึ่งมนุษย์ควรเรียนรู้เพื่อใช้ในการแก้ปัญหาที่ซับซ้อนมากขึ้น (Wing, 2006) การรู้และเข้าใจในวิทยาการคอมพิวเตอร์และการเขียนโปรแกรมช่วยให้มนุษย์ไม่เพียงแต่กลายเป็นผู้บริโภคเทคโนโลยีแต่ยังทำหน้าที่เป็นผู้สร้างเทคโนโลยีและนักพัฒนาความรู้อีกด้วย (Rahimi et al., 2013) การเขียนโปรแกรมโดยทั่วไปสามารถแบ่งออกเป็นสองส่วนหลักที่สำคัญดังนี้คือ 1. การแก้ปัญหาด้วยอัลกอริทึม (algorithmic problem solving) และการเข้ารหัส (coding) สำหรับในส่วนของการแก้ปัญหาประกอบไปด้วย การวิเคราะห์ปัญหา การจำลองวิธีแก้ปัญหา และการออกแบบอัลกอริทึมสำหรับการนำวิธีแก้ปัญหาไปใช้ ส่วนการเข้ารหัสประกอบด้วยการใช้อัลกอริทึมที่ประดิษฐ์ขึ้นโดยใช้ภาษาโปรแกรมคอมพิวเตอร์ การแก้ไขจุดบกพร่องในการเขียนโปรแกรม (debugging) และการแก้ไข (modification) (Shneiderman, et al., 1977) ดังนั้นการคิดแบบอัลกอริทึมจึงมีบทบาทสำคัญใน

การแก้ปัญหาและการเขียนโปรแกรมอย่างยิ่ง (Wing, 2006) ด้วยนโยบายของสำนักงานส่งเสริมเศรษฐกิจดิจิทัล (depa) เล็งเห็นความสำคัญเกี่ยวกับการส่งเสริมการเขียนโปรแกรม จึงได้จัดโครงการยกระดับทักษะโค้ดดิ้งสู่การประยุกต์ใช้ในท้องถิ่นสำหรับนักเรียนในโรงเรียนระดับการศึกษาขั้นพื้นฐานในพื้นที่จังหวัดภาคเหนือ ซึ่งเป้าหมายของกิจกรรมต้องการให้นักเรียนในระดับประถมศึกษาสามารถออกแบบผังงานโปรแกรม และเขียนโปรแกรมเชื่อมต่อกับไมโครคอนโทรลเลอร์เพื่อใช้ในการแก้ปัญหาในท้องถิ่นได้

สำหรับผังงานโปรแกรม (program flowchart) ถูกนำมาใช้เพื่อแสดงภาพของอัลกอริทึมหรือตรรกะในการเขียนโปรแกรม ซึ่งเป็นส่วนสำคัญอันดับแรก ๆ ของการเขียนโปรแกรมตั้งแต่มีคอมพิวเตอร์เข้ามาใช้ในปี ค.ศ. 1940 (Shneiderman, et al., 1977; Zhao, et al., 2022) ผังงานโปรแกรม ถูกใช้เป็นเครื่องมือที่ง่ายและมีประสิทธิภาพในการทำความเข้าใจ ประดิษฐ์ แก้ไข แสดงภาพ แก้ไขจุดบกพร่อง แสดงและสื่อสารอัลกอริทึมและโปรแกรม (Scanlan, 1989) จากข้อสังเกตของ Shneiderman, et al. (1977) ผังงานโปรแกรม เป็นสิ่งที่แสดงถึง ระดับความสามารถ ในการแก้ปัญหาของการเขียนโปรแกรมนั้น ๆ นอกจากนี้ยังเป็นเครื่องมือที่มีประสิทธิภาพสำหรับนักพัฒนาโปรแกรมซึ่งสามารถใช้ในการปรับปรุงขั้นตอนในการแก้ปัญหาและการคิดแบบอัลกอริทึมก่อนที่จะเขียนโปรแกรมและแสดงถึงความสัมพันธ์ระหว่างขั้นตอนการทำงานของคอมพิวเตอร์และโปรแกรมถูกเขียนขึ้นได้อย่างชัดเจน (Bohl, 1971; Zhao, et al., 2022) จากการสังเกตของผู้วิจัยเมื่อมอบหมายให้นักเรียนได้ออกแบบผังงานภายใต้โครงการยกระดับทักษะโค้ดดิ้งสู่การประยุกต์ใช้ในท้องถิ่นสำหรับนักเรียนในโรงเรียนระดับการศึกษาขั้นพื้นฐานในพื้นที่จังหวัดภาคเหนือพบว่า อุปสรรคที่

สำคัญในการเขียนโปรแกรมคือการออกแบบผังงานโปรแกรม ซึ่งสอดคล้องกับรายงานวิจัยที่ผ่านมา (Chetty & van der Westhuizen, 2015; Sirki & Sorva, 2012) โดยรายงานการวิจัยพบว่านักเรียนขาดความรู้ความเข้าใจในการเขียนโปรแกรม รวมถึงความรู้ในภาษาโปรแกรมเฉพาะและความรู้ความเข้าใจเกี่ยวกับแนวคิดและโครงสร้างของการเขียนโปรแกรมเบื้องต้น เช่น การกำหนด ตัวแปร การเขียนการวนซ้ำของคำสั่ง การกำหนดเงื่อนไข ซึ่งล้วนแล้วเป็นการเปลี่ยนจากสิ่งที่เป็นามธรรมให้เป็นขั้นตอนและวิธีแก้ปัญหาเชิงรูปธรรม สำหรับอุปสรรคพื้นฐานในการเขียนโปรแกรมอาจส่งผลต่อการเขียนโปรแกรมที่ซับซ้อนขึ้นในอนาคต ดังนั้นการตระหนักและการแก้ไขมโนทัศน์ที่คลาดเคลื่อนเหล่านี้จึงเป็นประโยชน์ในการลดปัญหาในการเขียนโปรแกรมที่อาจเกิดขึ้นได้ ปัจจุบันมีการศึกษาและวิจัย มโนทัศน์ที่คลาดเคลื่อนของนักเรียนเกี่ยวกับโครงสร้างและการเขียนโปรแกรมพื้นฐานอย่างแพร่หลาย(Clancy, 2004; Mangaroska, et al., 2022; Putnam, et al., 1986; Robins, et al., 2003; Sorva, 2013; Spohrer & Soloway, 1986)

มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับแนวคิดอัลกอริทึมพื้นฐานที่ปรากฏในผังงานโปรแกรมนั้นสามารถพิจารณาได้จากการเขียนชุดคำสั่ง การลำดับชุดคำสั่ง การเปลี่ยนชุดคำสั่งเป็นสมการทางคณิตศาสตร์ และการกำหนดเงื่อนไขที่เหมาะสม เป็นต้น (Spohrer & Soloway, 1986; Sirki & Sorva, 2012; Rahimi, et al., 2017) การศึกษามโนทัศน์ที่คลาดเคลื่อนและปัญหาของนักเรียนเกี่ยวกับการใช้อัลกอริทึมในการเขียนโปรแกรมอย่างเป็นระบบทำให้ทราบว่านักเรียนแต่ละคนมีมโนทัศน์ในการเขียนโปรแกรมอย่างไร ตัวอย่าง เช่น Spohrer & Soloway (1986) ได้ทำการจัดกลุ่มมโนทัศน์ที่คลาดเคลื่อนของการใช้อัลกอริทึมในการเขียน

โปรแกรมเป็น 2 ลักษณะ โดยพิจารณาจากข้อผิดพลาดในการเขียน ผังงานโปรแกรม กล่าวคือ 1. plan-composition problems และ 2. construct-based problems ซึ่ง plan-composition problems หมายถึง ปัญหาหรืออุปสรรคของนักเรียนที่พบจากการสร้างหรือวิธีการในการแก้ปัญหาจากการนำผังงานโปรแกรมที่สร้างขึ้นมาประกอบเข้าด้วยกันสำหรับมโนทัศน์ที่คลาดเคลื่อนซึ่งพบจากข้อผิดพลาดแบบ plan-composition problems นี้เกี่ยวข้องกับการกำหนดวิธีการแก้ปัญหา ในการกำหนดอัลกอริทึม และการวางแผน ความหมายและตรรกะของโปรแกรมที่เขียนขึ้น (Chetty & van der Westhuizen, 2015) ในทางกลับกัน construct-based problems หมายถึง ปัญหาหรืออุปสรรคของนักเรียนที่พบจากโครงสร้างภาษาและไวยากรณ์ของการเขียนโปรแกรม (Spohrer & Soloway, 1986) สำหรับงานวิจัยนี้ผู้วิจัยได้ศึกษามโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับแนวคิดอัลกอริทึมในการออกแบบ ผังงานโปรแกรมของนักเรียนระดับประถมศึกษาที่เข้าร่วมโครงการ ยุกระดับทักษะโค้ดดิ้งสู่การประยุกต์ใช้ในท้องถิ่น สำหรับนักเรียนในโรงเรียนระดับการศึกษาขั้นพื้นฐานในพื้นที่จังหวัดภาคเหนือ โดยประยุกต์ใช้เกณฑ์การพิจารณาของ Rahimi et al. (2017) ในการจัดกลุ่มมโนทัศน์ที่คลาดเคลื่อน

วัตถุประสงค์การวิจัย

เพื่อศึกษามโนทัศน์ที่คลาดเคลื่อนของนักเรียนระดับประถมศึกษาเกี่ยวกับแนวคิดอัลกอริทึมในการออกแบบ ผังงานโปรแกรม (program flowchart)

ขอบเขตการวิจัย

ขอบเขตเกี่ยวกับเนื้อหา ได้แก่ บริบทเกี่ยวกับการออกแบบผังงานโปรแกรมเพื่อเขียนโปรแกรมในการแก้ปัญหาร่วมกับบอร์ดไมโครคอนโทรลเลอร์ Kidbright ในการดูแลผู้สูงอายุ ในเขตภาคเหนือตอนบน

วิธีดำเนินการวิจัย

การวิจัยนี้เป็นการวิจัยเชิงคุณภาพซึ่งใช้วิธีการวิเคราะห์เนื้อหา จัดกลุ่ม มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับอัลกอริทึมในการออกแบบผังงานโปรแกรม พร้อมทั้งจำแนกลักษณะของมโนทัศน์ที่คลาดเคลื่อนที่ประยุกต์มาจาก Rahimi, et al. (2017)

ประชากรและกลุ่มตัวอย่าง

กลุ่มเป้าหมายของการวิจัยครั้งนี้ คือ นักเรียนระดับประถมศึกษาตอนปลาย จำนวน 76 คน ของโรงเรียนขนาดกลาง 3 โรงเรียน ซึ่งเข้าร่วมโครงการยกระดับทักษะโค้ดดิ้งสู่การประยุกต์ใช้ในท้องถิ่น สำหรับนักเรียนในโรงเรียนระดับการศึกษาขั้นพื้นฐานในพื้นที่จังหวัดภาคเหนือ (Empowering Coding Skill to Local Application for Students from Basic Education Schools in Northern Area) พ.ศ.2565 ซึ่งสามารถสรุปข้อมูลของประชากรได้ดังตารางที่ 1

ตารางที่ 1 ข้อมูลของกลุ่มเป้าหมายสำหรับการวิจัยในครั้งนี้

โรงเรียน	ระดับชั้น	จำนวนนักเรียน (คน)	จำนวนผังงานโปรแกรม(ผัง)
A	ป.4	30	14
B	ป.5	6	6
C	ป.6	40	24
รวม		76	44

เครื่องมือในการวิจัย

1. โจทย์คำถามการสร้างผังงานโปรแกรมเพื่อใช้แก้ปัญหาตามสถานการณ์เกี่ยวกับการดูแลผู้สูงอายุ ควบคู่กับบอร์ดไมโครคอนโทรลเลอร์ Kidbright

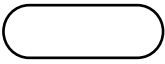
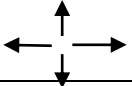
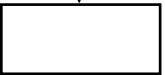
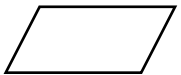
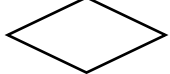
2. แบบวิเคราะห์เนื้อหา เพื่อการจำแนกลักษณะความคลาดเคลื่อนของมโนทัศน์ฯ ที่ประยุกต์ใช้จาก Rahimi et al. (2017)

การเก็บรวบรวมข้อมูล

สำหรับกิจกรรมในโครงการยกระดับทักษะโค้ดดิ้งสู่การประยุกต์ใช้ในท้องถิ่น สำหรับนักเรียนใน

โรงเรียนระดับการศึกษาขั้นพื้นฐานในพื้นที่จังหวัดภาคเหนือ ถูกพัฒนาขึ้นโดยนักวิจัย นักเรียนจะได้รับการแนะนำให้รู้จักกับลักษณะของบอร์ดไมโครคอนโทรลเลอร์ Kidbright หลักการในการเขียนผังงานโปรแกรมและการเขียนโปรแกรมสำหรับบอร์ดไมโครคอนโทรลเลอร์ Kidbright โดยการบรรยาย และฝึกปฏิบัติการเขียนผังงานโปรแกรม นักเรียนจะได้รับการแนะนำเกี่ยวกับลักษณะการเขียนผังงานโปรแกรมพื้นฐานโดยการเรียงลำดับสัญลักษณ์ ดังแสดงในตารางที่ 2

ตารางที่ 2 สัญลักษณ์ที่จำเป็นในการเขียนผังงาน

สัญลักษณ์	ชื่อที่เรียก	ความหมาย
	Terminator	จุดเริ่มต้น และ จุดสิ้นสุดของการทำงาน
	Flowline / Direction	เส้นแสดงทิศทางการทำงาน ต้องมีหัวลูกศรเดียวทั้งนั้น
	Process	การปฏิบัติงาน / ประมวลผล
	Input / Output	การรับหรือแสดงผลโดยไม่กำหนดชนิดของอุปกรณ์ในการทำงาน
	Keyboard	รับ / อ่านข้อมูลที่รับมาจากคีย์บอร์ด
	Monitor	แสดงรายละเอียดข้อมูล หรือผลลัพธ์ทางจอภาพ
	Printer	แสดงรายละเอียดข้อมูล หรือผลลัพธ์ทางเครื่องพิมพ์
	Decision	การตัดสินใจ หรือ จุดที่ต้องเลือกปฏิบัติอย่างใดอย่างหนึ่ง
	In-paper Connector	จุดเชื่อมต่อภายในหน้าเดียวกัน

จากนั้นเปิดโอกาสให้นักเรียนได้ฝึกการเขียนและปรับปรุงแก้ไขผังงานโปรแกรมให้มีประสิทธิภาพมากขึ้น จากนั้นนักเรียนจะได้รับการมอบหมายให้ออกแบบผังงานโปรแกรมเพื่อแก้ปัญหาตามสถานการณ์เกี่ยวกับการดูแลผู้สูงอายุ เช่น การออกแบบเครื่องป้องกันการเดินชนสิ่งกีดขวางสำหรับผู้สูงอายุ การดูแลเกี่ยวกับอากาศและความชื้นของที่อยู่อาศัยที่เหมาะสมต่อการอยู่และอาศัยของผู้สูงอายุ เป็นต้น ผังงานดังกล่าวจะได้รับการประเมินและถูกนำมาวิเคราะห์เพื่อระบุโมทัศน์ที่คลาดเคลื่อนซึ่งเป็นข้อมูลของการวิจัยในครั้งนี้

การวิเคราะห์ข้อมูลและสถิติที่ใช้

ผังงานโปรแกรมทั้งหมด 44 ผัง จะถูกนักวิจัยแต่ละคนนำมาวิเคราะห์โดยใช้วิธีการวิเคราะห์เชิงคุณภาพโดยวิธีวิเคราะห์เนื้อหา (content analysis)(Elo & Kyngäs, 2008) อย่างอิสระ โดยใช้เกณฑ์การพิจารณาโมทัศน์ที่คลาดเคลื่อน ซึ่งประยุกต์มาจาก (Rahimi, et al., 2017) โดยสามารถจำแนกออกเป็น 2 กลุ่ม คือ 1. โมทัศน์ที่คลาดเคลื่อนที่พบจากวิธีแก้ปัญหาในการนำผังงานโปรแกรมมาประกอบเข้าด้วยกัน (plan-composition problems) (Chetty & van der Westhuizen, 2015) 2. โมทัศน์ที่คลาดเคลื่อน

ที่พบจากโครงสร้างภาษาและไวยากรณ์ของการเขียนโปรแกรม (construct-based problems) (Spohrer & Soloway, 1986) ซึ่งเกณฑ์การพิจารณาของมโนทัศน์ที่คลาดเคลื่อนทั้ง 2 ลักษณะ มีลักษณะดังตารางที่ 3 และ 4 ตามลำดับ ผังงานแต่ละผัง อาจพบมโนทัศน์ที่คลาดเคลื่อน มากกว่า 1 มโนทัศน์

การระบุว่าผังงานโปรแกรมใดปรากฏมโนทัศน์ที่คลาดเคลื่อนได้นั้น เกิดจากการลงความเห็นอย่างเป็นเอกฉันท์ของนักวิจัยและผู้ร่วมวิจัยทุกคน จำนวนมโนทัศน์ที่คลาดเคลื่อนแต่ละมโนทัศน์จะถูกนำมาคิดเป็นร้อยละของจำนวนผังงานโปรแกรมทั้งหมด (44 ผัง)

ตารางที่ 3 เกณฑ์การวิเคราะห์มโนทัศน์ที่คลาดเคลื่อน มโนทัศน์ที่คลาดเคลื่อนที่พบจากวิธีแก้ปัญหำ ในการนำผังงานโปรแกรมมาประกอบเข้าด้วยกัน (plan-composition problems) ซึ่งประยุกต์จาก Rahimi, et al.(2017)

ประเภทของมโนทัศน์ที่คลาดเคลื่อน	เกณฑ์การพิจารณา	การแจกแจงของข้อมูล
1. มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการตั้งเงื่อนไข (condition misconceptions)	- ไม่พิจารณาตัวแปรที่มีค่าเท่ากัน หรือเป็นตัวแปรที่เป็นช้อยกเว้น เพื่อให้มีการดำเนินการต่อไป - การกำหนดเงื่อนไขเริ่มต้นที่ไม่ถูกต้อง เพื่อให้ให้มีการดำเนินการของคำสั่งถัดไป - ตีความแนวคิดเงื่อนไขผิด - อื่น ๆ ที่เกี่ยวข้องกับการตั้งเงื่อนไข	- ผู้วิจัยสามารถวิเคราะห์และจำแนก มโนทัศน์ที่คลาดเคลื่อนที่เกิดจากวิธีแก้ปัญหำ ในการนำผังงานโปรแกรมมาประกอบเข้าด้วยกัน (plan-composition problems) โดยการพิจารณาว่า ผังงานโปรแกรม ที่นักเรียนได้ออกแบบไว้นั้น พบมโนทัศน์ที่คลาดเคลื่อนลักษณะใด (มีทั้งหมด 4 ลักษณะ) ผังงานโปรแกรม 1 ผัง อาจพบมโนทัศน์ที่คลาดเคลื่อนได้มากกว่า 1 ลักษณะ เช่น หากพบว่าผังงานโปรแกรมมีการกำหนดเงื่อนไขเริ่มต้นที่ไม่ถูกต้อง และใช้อัลกอริทึมย่อยผิด ในการแก้ปัญหา ดังนั้น ผังงานโปรแกรมดังกล่าว ปรากฏมโน
2. มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการวนซ้ำ (loop misconceptions)	- ไม่สามารถวนซ้ำได้ - เงื่อนไขไม่สามารถทำให้มีการดำเนินการวนซ้ำได้ - มีการวนซ้ำที่เป็นช้อยกเว้น - อื่น ๆ ที่เกี่ยวข้องกับการวนซ้ำ	
3. มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับอัลกอริทึมย่อย (sub-algorithm misconceptions)	- เกิดข้อผิดพลาดในการใช้อัลกอริทึมย่อย ในการแก้ปัญหา - อัลกอริทึมย่อย ไม่สามารถนำมาใช้แก้ปัญหาได้ - อื่น ๆ ที่เกี่ยวข้องกับการใช้อัลกอริทึมย่อย	

ประเภทของมโนทัศน์ ที่คลาดเคลื่อน	เกณฑ์การพิจารณา	การเจแนบับของข้อมูล
4. มโนทัศน์ที่คลาดเคลื่อน เกี่ยวกับการตีความ (translation misconceptions)	- ภาษาที่ใช้ในการเขียนโปรแกรมไม่ สามารถนำมาใช้ในการแก้ปัญหาได้ - ลำดับของผังงานโปรแกรมไม่ถูกต้อง ตามวัตถุประสงค์ของการแก้ปัญหา - อื่น ๆ ที่เกี่ยวข้องกับการตีความของ การแก้ปัญหา	ทัศน์ที่คลาดเคลื่อน 2 ลักษณะ คือ มโนทัศน์ที่คลาดเคลื่อน เกี่ยวกับการตั้งเงื่อนไข และ มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับ อัลกอริทึมย่อย เป็นต้น

ตารางที่ 4 ตัวอย่างเกณฑ์การวิเคราะห์มโนทัศน์ที่คลาดเคลื่อนที่พบจากโครงสร้างภาษาและไวยากรณ์ของการเขียนโปรแกรม (construct-based problems) ซึ่งประยุกต์จาก Rahimi, et al.(2017)

ประเภทของมโนทัศน์ ที่คลาดเคลื่อน	เกณฑ์การพิจารณา	การเจแนบับของข้อมูล
1. มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับ การตั้งเงื่อนไข (condition misconceptions)	- ไม่พิจารณาตัวแปรที่มีค่าเท่ากัน หรือ เป็นตัวแปรที่เป็นข้อยกเว้น เพื่อให้มี การดำเนินการต่อไป - การกำหนดเงื่อนไขเริ่มต้นที่ไม่ถูกต้อง เพื่อให้มีการดำเนินการของคำสั่ง ถัดไป - ตีความแนวคิดเงื่อนไขผิด - อื่น ๆ ที่เกี่ยวข้องกับการตั้งเงื่อนไข	- ผู้วิจัยสามารถวิเคราะห์และ จำแนก มโนทัศน์ที่คลาดเคลื่อน ที่เกิดจากโครงสร้างภาษาและ ไวยากรณ์ของการเขียน โปรแกรม (construct-based problems) โดยการพิจารณา ว่า ผังงานโปรแกรม ที่นักเรียน ได้ออกแบบไว้นั้น พบมโนทัศน์
2. มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับ ลำดับ (sequence misconceptions)	- มีสถานการณ์ที่มีการดำเนินการ มากกว่าหนึ่งรายการแต่ประมวลผลใน ขั้นตอนเดียว - ลำดับในการแก้ปัญหาไม่เป็นไปตาม วัตถุประสงค์ - อื่น ๆ เกี่ยวข้องกับลำดับ	ที่คลาดเคลื่อนลักษณะใด (มี ทั้งหมด 6 ลักษณะ) ผังงาน โปรแกรม 1 ผัง อาจพบมโน ทัศน์ที่คลาดเคลื่อนได้มากกว่า 1 ลักษณะ เช่น หากพบว่าผัง งานโปรแกรมมีการเขียนคำสั่ง

ประเภทของมโนทัศน์ ที่คลาดเคลื่อน	เกณฑ์การพิจารณา	การเจงนั้บของข้อมูล
3. มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการวนซ้ำ (loop misconceptions)	- การสร้างการวนซ้ำที่ไม่ถูกต้อง ไม่สามารถวนได้ - การวนซ้ำมีข้อยกเว้น ไม่ครอบคลุมกับวัตถุประสงค์ - อื่น ๆ เกี่ยวกับ การวนซ้ำ	ในการวนซ้ำไม่ถูกต้อง และสัญลักษณ์ในการสร้างผังงานโปรแกรมไม่ถูกต้อง ดังนั้นผังงานโปรแกรมหังกล่าว ปรากฏมโนทัศน์ที่ คลาดเคลื่อน 2
4. มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการสั่งงาน (assignment misconceptions)	- ตำแหน่งของคำสั่งเริ่มต้นและคำสั่งสุดท้ายอยู่ผิดตำแหน่งหรือสลับกัน - อื่น ๆ เกี่ยวกับ การสั่งงาน	ลักษณะคือ มโนทัศน์ ที่คลาดเคลื่อนเกี่ยวกับการวนซ้ำและมโนทัศน์ที่คลาดเคลื่อน
5. มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการอัลกอริทึมย่อย (sub-algorithm misconceptions)	- การใช้อัลกอริทึมย่อยไม่ถูกต้อง เช่น ไม่สอดคล้องกับชนิดของตัวแปร - อัลกอริทึมย่อยไม่สามารถใช้ในการแก้ปัญหาได้ตามวัตถุประสงค์ - อื่น ๆ เกี่ยวกับอัลกอริทึมย่อย	เกี่ยวกับการแสดงผังงานโปรแกรม เป็นต้น
6. มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการแสดงผังงาน โปรแกรม (program flowchart presentation misconceptions)	- ใช้สัญลักษณ์สำหรับการสร้างผังงาน (flowchart) ที่ไม่ถูกต้อง - อื่น ๆ เกี่ยวกับการแสดงผังงานโปรแกรม	

สรุปผลการวิจัย

จากการวิเคราะห์ผังงานโปรแกรมที่นักเรียนได้สร้างขึ้นจำนวน 44 ผัง สามารถแสดงผลการวิจัยได้ดัง ตารางที่ 5

ตารางที่ 5 จำนวนผังงานโปรแกรมที่พบมโนทัศน์ที่คลาดเคลื่อนที่เกิดจากวิธีแก้ปัญหา ในการนำผังงานโปรแกรมมาประกอบเข้าด้วยกัน (plan composition problems) และ จากโครงสร้างภาษาและไวยากรณ์ของการเขียนโปรแกรม (construct-based problems)

ประเภทของมโนทัศน์ที่คลาดเคลื่อน	จำนวนผังงาน (เปอร์เซ็นต์จากทั้งหมด 44 ผัง)	
	construct-based problems	Plan- composition problems
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการตั้งเงื่อนไข (condition misconceptions)	84%	84%
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับลำดับ (sequence misconceptions)	70%	-
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการวนซ้ำ (loop misconceptions)	84%	84%
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการสั่งงาน (assignment misconceptions)	70%	-
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับอัลกอริทึมย่อย (sub-algorithm misconceptions)	84%	84%
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการตีความ (translation misconceptions)	-	11%
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการแสดงผังงาน (flowchart presentation misconceptions)	72%	-

อภิปรายผลการวิจัย

1. มโนทัศน์ที่คลาดเคลื่อนที่พบจากวิธีแก้ปัญหา ในการนำผังงานโปรแกรมมาประกอบเข้าด้วยกัน (plan-composition problems)

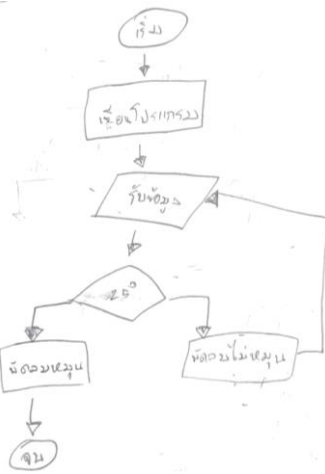
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการตั้งเงื่อนไข ที่พบมีจำนวนเท่ากับ 84% จากผังงานโปรแกรมทั้งหมด ดังตารางที่ 5 มโนทัศน์ที่

คลาดเคลื่อนเกี่ยวกับการตั้งเงื่อนไขมีลักษณะดังนี้

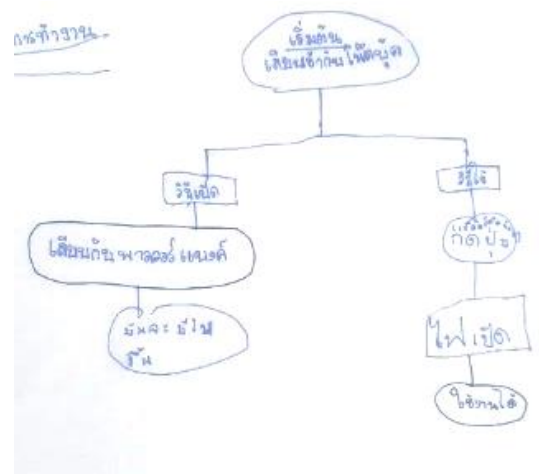
1. ไม่กำหนดเงื่อนไข ดังตัวอย่างในภาพที่ 1(a) ซึ่งเป็นมโนทัศน์ที่คลาดเคลื่อนซึ่งพบได้มากที่สุดสำหรับมโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการตั้งเงื่อนไข (condition) และมีความสอดคล้องกับรายงานผลการวิจัยที่ผ่านมา (Rahimi et al., 2017)



(a)



(b)



(c)

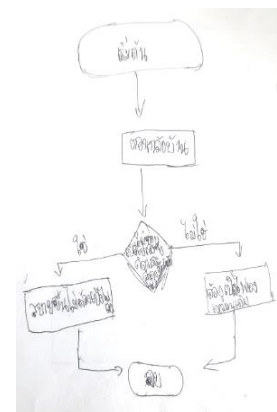
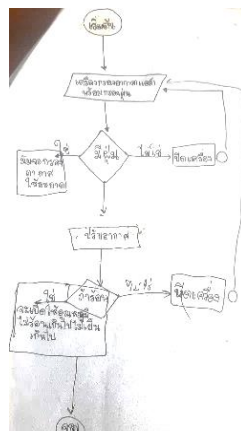
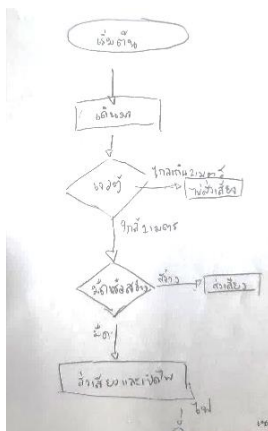
ภาพที่ 1 (a) ผังงานโปรแกรมที่ไม่มีการกำหนดเงื่อนไข (b) ผังงานโปรแกรมที่ไม่มีการพิจารณาการเท่ากันหรือเปรียบเทียบตัวแปรที่ได้รับมา (c) ผังงานโปรแกรมที่ไม่มีการกำหนดตัวแปรเริ่มต้นที่จะนำไปสู่การดำเนินการของชุดคำสั่งต่อไป

2. ไม่มีการพิจารณาการเท่ากันหรือเปรียบเทียบตัวแปรที่ได้รับมา(input) ดังแสดงตัวอย่างในภาพที่1(b) ซึ่งมีการรับข้อมูล แต่ไม่มีการนำข้อมูลดังกล่าวมาเปรียบเทียบกับค่าที่กำหนดไว้ (25 องศาเซลเซียส) ซึ่งเป็นอุณหภูมิที่เหมาะสมสำหรับผู้สูงอายุ

3. ไม่กำหนดตัวแปรเริ่มต้นที่จะนำไปสู่การดำเนินการของชุดคำสั่งต่อไป ดังแสดงตัวอย่างไว้ในภาพที่ 1(c) ผังงานถูกออกแบบให้แยกเป็นสองทาง

โดยไม่มีการกำหนดตัวแปรเริ่มต้นใด ๆ เพื่อนำไปประมวลผลเพื่อดำเนินการในขั้นตอนต่อไป

มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการวนซ้ำที่พบมีจำนวนเท่ากับ 84% จากผังงานทั้งหมด ดังตารางที่ 5 ซึ่งมีลักษณะที่สำคัญคือ 1. การตั้งเงื่อนไขของการวนซ้ำไม่ถูกต้อง 2. ไม่สามารถสร้างการวนซ้ำที่ซ้อนกันได้ 3. ไม่สามารถใช้การวนซ้ำแก้ปัญหาได้อย่างเหมาะสม ดังแสดงในภาพที่ 2(a) 2(b) และ 2(c) ตามลำดับ



(a)

(b)

(c)

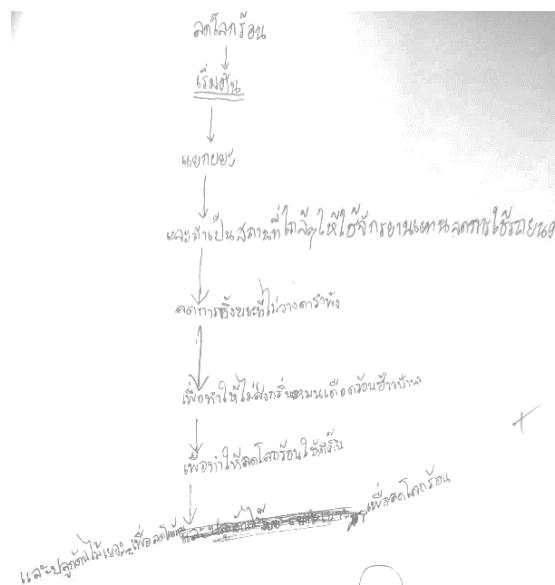
ภาพที่ 2 (a) ผังงานโปรแกรมที่ตั้งเงื่อนไขของการวนซ้ำไม่ถูกต้อง (b) ผังงานโปรแกรมที่ไม่สามารถสร้างการวนซ้ำที่ซ้อนกันได้ (c) ผังงานโปรแกรมที่ไม่สามารถใช้การวนซ้ำแก้ปัญหาได้อย่างเหมาะสม)

ในภาพที่ 2(a) นักเรียนต้องการให้เซนเซอร์วัดระยะสิ่งของที่อยู่ด้านหน้า ถ้าระยะน้อยกว่า 1 เมตร จะมีการส่งเสียงเตือน หากบริเวณดังกล่าวมีความเข้มแสงไม่พอ แสงสว่างจะสว่างขึ้น จากผังงานโปรแกรมดังกล่าวพบว่าไม่มีการกำหนดเงื่อนไขความเข้มแสงและระยะเพื่อให้มีการวนซ้ำเพื่อกลับมารับข้อมูลความเข้มแสงและระยะที่ตรวจวัดได้จากเซนเซอร์ ในภาพที่ 2(b) เป็นการวนซ้ำของการพิจารณาตัวแปร 2 ตัวคือ อุณหภูมิ และปริมาณฝุ่น เพื่อควบคุมการทำงานของเครื่องปรับอากาศที่สามารถกรองฝุ่นได้ จากผังงานโปรแกรมจะเห็นว่าตัวแปร ทั้ง 2 ตัว ไม่ได้ถูกนำมาพิจารณาพร้อมกัน หากแต่พิจารณาปริมาณฝุ่นก่อนแล้วจึงมาพิจารณาตัวแปรเกี่ยวกับอุณหภูมิโดยที่ไม่ได้กำหนดเงื่อนไขเริ่มต้น มีเพียงลูกศรขึ้นมาจากคำว่า “มีฝุ่น” เท่านั้น ซึ่งไม่แสดงถึงการตั้งเงื่อนไขเริ่มต้นแต่อย่างใด สำหรับภาพที่ 2(c) แสดงให้เห็นว่านักเรียนไม่สามารถระบุตัวแปรที่ควรนำมาใช้ในการวนซ้ำเพื่อเปิดและปิดไฟอัตโนมัติได้ ซึ่งมีโน้ตที่คลาดเคลื่อนเกี่ยวกับการวนซ้ำนี้มีความสอดคล้องกับงานวิจัยที่ผ่านมาของ Choi (2017)

มีโน้ตที่คลาดเคลื่อนเกี่ยวกับอัลกอริทึมย่อยที่ พบมีจำนวน เท่ากับ 84% จากผังงานโปรแกรมทั้งหมด ดังตารางที่ 5 โดยพิจารณาจากความเหมาะสมของผังงานโปรแกรมย่อยที่นำมาใช้ในการแก้ปัญหา ดังตัวอย่างในภาพที่ 2(b) ซึ่งผังงานโปรแกรมนี้แสดงการรับข้อมูลและการตรวจสอบเงื่อนไขที่ไม่เหมาะสมเมื่ออากาศร้อน ผังงานโปรแกรมควรแสดงเงื่อนไขเริ่มต้นว่า อากาศร้อนหมายถึงอุณหภูมิเท่าไร และควรมีผังงานโปรแกรม

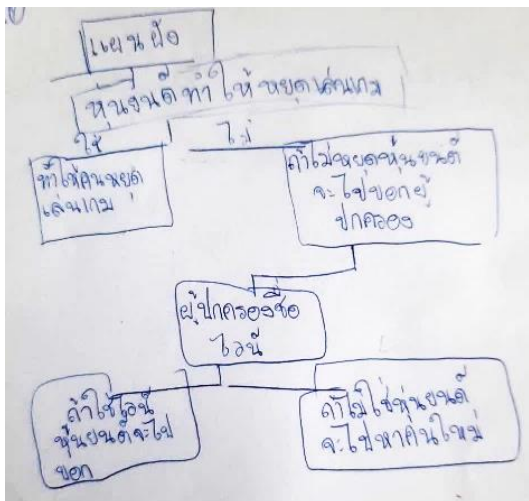
ที่เป็นการตรวจสอบอุณหภูมิที่เป็นข้อมูลรับเข้า (input) และเปรียบเทียบกับเงื่อนไขตั้งต้น ตลอดจนนำไปสู่ผังงานที่เป็นการดำเนินการต่อไป (Rahimi et al., 2017)

มีโน้ตที่คลาดเคลื่อนเกี่ยวกับการตีความที่พบมีจำนวนเท่ากับ 11% จากผังงานโปรแกรมทั้งหมดดังตารางที่ 5 มีลักษณะที่สำคัญคือ นักเรียนไม่สามารถเขียนผังงานโปรแกรมเพื่อแสดงการดำเนินการต่าง ๆ ได้ กรณีที่พบจากงานวิจัยนี้ นักเรียนไม่สามารถแสดงสัญลักษณ์เพื่อแสดงการตรวจสอบเงื่อนไขการรับข้อมูลหรือการวนซ้ำใด ๆ ได้ มีเพียงข้อความแสดงลำดับของ การดำเนินการเท่านั้น ซึ่งต่างจากผลการวิจัยที่รายงานโดย Rahimi et al.(2017) และ Choi (2017) ดังแสดงตัวอย่างในภาพที่ 3

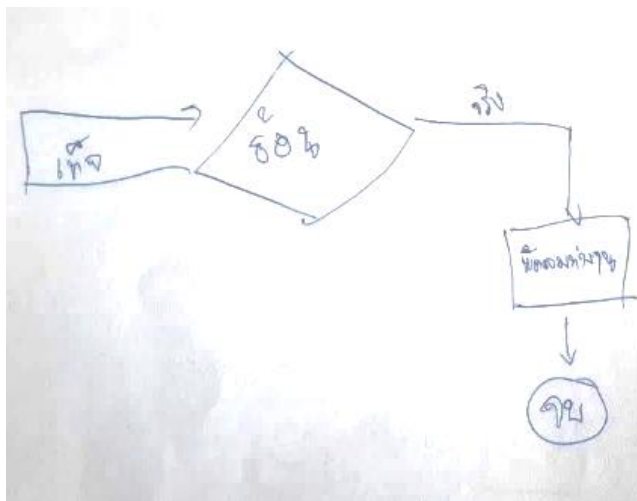


ภาพที่ 3 ลำดับของการดำเนินการต่าง ๆ ไม่สามารถนำมาแสดงเป็นผังงานโปรแกรมได้

2. มโนทัศน์ที่คลาดเคลื่อนที่พบจากโครงสร้างภาษาและไวยากรณ์ของการเขียนโปรแกรม (construct-based problems)



(a)



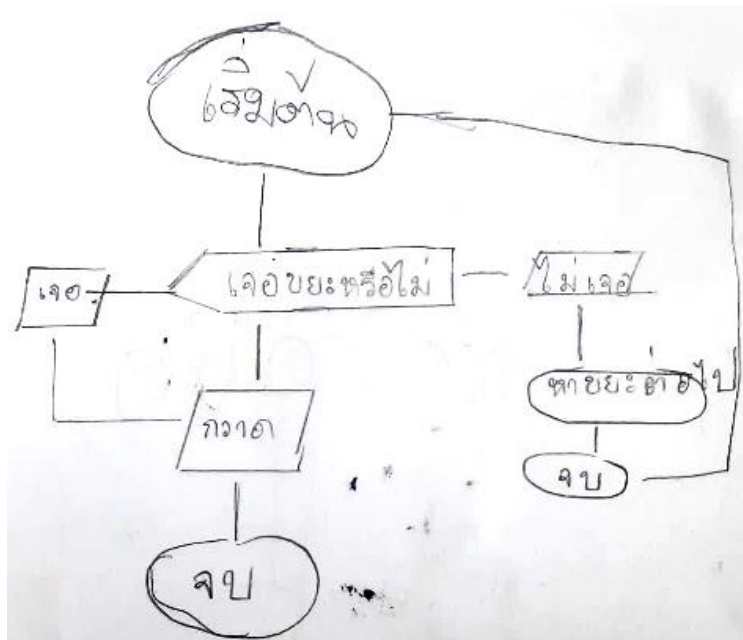
(b)

ภาพที่ 4 (a) ไม่มีส่วนที่มีการตรวจสอบเงื่อนไขที่เป็นเท็จหรือจริง (b) ไม่มีการระบุตัวแปรเริ่มต้นในผังงานโปรแกรม

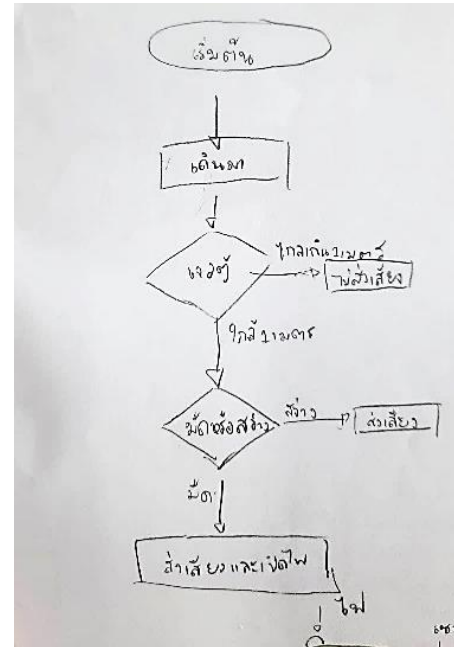
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการตั้งเงื่อนไขที่พบมีจำนวนเท่ากับ 84% จากผังงานโปรแกรมทั้งหมด ดังตารางที่ 5 ซึ่งมีลักษณะที่สำคัญคือ (1) การไม่มีส่วนที่มีการตรวจสอบเงื่อนไขที่เป็นเท็จหรือจริง (2) ไม่มีการระบุตัวแปรเริ่มต้นในผังงาน (Choi, 2017; Rahimi et al., 2017) ดังแสดงในภาพที่ 4(a) และ 4(b) ตามลำดับ

มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับลำดับที่พบมีจำนวน เท่ากับ 70% จากผังงานโปรแกรมทั้งหมด ดังตารางที่ 5 ซึ่งมีลักษณะดังนี้คือ 1. ลำดับของการดำเนินการในผังงานโปรแกรมมีลักษณะที่คล้ายกับ

การทำงานของมนุษย์ ดังตัวอย่างในภาพที่ 5(a) ซึ่งนักเรียนต้องการให้เซนเซอร์ตรวจจับระยะของขยะรอบถังขยะหากมีขยะอยู่รอบ ๆ ถังขยะมีการกวาดขยะแต่ถ้าไม่สามารถตรวจพบขยะจะทำการตรวจจับขยะต่อไป ซึ่งหากพิจารณาจาก ภาพที่ 5(a) ลำดับในการตรวจจับขยะและกำหนดระยะของขยะมีลักษณะเป็นภาษามนุษย์ที่ใช้สื่อสารกันในชีวิตประจำวันซึ่งไม่แสดงการระบุตัวแปรและเงื่อนไขที่สำคัญซึ่งเป็นภาษาในการเขียนโปรแกรมเกี่ยวกับระยะแต่อย่างใด (Choi, 2017; Rahimi et al., 2017)



(a)



(b)

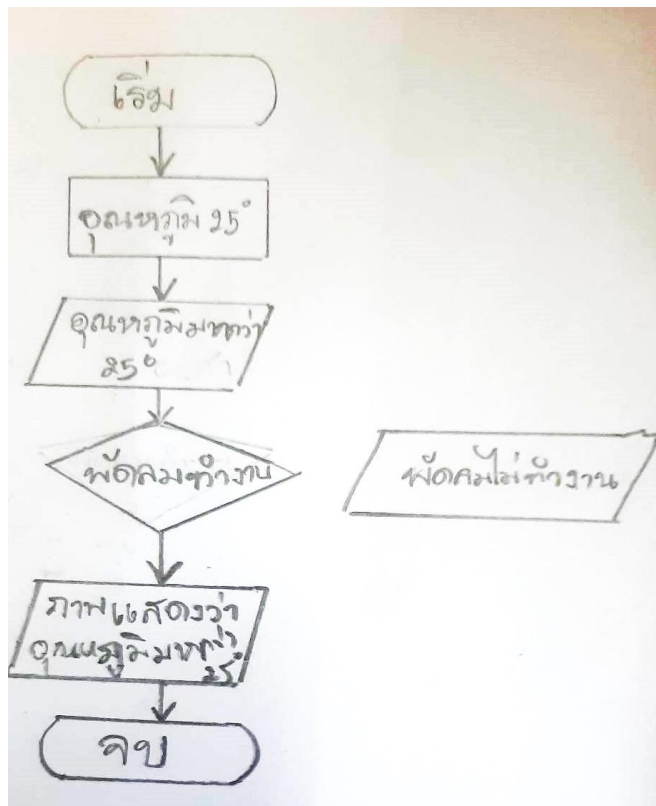
ภาพที่ 5 (a) ลำดับการทำงานในผังงานโปรแกรมเป็นลักษณะที่คล้ายกับการทำงานของมนุษย์ (b) ผังงานโปรแกรมที่แสดงการรับข้อมูลของตัวแปรสองตัวทีละครั้งซึ่งในความเป็นจริงแล้วลำดับของ การรับข้อมูลตัวแปรทั้งสองตัวสามารถทำได้พร้อมกัน

และ ภาพที่ 5(b) ซึ่งเป็นผังงานโปรแกรมที่นักเรียนต้องการแสดงการรับค่าของระยะสิ่งกีดขวางและความสว่างของบริเวณด้านหน้าผู้สูงอายุ เพื่อเตือนเป็นเสียงและเปิดไฟหากบริเวณดังกล่าวมีความสว่างไม่เพียงพอ ซึ่งจากการพิจารณาจะสังเกตได้ว่าไม่แสดงการรับตัวแปรของข้อมูลที่ชัดเจนตั้งแต่เริ่มต้น นอกจากนี้ยังเป็นการรับข้อมูลที่ทีละครั้ง ซึ่งในความเป็นจริงแล้วลำดับของการรับข้อมูล ตัวแปรทั้งสองตัวนั้นสามารถรับได้พร้อมกันและอาศัยความรู้เกี่ยวกับอัลกอริทึมเกี่ยวกับตัวดำเนินการ “และ” หรือ “หรือ” ในการตรวจสอบเงื่อนไขเพื่อดำเนินการในขั้นต่อไป(Choi, 2017; Rahimi et al., 2017)

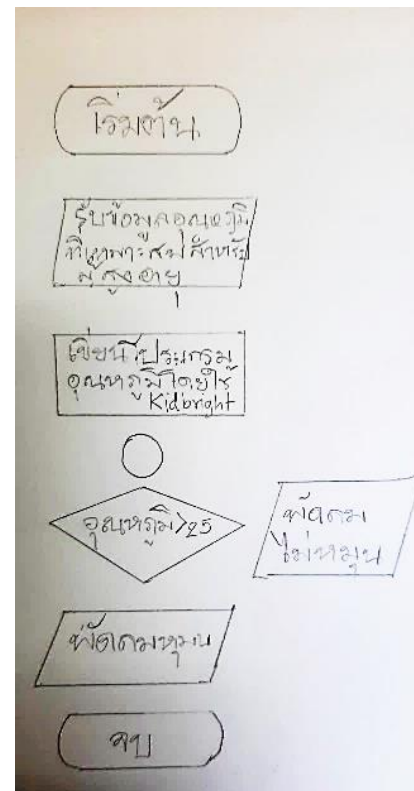
มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการวนซ้ำที่พบมีจำนวนเท่ากับ 84% จากผังงานโปรแกรมทั้งหมด ดังตารางที่ 5 โดยมีลักษณะที่สำคัญคือการ

ตรวจสอบตัวแปรควบคุมของ การวนซ้ำไม่ถูกต้องหรือไม่กำหนดให้ชัดเจน ดังตัวอย่างในภาพที่ 6(a) ซึ่งตัวแปรที่สำคัญที่ใช้ในการวนซ้ำเพื่อเปิดพัดลมคืออุณหภูมิที่มากกว่า 25 องศาเซลเซียส เนื่องจากตัวแปรที่ใช้ในการกำหนดการวนซ้ำไม่ถูกนำมาพิจารณาและตรวจสอบเงื่อนไข ดังนั้นจึงไม่สามารถสร้างการวนซ้ำที่สมบูรณ์ได้(Choi, 2017; Rahimi et al., 2017)

มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการสั่งงานที่พบมีจำนวนเท่ากับ 70% จากผังงานโปรแกรมทั้งหมด ดังตารางที่ 5 ซึ่งลักษณะสำคัญคือ (1) การสั่งงานมีการสลับลำดับและไม่เหมาะสมต่อการแก้ปัญหาเพื่อให้ไปถึงเป้าหมาย (2) ไม่สามารถแสดงลำดับขั้นของการสั่งงานต่าง ๆ ใน ผังงานโปรแกรมได้ซึ่งไม่พบในการรายงานจากการวิจัยอื่น ๆ ดังแสดงตัวอย่างในภาพที่ 6(b) เป็นต้น



(a)



(b)

ภาพที่ 6 (a) ผังงานโปรแกรมที่ไม่แสดงการตรวจสอบตัวแปรควบคุมของการวนซ้ำที่ชัดเจน (b) ผังงานโปรแกรมที่ไม่แสดงลำดับขั้นของการสั่งงานต่าง ๆ

มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับมโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับอัลกอริทึมย่อยที่พบมีจำนวนเท่ากับ 84% จากผังงานโปรแกรมทั้งหมด ดังตารางที่ 5 มีลักษณะที่สำคัญคือ นักเรียนไม่สามารถแสดงการส่งค่าของตัวแปรที่รับมา(input) ไปยังอัลกอริทึมย่อย นอกจากนี้นักเรียนบางคนยังไม่สามารถแสดงการส่งค่าของต่อแปรออก(output)จากอัลกอริทึมย่อยหนึ่งไปยังอีกอัลกอริทึมย่อยหนึ่งเพื่อให้มีการดำเนินการถัดไป (Choi, 2017; Rahimi et al., 2017)

มโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับการแสดงผังงานโปรแกรมที่พบมีจำนวนเท่ากับ 72% จากผังงาน

โปรแกรมทั้งหมด ดังตารางที่ 5 ซึ่งจากงานวิจัยนี้พบว่ามีความหลากหลายกว่ารายงานการวิจัยที่ผ่านมา (Choi, 2017; Rahimi, Barendsen, and Henze, 2017) เช่น นักเรียนใช้เครื่องหมายสื่เหลี่ยมผืนผ้าสลับกับสื่เหลี่ยมขนมเปียกปูนซึ่งแสดงถึงความสับสนของนักเรียนระหว่างสัญลักษณ์ที่เป็นการดำเนินการ(สื่เหลี่ยมผืนผ้า) และรับข้อมูล(สื่เหลี่ยมขนมเปียกปูน) ตามลำดับ นอกจากนี้ยังพบว่านักเรียนส่วนใหญ่ใช้เครื่องหมายตัดสินใจ(สื่เหลี่ยมข้าวหลามตัด) ไม่ถูกต้องโดยใช้เพื่อแสดงการรับข้อมูลซึ่งแทนด้วยสัญลักษณ์ สื่เหลี่ยมขนมเปียกปูน เป็นต้น

องค์ความรู้จากการวิจัยสู่การพัฒนา

จากการวิจัยนี้พบว่านักเรียนในระดับประถมศึกษาที่มีมโนทัศน์ที่คลาดเคลื่อนแนวคิดอัลกอริทึมในการออกแบบผังงานที่หลากหลาย ผังงานส่วนใหญ่ยังคงออกแบบบนพื้นฐานของการใช้ภาษามนุษย์ในการสื่อสารการส่งเสริมให้นักเรียนสามารถออกแบบผังงานได้อย่างถูกต้องและสอดคล้องกับอัลกอริทึมที่ใช้ในการแก้ปัญหาเป็นสิ่งที่จำเป็น (Chetty & van der Westhuizen, 2015) จะช่วยลดอุปสรรคในการเขียนโปรแกรมในระดับที่สูงขึ้นและทำให้การเรียนรู้ในชั้นเรียนเกี่ยวกับการเขียนโปรแกรมมีความราบรื่นมากยิ่งขึ้น (Choi, 2017)

ข้อเสนอแนะ

ข้อเสนอแนะทั่วไป

1. ผลการวิจัยนี้เป็นผลการวิจัยที่ได้จากการศึกษากลุ่มตัวอย่าง เฉพาะเท่านั้น ไม่สามารถนำมาใช้แทนประชากรของนักเรียนระดับประถมศึกษาในประเทศไทยได้

2. ลักษณะมโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับแนวคิดอัลกอริทึมที่ปรากฏในแต่ละลักษณะสามารถนำไปใช้ออกแบบกิจกรรมที่สามารถส่งเสริมให้นักเรียนมีมโนทัศน์ที่ถูกต้องได้

ข้อเสนอแนะในการทำวิจัยครั้งต่อไป

1. เกณฑ์ที่ใช้วิเคราะห์เพื่อจัดกลุ่มมโนทัศน์ที่คลาดเคลื่อนนั้นสามารถปรับเปลี่ยนตามลักษณะของมโนทัศน์ที่ปรากฏของแต่ละงานวิจัย เนื่องจากมโนทัศน์ที่คลาดเคลื่อนที่พบในการวิจัยครั้งต่อไปอาจเป็นมโนทัศน์ที่ไม่ปรากฏในรายงานการวิจัยที่ผ่านมา

2. ลักษณะมโนทัศน์ที่คลาดเคลื่อนเกี่ยวกับแนวคิดอัลกอริทึมที่ปรากฏในแต่ละลักษณะเป็นข้อมูลพื้นฐานที่สำคัญในการส่งเสริมความสามารถในการเขียนโปรแกรม ดังนั้นหากเปิดโอกาสให้กลุ่มตัวอย่างแต่ละคน ได้ออกแบบผังงานโปรแกรมของตนเอง จะทำให้ทราบถึงมโนทัศน์ที่คลาดเคลื่อนอย่างหลากหลายมากขึ้น

เอกสารอ้างอิง

- An, S., & Lee, Y. (2015). Designing a Programming Course for Enhancing Pre-service Teachers' Programming PCK. In *EdMedia+ Innovate Learning* (pp. 222-225). Association for the Advancement of Computing in Education (AACE).
- Barendsen, E., Grgurina, N., & Tolboom, J. (2016). A new informatics curriculum for secondary education in the Netherlands. In *International conference on informatics in schools: Situation, evolution, and perspectives* (pp. 105-117). Springer, Cham.
- Bohl, M. (1971). *Flowcharting techniques*. Science Research Associates, Chicago.
- Chetty, J., & van der Westhuizen, D. (2015). Towards a pedagogical design for teaching novice programmers: design-based research as an empirical determinant for success. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research* (pp. 5-12).
- Clancy, M. (2005). Misconceptions and attitudes that interfere with learning to program. In *Computer science education research* (pp. 95-110). Taylor & Francis.
- Choi, Y. (2017). Analysis and Application of Misconception Flowchart for Programming Control Structure Concept Learning. *Journal of Korea Multimedia Society*, 20(12), 2000-2008.
- Elo, S., & Kyngäs, H. (2008). The qualitative content analysis process. *Journal of advanced nursing*, 62(1), 107-115.
- Mangaroska, K., Sharma, K., Gašević, D., & Giannakos, M. (2022). Exploring students' cognitive and affective states during problem solving through multimodal data: Lessons learned from a programming activity. *Journal of Computer Assisted Learning*, 38(1), 40-59.
- Putnam, R. T., Sleeman, D., Baxter, J. A., & Kuspa, L. K. (1986). A summary of misconceptions of high school Basic programmers. *Journal of Educational Computing Research*, 2(4), 459-472.
- Rahimi, E., van den Berg, J. and Veen, W. (2013). Investigating teachers' perceptions about the educational benefits of Web 2.0 personal learning environments. *eLearning Papers*, (35), 1-23.
- Rahimi, E., Barendsen, E., & Henze, I. (2017). Identifying students' misconceptions on basic algorithmic concepts through flowchart analysis. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 155-168). Springer, Cham.
- Robins, A., Rountree, J. and Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2), 137-172.

- Scanlan, D. A. (1989). Structured flowcharts outperform pseudocode: An experimental comparison. *IEEE software*, 6(5), 28-36.
- Seppälä, O., Malmi, L. and Korhonen, A. (2006). Observations on student misconceptions-A case study of the Build-Heap Algorithm. *Computer Science Education*, 16(3), 241-255.
- Shneiderman, B., Mayer, R., McKay, D. and Heller, P. (1977). Experimental investigations of the utility of detailed flowcharts in programming. *Communications of the ACM*, 20(6), 373-381.
- Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational researcher*, 15(2), 4-14.
- Sirkiä, T. and Sorva, J. (2012). Exploring programming misconceptions: an analysis of student mistakes in visual program simulation exercises. *Proceedings of the 12th Koli Calling International Conference on Computing Education Research* (pp. 19-28).
- Sorva, J. (2007). Notional machines and introductory programming education. *Transactions on Computing Education*, 13(2), 1-31.
- Spohrer, J. C. and Soloway, E. (1986). Novice mistakes: Are the folk wisdoms correct?. *Communications of the ACM*, 29(7), 624-632.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Zhao, L., Liu, X., Wang, C., & Su, Y. S. (2022). Effect of different mind mapping approaches on primary school students' computational thinking skills during visual programming learning. *Computers & Education*, 181, 104445.