



โครงการฝึกทักษะการเขียนโปรแกรม ครั้งที่ 1
โปรแกรมสำหรับจัดเก็บข้อมูลบัตรนักศึกษา ด้วยเครื่องมือ Array List

นายจิรเมธ พัวพันธ์
รหัสนิสิต 60160157

เสนอ
ผู้ช่วยศาสตราจารย์นวลศรี เต็มวัฒนา
อาจารย์พจน์สพร แซ่ลิ้ม

โครงการนี้เป็นส่วนหนึ่งของ
รายวิชา 88823359 โครงสร้างข้อมูลและขั้นตอนวิธีสำหรับวิศวกรรมซอฟต์แวร์
ภาคการศึกษาที่ 1 ปีการศึกษา 2561 สาขาวิชาวิศวกรรมซอฟต์แวร์
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

สารบัญ

เรื่อง

หน้า

สารบัญ	ก
สารบัญรูปภาพ.....	ข
สารบัญตาราง	ค
ส่วนที่ 1 การวิเคราะห์โจทย์ปัญหา	1
1.1 โจทย์ปัญหา.....	1
1.2 วิเคราะห์แนวทางแก้ไขปัญหา	1
1.3 การออกแบบ.....	1
1.4 รหัสเทียม	4
ส่วนที่ 2 รายละเอียดของผลลัพธ์	8
2.1 หน้าจอรระบบ.....	8
ส่วนที่ 3 การวิเคราะห์และประเมินผลลัพธ์.....	11
3.1 ประสิทธิภาพของโปรแกรม	11
3.2 สรุปผลลัพธ์	15

สารบัญรูปภาพ

ภาพที่	หน้า
ภาพที่ 1-1 แผนภาพคลาสของโครงงาน	2
ภาพที่ 2-1 การกำหนดจำนวนข้อมูลของนักศึกษาที่ต้องการจัดเก็บ	8
ภาพที่ 2-2 ส่วนการแสดงผลของเมนูระบบ	8
ภาพที่ 2-3 การเพิ่มข้อมูลนักศึกษา	9
ภาพที่ 2-4 การค้นหาข้อมูลนักศึกษา	9
ภาพที่ 2-5 การยืนยันว่าต้องการแก้ไขข้อมูลหรือไม่	9
ภาพที่ 2-6 ส่วนการแก้ไขข้อมูลนักศึกษา	10
ภาพที่ 2-7 ส่วนการแสดงผลข้อมูลนักศึกษาทั้งหมด	10
ภาพที่ 3-1 โปรแกรมส่วนของมอดูลการแสดงผลเมนูระบบ	11
ภาพที่ 3-2 โปรแกรมส่วนของมอดูลค้นหาข้อมูล	13
ภาพที่ 3-3 โปรแกรมส่วนของมอดูลแสดงผลข้อมูล	14

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1-1 แสดงส่วนประกอบของคลาส ARRAYLIST	2
ตารางที่ 1-2 แสดงส่วนประกอบของคลาส STUDENT	3
ตารางที่ 3-1 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลการแสดงผลของเมนูระบบ	12
ตารางที่ 3-2 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลค้นหาข้อมูล	13
ตารางที่ 3-3 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลแสดงข้อมูล	14

ส่วนที่ 1

การวิเคราะห์โจทย์ปัญหา

1.1 โจทย์ปัญหา

โปรแกรมสำหรับจัดเก็บข้อมูลบัตรนักศึกษา ด้วยเครื่องมือ Array List

ความสามารถของโปรแกรม

1. ผู้ใช้งานสามารถเพิ่มหรือจัดเก็บข้อมูลส่วนตัวของนักศึกษาแต่ละคน โดยมีข้อมูลที่ต้องจัดเก็บ ดังนี้
 - 1.1 รหัสนักศึกษา
 - 1.2 คำนาม
 - 1.3 ชื่อ – นามสกุล
 - 1.4 วันที่ออกบัตร
 - 1.5 วันที่หมดอายุบัตร
 - 1.6 คณะที่กำลังศึกษา
- หมายเหตุ : ในส่วนการจัดเก็บข้อมูลนักศึกษานี้ ต้องดำเนินการแบบเชิงวัตถุ (Class) เท่านั้น
2. ผู้ใช้งานสามารถกำหนดจำนวนข้อมูลของนักศึกษาที่ต้องการจัดเก็บได้ (กรณีเริ่มต้นโปรแกรม)
3. ผู้ใช้งานสามารถค้นหาข้อมูลบัตรของนักศึกษาได้
4. ผู้ใช้งานสามารถแก้ไขข้อมูลบัตรของนักศึกษาได้
5. ผู้ใช้งานสามารถเรียกดูข้อมูลของบัตรนักศึกษาทั้งหมดที่มีอยู่ได้

1.2 วิเคราะห์แนวทางแก้ไขปัญหา

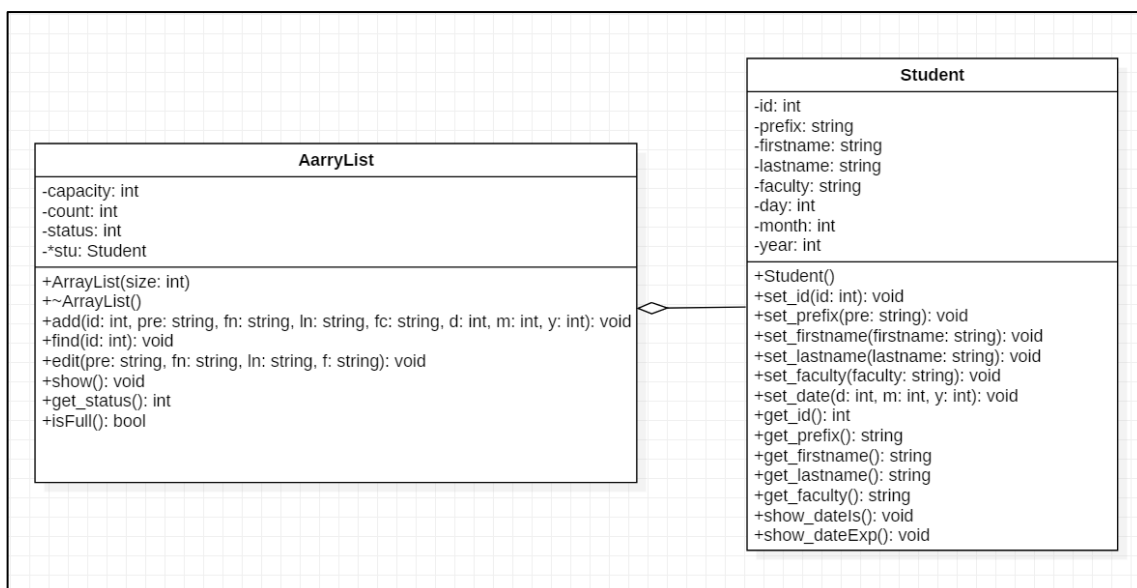
จากโจทย์วิเคราะห์ได้ว่าจะมีการการสร้างคลาสจำนวน 2 คลาส ในคลาสที่หนึ่งจัดเก็บข้อมูลส่วนตัวของนักศึกษา ควรมี Attribute (ตัวแปร) ที่เป็นชนิด String เพื่อเก็บข้อมูลที่เป็นอักขระ และชนิด Int เพื่อเก็บข้อมูลที่เป็นจำนวนเต็ม และในคลาสที่สอง เป็นคลาส ArrayList ใช้เพื่อเก็บข้อมูลเป็นแถวแต่เป็นลำดับแบบชัดเจน โดยทำหน้าที่สำหรับเก็บเพิ่ม ลบ และการค้นหาข้อมูล

1.3 การออกแบบ

จากการวิเคราะห์นำมาสร้างแผนภาพ ทำให้ได้แผนภาพคลาส ดังหัวข้อที่ 1.3.1

1.3.1 แผนภาพคลาส (Class Diagram)

แผนภาพคลาสแสดงได้ดังภาพที่ 1-1 แผนภาพคลาสของโครงงาน รวมถึงการแสดงค่าต่างๆ ในคลาสดังตารางที่ 1-1 และ ตารางที่ 1-2 แสดงส่วนประกอบของคลาส ArrayList และ Student



ภาพที่ 1-1 แผนภาพคลาสของโครงการ

ตารางที่ 1-1 แสดงส่วนประกอบของคลาส ArrayList

ข้อมูล	คำอธิบาย
Attribute	
capacity int	ตัวแปรสำหรับใช้ระบุความจุสูงสุดในการเก็บข้อมูล
count int	ตัวแปรสำหรับใช้นับจำนวนข้อมูลที่เก็บไว้ทั้งหมด
status int	ตัวแปรสำหรับการเช็คเงื่อนไขใน main
*stu Student	ตัวแปรใช้ในการเก็บค่าของคลาส Student
Method	
ArrayList(sizer int)	ค่า Constructor
~ArrayList()	ค่า Deconstructor
add(id:int,pre:string,fn:string,ln:string,fc:string,d:int,m:int,y:int) : void	เป็น Method สำหรับเพิ่มข้อมูล
find(id:int) : void	เป็น Method สำหรับค้นหาข้อมูล
edit(pre:string,fn:string,ln:string,f:string) : void	เป็น Method สำหรับแก้ไขข้อมูล
show() : void	เป็น Method สำหรับแสดงข้อมูล
get_status() : int	เป็น Method สำหรับคืนค่าไปยัง main
isFull() : bool	เป็น Method สำหรับเช็คค่าว่าง

ตารางที่ 1-2 แสดงส่วนประกอบของคลาส Student

ข้อมูล	คำอธิบาย
Attribute	
Id int	ตัวแปรสำหรับการเก็บข้อมูลรหัสนักศึกษา
prefix string	ตัวแปรสำหรับการเก็บข้อมูลคำนำหน้าชื่อ
firstname string	ตัวแปรสำหรับการเก็บข้อมูลชื่อจริง
lastname string	ตัวแปรสำหรับการเก็บข้อมูลนามสกุล
faculty string	ตัวแปรสำหรับการเก็บข้อมูลคณะ
day int	ตัวแปรสำหรับการเก็บข้อมูลวันที่ออกบัตร
month int	ตัวแปรสำหรับการเก็บข้อมูลเดือนที่ออกบัตร
year int	ตัวแปรสำหรับการเก็บข้อมูลปีที่ออกบัตร
Method	
Student()	ค่า Constructor
set_id(id:int) : void	เป็น Method สำหรับกำหนดค่าให้ตัวแปร id
set_prefix(pre:string) : void	เป็น Method สำหรับกำหนดค่าให้ตัวแปร prefix
set_firstname(firstname:string) : void	เป็น Method สำหรับกำหนดค่าให้ตัวแปร firstname
set_lastname(lastname:string) : void	เป็น Method สำหรับกำหนดค่าให้ตัวแปร lastname
set_faculty(faculty:string) : void	เป็น Method สำหรับกำหนดค่าให้ตัวแปร faculty
set_date(d:int,m:int,y:int) : void	เป็น Method สำหรับกำหนดวันที่ออกบัตร
get_id() : int	เป็น Method สำหรับคืนค่าตัวแปร id
get_prefix() : string	เป็น Method สำหรับคืนค่าตัวแปร prefix
get_firstname() : string	เป็น Method สำหรับคืนค่าตัวแปร firstname
get_lastname() : string	เป็น Method สำหรับคืนค่าตัวแปร lastname
get_faculty() : string	เป็น Method สำหรับคืนค่าตัวแปร faculty
show_datels() : void	เป็น Method สำหรับแสดงวันที่ออกบัตร
show_dateExp() : void	เป็น Method สำหรับแสดงวันที่บัตรหมดอายุ

1.4 รหัสเทียม

รหัสเทียมจะแสดงเขียนแยกตาม Method หรือ Module ของโปรแกรม

1.4.1 มอดูลการแสดงผลเมนู

อัลกอริทึม	มอดูลการแสดงผลเมนูของโปรแกรม
ข้อมูลนำเข้า	ไม่มี
ผลลัพธ์	แสดงเมนู

1. ประกาศตัวแปรชื่อ choice
2. ลูป
 - 2.1. แสดงเมนู
 - 2.2. เลือกเมนูในช่วงที่กำหนด
 - 2.3. ถ้าเลือกเมนูในช่วงที่กำหนด จบลูป
 - 2.4. ถ้าไม่ กลับไปที่ 2.1
3. ทำงานตามเมนูที่ระบุ
4. จบการทำงาน

1.4.2 มอดูลการเพิ่มข้อมูลนักศึกษา

อัลกอริทึม	มอดูลการเพิ่มข้อมูลนักศึกษา
ข้อมูลนำเข้า	รับค่ารหัสนิสิต คำนำหน้าชื่อ ชื่อ-นามสกุล คณะ และวันที่จากแป้นพิมพ์
ผลลัพธ์	จัดเก็บข้อมูลไว้ในระบบ

1. ประกาศตัวแปรชื่อ id, day, month, year เป็นชนิดจำนวนเต็ม
2. ประกาศตัวแปรชื่อ prefix, firstname , lastname, faculty เป็นชนิดอักขระ
3. กำหนดค่าให้ตัวแปรผ่านแป้นพิมพ์
 - 3.1 กำหนดค่าให้ตัวแปร id
 - 3.2 กำหนดค่าให้ตัวแปร prefix
 - 3.3 กำหนดค่าให้ตัวแปร firstname
 - 3.4 กำหนดค่าให้ตัวแปร lastname
 - 3.5 กำหนดค่าให้ตัวแปร faculty
 - 3.6 กำหนดค่าให้ตัวแปร day
 - 3.7 กำหนดค่าให้ตัวแปร month
 - 3.8 กำหนดค่าให้ตัวแปร year
4. เรียกใช้ Method add จากคลาส ArrayList โดยมีพารามิเตอร์ เป็นตัวแปรที่รับผ่านแป้นพิมพ์
5. การทำงานภายใน Method add
 - 5.1 เข้าเงื่อนไข ถ้าจำนวนความจุข้อมูลทั้งหมดยังว่างอยู่

- 5.1.1 เรียกใช้ Method set_id ของ Object stu จากคลาส Student ช่องที่ count โดยมีพารามิเตอร์เป็น id
- 5.1.2 เรียกใช้ Method set_prefix ของ Object stu จากคลาส Student ช่องที่ count โดยมีพารามิเตอร์เป็น prefix
- 5.1.3 เรียกใช้ Method set_firstname ของ Object stu จากคลาส Student ช่องที่ count โดยมีพารามิเตอร์เป็น firstname
- 5.1.4 เรียกใช้ Method set_lastname ของ Object stu จากคลาส Student ช่องที่ count โดยมีพารามิเตอร์เป็น lastname
- 5.1.5 เรียกใช้ Method set_faculty ของ Object stu จากคลาส Student ช่องที่ count โดยมีพารามิเตอร์เป็น faculty
- 5.1.6 เรียกใช้ Method set_date ของ Object stu จากคลาส Student ช่องที่ count โดยมีพารามิเตอร์เป็น day month และ year
- 5.1.7 เพิ่มค่าของตัวแปร count ขึ้น 1 จำนวน
- 5.2 เข้าเงื่อนไข ถ้าจำนวนความจุข้อมูลทั้งหมดเต็ม
 - 5.2.1 แสดงข้อความ Is Full

6. จบการทำงาน

1.4.3 มอดูลการค้นหาข้อมูลนักศึกษา

อัลกอริทึม	มอดูลการค้นหาข้อมูลนักศึกษา
ข้อมูลนำเข้า	รับค่ารหัสนิสิตจากแป้นพิมพ์
ผลลัพธ์	แสดงข้อมูลนิสิตที่ค้นหา

1. ประกาศตัวแปรชื่อ id เป็นชนิดจำนวนเต็ม
2. กำหนดค่าให้ตัวแปร id ผ่านแป้นพิมพ์
3. เรียกใช้ Method find จากคลาส ArrayList โดยมีพารามิเตอร์ เป็นตัวแปรที่รับผ่านแป้นพิมพ์
4. การทำงานภายใน Method find
 - 4.1 ประกาศตัวแปรชื่อ check เป็นชนิดจำนวนเต็ม
 - 4.2 เข้าลูป
 - 4.2.1 เข้าเงื่อนไข ถ้า id เท่ากับ Method set_date ของ Object stu จากคลาส Student ช่องที่ i
 - 4.2.1.1 กำหนดค่า check ให้เท่ากับค่า i
 - 4.3 เข้าเงื่อนไข ถ้า check ไม่เท่ากับ -1
 - 4.3.1 แสดงข้อมูลส่วนตัวของนักศึกษาทั้งหมดในช่องที่ check โดยเรียกผ่าน Method get Object stu จากคลาส Student

5. จบการทำงาน

1.4.4 มอดูลการแก้ไขข้อมูลนักศึกษา

อัลกอริทึม	มอดูลการแก้ไขข้อมูลนักศึกษา
ข้อมูลนำเข้า	รับค่ารหัสนิสิต คำนำหน้าชื่อ ชื่อจริง นามสกุล และ คณะจากแป้นพิมพ์
ผลลัพธ์	จัดเก็บข้อมูลนิสิตที่แก้ไข

1. ประกาศตัวแปรชื่อ id และ status เป็นชนิดจำนวนเต็ม
2. กำหนดค่าให้ตัวแปร id ผ่านแป้นพิมพ์
3. เรียกใช้ Method find จากคลาส ArrayList โดยมีพารามิเตอร์ เป็นตัวแปรที่รับผ่านแป้นพิมพ์
4. การทำงานภายใน Method find
 - 4.1 ประกาศตัวแปรชื่อ check เป็นชนิดจำนวนเต็ม
 - 4.2 เข้าลูป
 - 4.2.1 เข้าเงื่อนไข ถ้า id เท่ากับ Method set_date ของ Object stu จากคลาส Student ช่องที่ i
 - 4.2.1.1 กำหนดค่า check ให้เท่ากับค่า i
 - 4.3 เข้าเงื่อนไข ถ้า check ไม่เท่ากับ -1
 - 4.3.1 แสดงข้อมูลส่วนตัวของนักศึกษาทั้งหมดในช่องที่ check โดยเรียกผ่าน Method get Object stu จากคลาส Student
 - 4.3.2 แสดงข้อความผ่านทางหน้าจอว่า "Do you want to edit the information? [Y|N]: "
 - 4.3.3 ประกาศตัวแปร yes เป็นชนิดอักขระ
 - 4.3.4 กำหนดค่าให้ตัวแปร yes ผ่านแป้นพิมพ์
 - 4.3.5 เข้าเงื่อนไข ถ้า yes มีค่าเท่ากับ "Y"
 - 4.3.5.1 กำหนดค่าให้ status มีค่าเท่ากับ check
5. เข้าสู่การทำงานของ main
6. เข้าเงื่อนไข ถ้า -1 ไม่เท่ากับ Method get_status จากคลาส ArrayList
 - 6.1 ประกาศตัวแปรชื่อ prefix, firstname, lastname, faculty เป็นชนิดอักขระ
 - 6.1.1 กำหนดค่าให้ตัวแปร prefix
 - 6.1.2 กำหนดค่าให้ตัวแปร firstname
 - 6.1.3 กำหนดค่าให้ตัวแปร lastname
 - 6.1.4 กำหนดค่าให้ตัวแปร faculty
7. เรียกใช้ Method edit จากคลาส ArrayList โดยมีพารามิเตอร์ เป็นตัวแปรที่รับผ่านแป้นพิมพ์

8. การทำงานภายใน Method edit

8.1 เรียกใช้ Method set_prefix ของ Object stu จากคลาส Student ช่องที่ status โดยมีพารามิเตอร์เป็น prefix

8.2 เรียกใช้ Method set_firstname ของ Object stu จากคลาส Student ช่องที่ status โดยมีพารามิเตอร์เป็น firstname

8.3 เรียกใช้ Method set_lastname ของ Object stu จากคลาส Student ช่องที่ status โดยมีพารามิเตอร์เป็น lastname

8.4 เรียกใช้ Method set_faculty ของ Object stu จากคลาส Student ช่องที่ status โดยมีพารามิเตอร์เป็น faculty

8.5 กำหนดค่า status ให้มีค่าเท่ากับ -1

8.6 แสดงข้อความผ่านทางหน้าจอ "-> Edit Complete!!!"

9. จบการทำงาน

1.4.5 มอดูลการแสดงผลข้อมูลนักศึกษาทั้งหมด

อัลกอริทึม	มอดูลการแสดงผลข้อมูลนักศึกษา
ข้อมูลนำเข้า	ไม่มี
ผลลัพธ์	แสดงข้อมูลส่วนตัวนักศึกษา

1. เรียกใช้ Method show จากคลาส ArrayList

2. การทำงานภายใน Method show

2.1 ลูป

2.1.1 เข้าเงื่อนไข ถ้า -1 ไม่เท่ากับ Method get_id ของ Object stu จากคลาส Student ช่องที่ i

2.1.1.1 แสดงข้อมูลส่วนตัวของนักศึกษาทั้งหมดในช่องที่ check โดยเรียกผ่าน Method get ของ Object stu จากคลาส Student ในช่องที่ i

3. จบการทำงาน

ส่วนที่ 2

รายละเอียดของผลลัพธ์

2.1 หน้าจอระบบ

2.1.1 การแสดงผลของเมนูระบบ

เมื่อเริ่มเปิดโปรแกรม โปรแกรมจะแสดงให้ระบุ จำนวนข้อมูลนักศึกษาที่ต้องการจัดเก็บ ดังภาพที่ 2- 1

```
Enter the number of students you want to store : █
```

ภาพที่ 2-1 การกำหนดจำนวนข้อมูลของนักศึกษาที่ต้องการจัดเก็บ

โปรแกรมจะแสดงข้อมูลเมนูระบบ ดังภาพที่ 2 – 2 โดย การใช้งานคือ ให้ระบุ เลขตัวเลือก 0 1 2 และ 3 โดย แต่ละ ตัวเลขที่การทำงานที่แตกต่างการ ดังนี้

- 0. Exit คือ ออกจากโปรแกรม
- 1. Add Student Info คือ การเพิ่มข้อมูลนักศึกษา
- 2. Search & Edit Student Info คือ การค้นหาและแก้ไขข้อมูลนักศึกษา
- 3. Show All Student Info คือ การแสดงข้อมูลนักศึกษาที่มีทั้งหมด

```
=====
                        MENU
      1.Add Student Info
      2.Search & Edit Student Info
      3.Show All Student Info
      0.EXIT
=====
Please choose menu: █
```

ภาพที่ 2-2 ส่วนการแสดงผลของเมนูระบบ

2.1.2 การเพิ่มข้อมูลนักศึกษา

เมื่อเข้าสู่เมนูที่ 1 โปรแกรมจะแสดงผล ดังภาพที่ 2 – 3 โดย การใช้งานคือ ให้ระบุข้อมูลดังนี้

- 1. รหัสนิสิต
- 2. คำนำหน้า
- 3. ชื่อจริง
- 4. นามสกุล
- 5. คณะ
- 6. วันที่ปัจจุบัน

```

=====
                Add Student Info
id : 60160157
prefix : Mr
firstname : Jiramate
lastname : Phuaphan
faculty : Informatics
date [for example] -> 20 14 2018 : 11 9 2561
->[0] ID:60160157 ADD Complete.
=====

```

ภาพที่ 2-3 การเพิ่มข้อมูลนักศึกษา

2.1.3 การค้นหาข้อมูลนักศึกษา

เมื่อเข้าสู่เมนูที่ 2 โปรแกรมจะแสดงผล ดังภาพที่ 2 – 4 โดย การใช้งานคือ ให้ระบุข้อมูลรหัสนักศึกษาที่มีอยู่ในระบบ เพื่อทำการค้นหา

หากค้นหาไม่เจอข้อมูลรหัสนักศึกษา โปรแกรมจะกลับไปหน้าจอเมนู ดังภาพที่ 2 - 2

Enter Student ID : 60160157							
Search Student Info							
NO.	ID	Prefix	Firstname	Lastname	Faculty	Date Issue	Date Expiration
*	60160157	Mr	Jiramate	Phuaphan	Informatics	11/9/2561	11/9/2565

ภาพที่ 2-4 การค้นหาข้อมูลนักศึกษา

2.1.4 การแก้ไขข้อมูลนักศึกษา

เมื่อได้ผลลัพธ์ ดังภาพที่ 2 – 4 จะมีข้อความให้ระบุ เพื่อยืนยันว่าต้องการแก้ไข ดังภาพที่ 2 – 5 โดยวิธีใช้งานคือ ให้ระบุข้อมูลที่เป็นตัวอักษรดังนี้

Y คือ ยืนยันเพื่อไปหน้าแก้ไขข้อมูล

N คือ กลับหน้าจอเมนู

```

Do you want to edit the information? [Y|N]: █

```

ภาพที่ 2-5 การยืนยันว่าต้องการแก้ไขข้อมูลหรือไม่

เมื่อทำการระบุ Y จะได้ผลลัพธ์ ดังภาพที่ 2 – 6 โดย การใช้งานคือ ให้แก้ไขข้อมูลดังนี้

1. คำนำหน้า
2. ชื่อจริง
3. นามสกุล
4. คณะ

```

=====
                        Edit Student Info
id : 60160157
prefix : Mr
firstname : Jiramate
lastname : Phuaphan
faculty : Informatics

-> Edit Complete!!!

=====
  
```

ภาพที่ 2-6 ส่วนการแก้ไขข้อมูลนักศึกษา

2.1.5 การแสดงข้อมูลนักศึกษาทั้งหมด

เมื่อเข้าสู่เมนูที่ 3 โปรแกรมจะแสดงผล ดังภาพที่ 2 – 7

Show Student							
NO.	ID	Prefix	Firstname	Lastname	Faculty	Date Issue	Date Expiration
1	60160157	Mr	Jiramate	Phuaphan	Informatics	11/9/2561	11/9/2565

ภาพที่ 2-7 ส่วนการแสดงข้อมูลนักศึกษาทั้งหมด

ส่วนที่ 3

การวิเคราะห์และประเมินผลลัพธ์

3.1 ประสิทธิภาพของโปรแกรม

เป็นส่วนการวิเคราะห์ข้อมูล ที่ได้จากผลลัพธ์ของโปรแกรม ซึ่งประกอบไปด้วยมอดูล การแสดงผลเมนูของระบบ การค้นหาข้อมูล และการแสดงข้อมูล โดยมีการวิเคราะห์ในแต่ละส่วนได้ดังนี้

3.1.1 มอดูลการแสดงผลของเมนูระบบ

ในส่วนมอดูลการแสดงผลของเมนูระบบมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3-1 และทำการวิเคราะห์ได้ดังตารางที่ 3 – 1

```
do {
    cout << "\n =====" << endl;
    cout << "\t\tMENU" << endl;
    cout << "\t1.Add Student Info" << endl;
    cout << "\t2.Search & Edit Student Info"<<endl;
    cout << "\t3.Show All Student Info" << endl;
    cout << "\t0.EXIT" << endl;
    cout << " =====" << endl;
    cout << " Please choose menu: " ;
    cin >> choice;
    cout << endl;
    if(choice == 1) {
        cout << " =====" << endl;
        cout << "\t Add Student Info" << endl;
        cout << " id : ";
        cin >> id;
        cout << " prefix : ";
        cin >> pre;
        cout << " firstname : ";
        cin >> fn;
        cout << " lastname : ";
        cin >> ln;
        cout << " faculty : ";
        cin >> fc;
        cout << " date [for example] -> 20 14 2018 : ";
        cin >> d;
        cin >> m;
        cin >> y;
        ary.add(id,pre,fn,ln,fc,d,m,y);
    }
    if(choice == 2) {
        cout << " Enter Student ID : ";
        cin >> id;
        ary.find(id);
        if(ary.get_status() != -1) {
            cout << " =====" << endl;
            cout << "\t Edit Student Info" << endl;
            cout << " id : " << id << endl;
            cout << " prefix : ";
            cin >> pre;
            cout << " firstname : ";
            cin >> fn;
            cout << " lastname : ";
            cin >> ln;
            cout << " faculty : ";
            cin >> fc;
            ary.edit(pre,fn,ln,fc);
        }
    }
    if(choice == 3) {
        ary.show();
    }
} while(choice != 0);
```

ภาพที่ 3-1 โปรแกรมส่วนของมอดูลการแสดงผลเมนูระบบ

ตารางที่ 3-1 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลการแสดงผลของเมนูระบบ

ค่า choice	ตรวจสอบเงื่อนไข choice != 0	แสดงผลข้อมูลภายใน loop
1	/	/
2	/	/
3	/	/
0	/	X
จำนวนครั้งที่ทำ	4	3
ถ้า n ครั้ง	n	n -1

สรุปได้ว่า Big-O ของการทำงานมอดูลนี้คือ $(n) + (n - 1) = 2n - 1$
 $= \text{BigO}(n)$

3.1.2 มอดูลค้นหาข้อมูล

ในส่วนมอดูลการค้นหาข้อมูลมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3 - 2 และทำการวิเคราะห์ได้ดังตารางที่ 3 - 2

```
int check = -1;
for(int i=0; i<capacity; i++) {
    if(stu[i].get_id() == id) {
        check = i;
    }
}
```

ภาพที่ 3-2 โปรแกรมส่วนของมอดูลค้นหาข้อมูล

กำหนดให้ค่า capacity เท่ากับ 5

ตารางที่ 3-2 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลค้นหาข้อมูล

ค่า capacity	ตรวจสอบเงื่อนไข $i < \text{capacity}$	แสดงผลข้อมูลภายใน loop
0	/	/
1	/	/
2	/	/
3	/	/
4	/	/
5	/	X
จำนวนครั้งที่ทำ	6	5
ถ้า n ครั้ง	$n - 1$	$n - 2$

สรุปได้ว่า Big-O ของการทำงานมอดูลนี้คือ $(n - 1) + (n - 2) = 2n - 3$
 $= \text{BigO}(n)$

3.1.3 มอดูลแสดงข้อมูล

ในส่วนมอดูลการแสดงผลข้อมูลมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3 - 3 และทำการวิเคราะห์ได้ดังตารางที่ 3 - 3

```
for(int i=0; i<capacity; i++) {
    if(stu[i].get_id() != -1) {
        cout << setw(8) <<left << i+1;
        cout << setw(16) <<left << stu[i].get_id() ;
        cout << setw(8) <<left << stu[i].get_prefix() ;
        cout << setw(16) <<left << stu[i].get_firstname();
        cout << setw(16) <<left << stu[i].get_lastname();
        cout << setw(8) <<left << stu[i].get_faculty();
        stu[i].show_dateIs();
        stu[i].show_dateExp();
        cout<<endl;
    }
}
```

ภาพที่ 3-3 โปรแกรมส่วนของมอดูลแสดงข้อมูล

กำหนดให้ค่า capacity เท่ากับ 5

ตารางที่ 3-3 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานของมอดูลแสดงข้อมูล

ค่า capacity	ตรวจสอบเงื่อนไข $i < \text{capacity}$	แสดงผลข้อมูลภายใน loop
0	/	/
1	/	/
2	/	/
3	/	/
4	/	/
5	/	X
จำนวนครั้งที่ทำ	6	5
ถ้า n ครั้ง	$n - 1$	$n - 2$

สรุปได้ว่า Big-O ของการทำงานของมอดูลนี้คือ $(n - 1) + (n - 2) = 2n - 3$
 $= \text{BigO}(n)$

3.2 สรุปผลลัพธ์

จากการวิเคราะห์การทำงานของโปรแกรม พบว่ามอดูลทุกมอดูลมีประสิทธิภาพเท่ากันในโปรแกรม
เนื่องจาก โปรแกรมมีการใช้ loop เพื่อวนเช็คเงื่อนไขและแสดงผลเท่านั้น