

ใบงานการทดลองที่ 2

เรื่อง ภาษาจาวากับการโปรแกรมเชิงวัตถุ

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการโปรแกรมเชิงวัตถุด้วยคำสั่งพื้นฐานในภาษาจาวา
- 1.2. รู้และเข้าใจเพื่อศึกษาและทดลองสร้างโปรแกรมเชิงวัตถุโดยใช้ภาษาการโปรแกรมเชิงวัตถุใหม่ๆ

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

3.1. คำสั่งที่ใช้ในการแสดงผลบนแถบ Console คือคำสั่งใด

```
console.log("JavaScript");
```

```
console.log(12);
```

```
console.log(true);
```

3.2. คำสั่งการคอมเมนต์แบบบรรทัดเดียว และแบบหลายบรรทัด คืออะไร ?

1. Line Comment เป็น Comment บรรทัดเดียว ใช้เครื่องหมาย // ซึ่ง Comment ประเภทนี้จะมีผลต่อบรรทัดหรือข้อความที่อยู่หลัง เครื่องหมาย // เพียงบรรทัดเดียวเท่านั้น ดังนั้นถ้าต้องการทำ comment หลาย ๆ บรรทัด จึงต้องเขียน // ในทุก ๆ บรรทัดที่ทำ Comment

2. Block Comment เป็น Comment หลายบรรทัด ที่ใช้ได้หลายบรรทัด การเขียน Comment ประเภทนี้ ใช้เครื่องหมาย /* และ */ ข้อความใดที่อยู่ในเครื่องหมาย /* และ */ จะเป็น Comment ทั้งหมด เช่น

3.3. ประเภทข้อมูลแบบ Integer คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

ข้อมูลชนิดเลขจำนวนเต็ม (Integer) คือ เลขจำนวนเต็มทั่วไป ไม่ว่าจะเป็นเลขจำนวนเต็มบวก จำนวนเต็มศูนย์และจำนวนเต็มลบ ซึ่งเลขจำนวนเต็มเหล่านี้ สามารถนำไปคำนวณได้ ตัวอย่าง เช่น 100, 56, 0, -20 เป็นต้น

3.4. ประเภทข้อมูลแบบ Double คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

ตัวแปรแบบ double เป็นตัวแปรที่เก็บข้อมูลที่เป็นเลขทศนิยมเหมือนกับ float แต่จะ ใช้พื้นที่ในการเก็บมากกว่าเดิม 2 เท่า คือมีขนาด 8 ไบต์ มีค่าระหว่าง $1.7E-308$ ถึง $1.7E+308$.

3.5. ประเภทข้อมูลแบบ Float คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

ข้อมูลชนิดตัวเลขทศนิยม (Float) คือ เลขทศนิยมชนิดคงที่ หรืออาจจะเป็นทศนิยม แบบไม่รู้จัก หรืออาจจะเป็นเลขทศนิยมที่เขียนในรูป E (หรือ e) ยกกำลัง ตัวเลขทศนิยมเหล่านี้ สามารถนำมาใช้ในการคำนวณได้ ตัวอย่าง เลขทศนิยมนี้ได้แก่ 20.25, -0.60, 58.96, $5.40e04$ เป็นต้น

3.6. ประเภทข้อมูลแบบ Bool คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

ข้อมูลแบบบูลีน คือ ข้อมูลที่เก็บค่าความเป็นจริง คือ TRUE กับค่าความเป็นเท็จ คือ FALSE ใช้เนื้อที่ในหน่วยความจำ 1 ไบต์ ต่อ 1 ค่า ข้อมูลแบบบูลีนนี้ไม่สามารถรับข้อมูลจากคีย์บอร์ดเข้ามาได้ทั้งนี้ เนื่องจก่าใช้ในการกำหนดหรือสร้างสถานะเพื่อใช้เป็นเงื่อนไขในโปรแกรม แต่ข้อมูลแบบบูลีนสามารถแสดงผลออกทางจอภาพหรือทางเครื่องพิมพ์ได้เช่นเดียวกับข้อมูลแบบอื่น ๆ การกำหนดตัวแปรชนิดนี้ต้องกำหนดด้วย “BOOLEAN”

3.7. ประเภทข้อมูลแบบ Char คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

คือชนิดข้อมูลประเภท char เป็นชนิดข้อมูลที่สามารถเก็บข้อมูลได้ 1 ตัวอักษรเท่านั้น โดยใช้หน่วยความจำ 1 ไบต์ (8 บิต) รูปแบบ char (variable name); เช่น char a; ประกาศตัวแปรชื่อ a.

3.8. ประเภทข้อมูลแบบ String คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

ข้อมูลชนิดข้อความ (String) เป็นข้อมูลแบบตัวอักษรที่มีความยาวมากกว่า 1 ตัวอักษร มาเรียงต่อกันเป็นข้อความ โดยที่ข้อความนั้นจะต้องถูกเขียนไว้ในเครื่องหมาย " " (Double Quote) ตัวอย่างเช่น "Phitsanulok", "Welcome" เป็นต้น

3.9. Widening Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Widening Casting คือการแปลงชนิดข้อมูลที่มีขนาดเล็กไปเป็นชนิดข้อมูลขนาดใหญ่

```
int number1 = 100;
long number2 = number1;
double number3 = number2;
System.out.println(number1);
System.out.println(number2);
System.out.println(number3);
```

```
run:
100
100
100.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

3.10. Narrowing Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เป็นการกำหนดค่าของ Variable ที่มี Data Type ลำดับสูงกว่า ให้กับ Variable ที่มี Data Type ต่ำกว่า ซึ่งการทำ Type Casting ประเภทนี้ต้อง ใส่ keyword (Data Type ที่ต้องการ) ให้กับ Variable ที่มี Data Type ลำดับสูงกว่า

```
double number1 = 100;
long number2 = (long)number1;
int number3 = (int)number2;
System.out.println(number1);
System.out.println(number2);
System.out.println(number3);
```

```
run:
100.0
100
100
BUILD SUCCESSFUL (total time: 0 seconds)
```

3.11. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการทางคณิตศาสตร์ดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
+	บวก (Addition)	$X + y$
-	ลบ (Subtraction)	$\text{sum}(k - 1);$
*	คูณ (Multiplication)	$X * y$
/	หาร (Division)	X / y
++	เพิ่มค่าอีกหนึ่งให้กับ a ก่อน แล้วจึงนำค่าใหม่ของ a ไปใช้งาน	$++a$
--	ลดค่าลงหนึ่งให้กับ a ก่อน แล้วจึงนำค่าใหม่ของ a ไปใช้งาน	$--a$
%	หารเอาผลลัพธ์เฉพาะเศษ (Modulus)	$X \% y$

3.12. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อกำหนดค่าดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
=	กำหนดค่าให้เท่ากับ (Assignment)	$\text{int } x = 10$
+=	การเพิ่มค่า (Addition)	$a+=b$ มาจาก $a = a+b$
-=	การลบค่า (Subtraction)	$a-=b$ มาจาก $a = a-b$
=	การคูณ (Multiply)	$a=b$ มาจาก $a = a*b$
/=	การหารได้ผลลัพธ์จำนวนเต็ม (Devide)	$a/=b$ มาจาก $a = a/b$
%=	การหารได้ผลลัพธ์เศษ (Devide)	$a\%=b$ มาจาก $a = a\%b$
&=	และ (and)เท่ากับ	$a \& = 2;$
=	หรือ (or)เท่ากับ	$a = 2;$

3.13. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อเปรียบเทียบดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
==	เท่ากับ (Equal)	if(x == 10)
>	มากกว่า (Greather Than)	X > y
>=	มากกว่าหรือเท่ากับ (Greather Than or Equal)	X >= y
<	น้อยกว่า (Less Than)	if(x < 20)
<=	น้อยกว่าหรือเท่ากับ (Less Than or Equal)	X <= y
!=	ไม่เท่ากับ (Not Equal)	X != y

3.14. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการตรรกศาสตร์ดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
&&	และ (and)	inStock > 0 && money
	หรือ (or)	a b
!	ไม่ใช่ (not)	!a

3.15. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if พร้อมยกตัวอย่างประกอบ

```
if (inStock > 0 && money >= 100) {
    money -= 100;
    printf("You can buy this iPad with 100 USD.\n");
    printf("You have left %d USD\n", money);
}
```

นี่เป็นการกำหนดเงื่อนไขจากที่กล่าวมาเราได้ตรวจสอบว่ามีสินค้าคงเหลือในตัวแปร inStock หรือไม่ และมีเงินมีมากกว่าหรือเท่ากับ 100 USD หรือไม่ โดยการเชื่อมเงื่อนไขย่อยทั้งสองด้วยตัวดำเนินการ AND ถ้าหากเงื่อนไขเป็นจริงโปรแกรมจะทำงานในบล็อกของคำสั่ง if ทำการหักเงินออกไป และแสดงข้อความแจ้งให้ทราบออกทางหน้าจอ

3.16. ประโยกเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if else พร้อมยกตัวอย่างประกอบ

```
Login
Username: Metin
Password: 1234
Hi Metin,
Login success.
```

```
Login
Username: Metin
Password: abcd
Invalid username or password.
```

เมื่อโปรแกรมเริ่มต้นการทำงาน มันจะถามให้ผู้ใช้กรอก Username และรหัสผ่าน และเก็บไว้ในตัวแปร String สองตัวได้แก่ username และ password ตามลำดับ สำหรับนำไปตรวจสอบว่าค่าที่กรอกเข้ามานั้นถูกต้องหรือไม่ด้วยคำสั่ง if else

3.17. ประโยกเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if, else if และ else พร้อมยกตัวอย่างประกอบ

```
#include <stdio.h>

int main()
{
    int n = -5;
    if (n > 0) {
        printf("n is positive.\n");
    } else if (n < 0) {
        printf("n is negative.\n");
    } else {
        printf("n is zero.\n");
    }
    return 0;
}
```

เนื่องจากค่าในตัวแปร n ที่เป็น -5 นั้นทำให้เงื่อนไขในคำสั่ง else if เป็นจริง ดังนั้นโปรแกรมจึงทำงานในบล็อกของคำสั่งที่สอง และแสดงข้อความบอกว่าตัวเลขเป็นจำนวนเต็มลบ คุณสามารถลองเปลี่ยนตัวเลขในตัวแปรเป็นจำนวนเต็มบวกหรือศูนย์และรันโปรแกรมเพื่อดูผลลัพธ์อีกครั้ง

3.18. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if แบบ Short hand พร้อมยกตัวอย่างประกอบ

```
$is_premium_user = ($user['permissions'] == 'premium' ? true : false);
```

การใช้เงื่อนไข Short hand เพื่อเช็คเงื่อนไขก่อนเก็บค่าตัวแปร

3.19. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง switch พร้อมยกตัวอย่างประกอบ

```
#include <stdio.h>

int main()
{
    int n = 2;
    switch (n) {
        case 1:
            printf("One\n");
            break;
        case 2:
            printf("Two\n");
            break;
        case 3:
            printf("Three\n");
            break;
        default:
            printf("Unknown\n");
    }
    return 0;
}
```

Two

เราได้ประกาศตัวแปร n สำหรับเก็บค่าของตัวเลขจำนวนเต็ม จากนั้นใช้คำสั่ง switch case เพื่อตรวจสอบค่าในตัวแปรนี้ เนื่องจากค่าในตัวแปรตรงกับเงื่อนไขที่กำหนดใน case 2 ดังนั้นโปรแกรมทำงานคำสั่งของเคสดังกล่าวโดยแสดงข้อความ "Two" ออกทางหน้าจอ

3.20. การวนรอบ – จงเขียน Syntax การใช้งานคำสั่ง for พร้อมยกตัวอย่างประกอบ

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    for (int i = 1; i <= 10; i++) {
```

```
        printf("%d\n", i);
```

```
    }
```

```
    return 0;
```

```
}
```

1

2

3

4

5

6

7

8

9

10

นี่เป็นโปรแกรมพื้นฐานอย่างง่ายสำหรับแสดงตัวอย่างการทำงานของลูปในภาษา C เราได้ใช้คำสั่ง for loop สำหรับนับและแสดงตัวเลขจาก 1-10 ออกทางหน้าจอ

3.21. การวนรอบ – จงเขียน Syntax การใช้งานคำสั่ง while พร้อมยกตัวอย่างประกอบ

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int n = 1;
```

```
    while (n <= 10) {
```

```
        printf("%d\n", n);
```

```
        n++;
```

```
    }
```

```
    printf("Loop ended\n", n);
```

```
    return 0;
```

```
}
```

1

2

3

4

5

6

7

8

9

10

Loop ended

นี่เป็นตัวอย่างพื้นฐานสำหรับการใช้งานคำสั่ง while loop ลูป เราได้ใช้มันนับตัวเลขจาก 1-10 และแสดงค่าออกทางหน้าจอ

3.22. การวนรอบ – จงเขียน Syntax การใช้งานคำสั่ง do while พร้อมยกตัวอย่างประกอบ

```
#include <stdio.h>

int main()
{
    int n = 1;
    do {
        printf("%d\n", n);
        n++;
    } while (n <= 10);
    return 0;
}
```

1
2
3
4
5
6
7
8
9
10

เริ่มต้นเราได้ประกาศตัวแปร n ที่มีค่าเป็น 1 จากนั้นโปรแกรมทำงานในลูปแต่ละรอบโดยแสดงค่าในตัวแปรออกทางหน้าจอ หลังจากจบลูปในแต่ละรอบเงื่อนไข $n \leq 10$ จะถูกตรวจสอบ ถ้าหากเป็นจริงมันจะทำงานในรอบถัดไป หรือจบการทำงานของลูปหากเงื่อนไขไม่เป็นจริง

3.23. อธิบายการทำงานของคำสั่ง break พร้อมยกตัวอย่างประกอบ

```
#include <stdio.h>

int main()
{
    for (int i = 1; i <= 10; i++) {
        if (i == 6) {
            break;
        }
        printf("%d ", i);
    }
    return 0;
}
```

1 2 3 4 5

ในตัวอย่างนี้ เราได้เขียนโปรแกรมเพื่อนับเลขจาก 1-10 โดยการใช้คำสั่งวนซ้ำ for loop และอย่างที่ คุณเห็นว่าตัวเลขที่นับและแสดงออกมานั้นมีเพียง 1-5 เท่านั้น

3.24. อธิบายการทำงานของคำสั่ง continue พร้อมยกตัวอย่างประกอบ

```
#include <stdio.h>

int main()
{
    for (int i = 1; i <= 10; i++) {
        if (i % 2 == 1) {
            continue;
        }
        printf("%d ", i);
    }
    return 0;
}
```

2 4 6 8 10

ในตัวอย่างนี้ เราใช้คำสั่ง for loop เพื่อวนแสดงตัวเลขจาก 1-10 แต่ผลลัพธ์จากการทำงานของโปรแกรมนั้นแสดงเพียงเลขคู่เท่านั้น และนี่ถูกต้องเนื่องจากเราสั่งให้มันทำโดยการใช้งานคำสั่ง continue เพื่อให้ข้ามการทำงานของลูป

3.25. อธิบายการทำงานของคำสั่ง Enum

Enum เป็นประเภทข้อมูลที่เป็นชุดของตัวเลขจำนวนเต็ม มันใช้กำหนดกลุ่มของค่าคงที่ที่ที่มีความหมายและเข้าใจได้ง่ายขึ้นในการเขียนโปรแกรม โดยค่าที่ใช้ในการประกาศ enum นั้นจะเป็น String literal ใดๆ โดยค่าเหล่านี้ถือว่าเป็นค่าคงที่ของจำนวนเต็ม รูปแบบการใช้ enum ในภาษา C# เป็นดังนี้:

```
enum Name{ Value1, Value2, ... };
```

4. ลำดับขั้นการปฏิบัติการ

4.1. จงแก้โจทย์ปัญหาดังต่อไปนี้

4.1.1. จงเขียนโปรแกรมเพื่อแสดงเลข 1 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้และแสดงคำตอบบรรทัดละ 5 ตัวเลขเท่านั้น

Test case 1	Test case 2
Please enter your value : 8 1, 2, 3, 4, 5 6, 7, 8	Please enter your value : 21 1, 2, 3, 4, 5 6, 7, 8, 9, 10 11, 12, 13, 14, 15 16, 17, 18, 19, 20 21

4.1.2. ฟังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามฟังงาน)

ผังงาน

```

graph TD
    Start([package LAB222]) --> Import[import java.util.Scanner]
    Import --> Class[public class lab2]
    Class --> Main[public static void main(String[] args)]
    Main --> Arg[arg]
    Arg --> Number[int Number]
    Number --> Scanner[Create a Scanner object  
Scanner myObj = new Scanner(System.in)]
    Scanner --> PrintPrompt[System.out.print("Please enter your value : ")]
    PrintPrompt --> ReadInput[Read user input  
Number = myObj.nextInt()]
    ReadInput --> PrintSep[System.out.println("-----")]
    PrintSep --> InitI[int i=1]
    InitI --> LoopCond{i <= Number}
    LoopCond -- True --> PrintI[System.out.print(i)]
    PrintI --> Div5Cond{i % 5 == 0}
    Div5Cond -- True --> PrintNewline[System.out.println("")]
    Div5Cond -- False --> PrintComma[System.out.print(", ")]
    PrintNewline --> IncI[i++]
    PrintComma --> IncI
    IncI --> LoopCond
    LoopCond -- False --> End([End])
    
```

โค้ดโปรแกรม

```

lab2.java X
1 package LAB222;
2
3 import java.util.Scanner;
4
5 public class lab2 {
6     public static void main(String[] args) {
7         int Number ;
8
9         Scanner myObj = new Scanner(System.in); // Create a Scanner object
10        System.out.print("Please enter your value : ");
11
12        Number = myObj.nextInt(); // Read user input
13        System.out.println("-----");
14
15        for(int i=1; i<=Number; i++) {
16            System.out.print(i);
17
18            if(i%5==0) {
19                System.out.println("");
20            }else {
21                System.out.print(" ,");
22            }
23        }
24    }
25 }
26
27
28 }
29

```

Problems Javadoc Declaration Console X

<terminated> lab2 (1) [Java Application] C:\Users\Augsupan tharawan\p2\pool\plugins\org.eclipse.justi.openjdkhots

Please enter your value : 8

1 , 2 , 3 , 4 , 5

6 , 7 , 8 ,

Problems Javadoc Declaration Console X

<terminated> lab2 (1) [Java Application] C:\Users\Augsupan tha

Please enter your value : 21

1 , 2 , 3 , 4 , 5

6 , 7 , 8 , 9 , 10

11 , 12 , 13 , 14 , 15

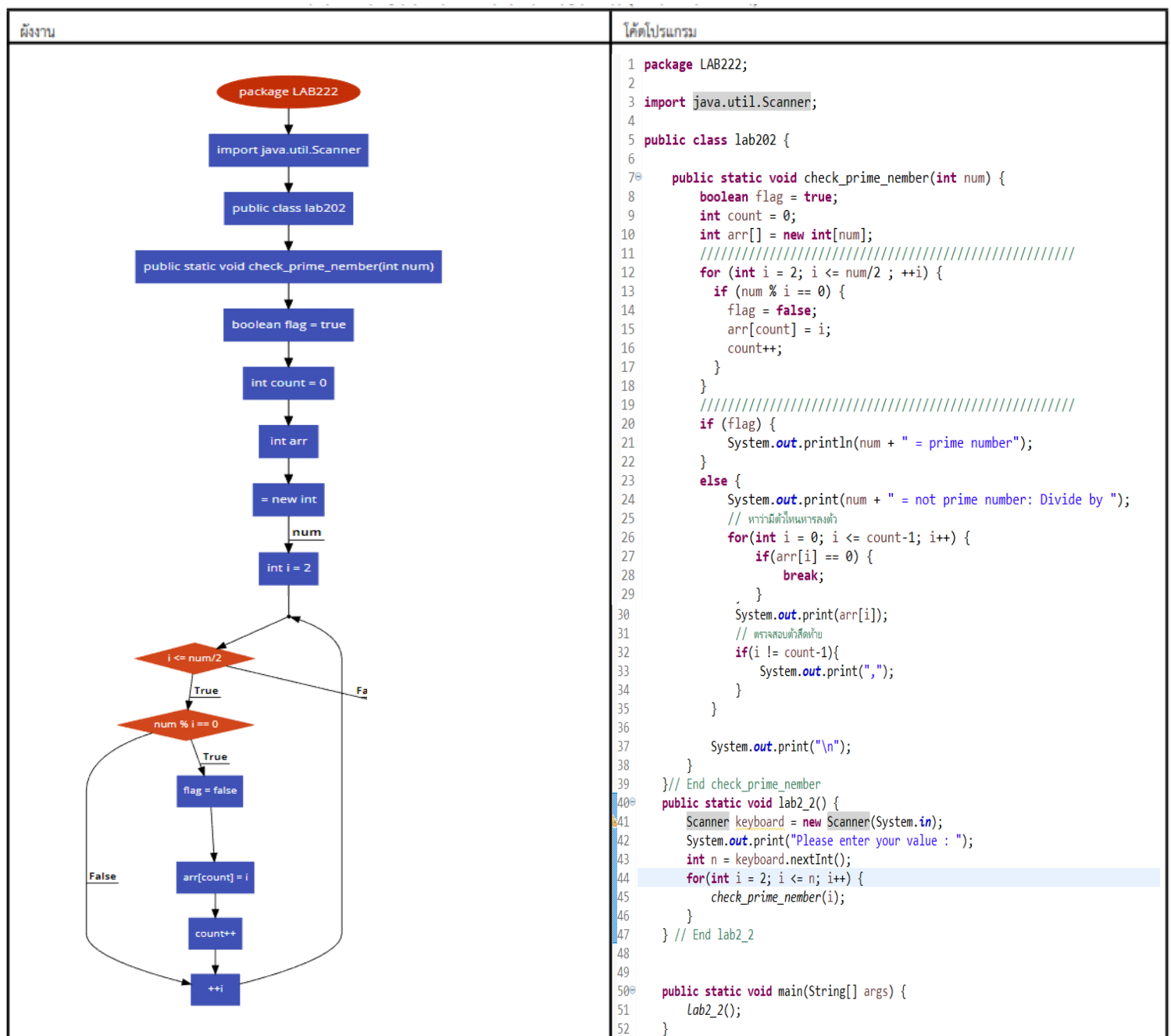
16 , 17 , 18 , 19 , 20

21 ,

4.2. จงแก้ไขข้อปัญหาดังต่อไปนี้

4.2.1. จงเขียนโปรแกรมเพื่อแสดงจำนวนเฉพาะที่อยู่ระหว่าง เลข 2 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้

Test case 1	Test case 2
Please enter your value : 8	Please enter your value : 16
2 = Prime Number	2 = Prime Number
3 = Prime Number	3 = Prime Number
4 = Not Prime : Divide by 2	4 = Not Prime : Divide by 2
5 = Prime Number	5 = Prime Number
6 = Not Prime : Divide by 2, 3	6 = Not Prime : Divide by 2, 3
7 = Prime Number	7 = Prime Number
8 = Not Prime : Divide by 2, 4	8 = Not Prime : Divide by 2, 4
	9 = Not Prime : Divide by 3
	10 = Not Prime : Divide by 2, 5
	11 = Prime Number
	12 = Not Prime : Divide by 2, 3, 4, 6
	13 = Prime Number
	14 = Not Prime : Divide by 2, 7
	15 = Not Prime : Divide by 2, 3, 5
	16 = Not Prime : Divide by 2, 4, 8



<terminated> lab202 [Java Application] C:\Users\Augsupan t

Please enter your value : 8
 2 = prime number
 3 = prime number
 4 = not prime number: Divide by 2
 5 = prime number
 6 = not prime number: Divide by 2,3
 7 = prime number
 8 = not prime number: Divide by 2,4

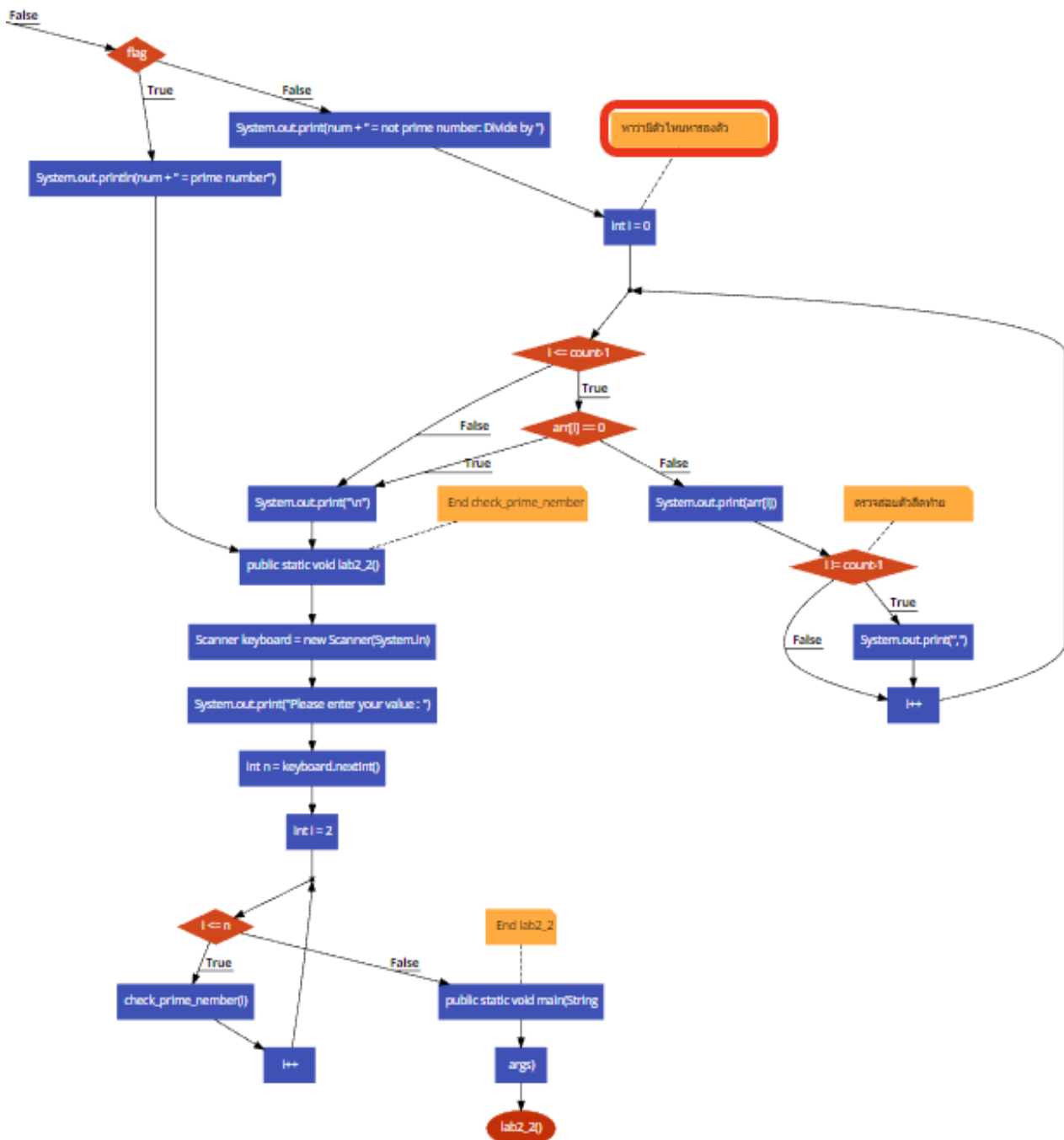
Problems @ Javadoc Declaration Console x

<terminated> lab202 [Java Application] C:\Users\Augsupan tharawan\.

Please enter your value : 16
 2 = prime number
 3 = prime number
 4 = not prime number: Divide by 2
 5 = prime number
 6 = not prime number: Divide by 2,3
 7 = prime number
 8 = not prime number: Divide by 2,4
 9 = not prime number: Divide by 3
 10 = not prime number: Divide by 2,5
 11 = prime number
 12 = not prime number: Divide by 2,3,4,6
 13 = prime number
 14 = not prime number: Divide by 2,7
 15 = not prime number: Divide by 3,5
 16 = not prime number: Divide by 2,4,8

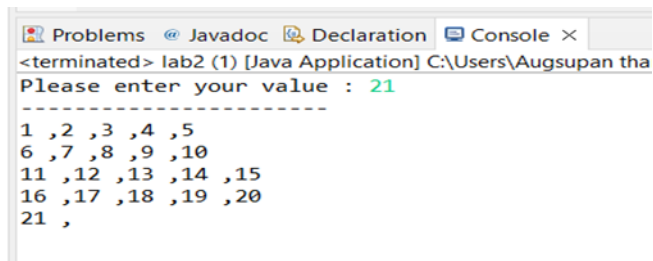
ผังงาน

โค้ดโปรแกรม



5. สรุปผลการปฏิบัติการ

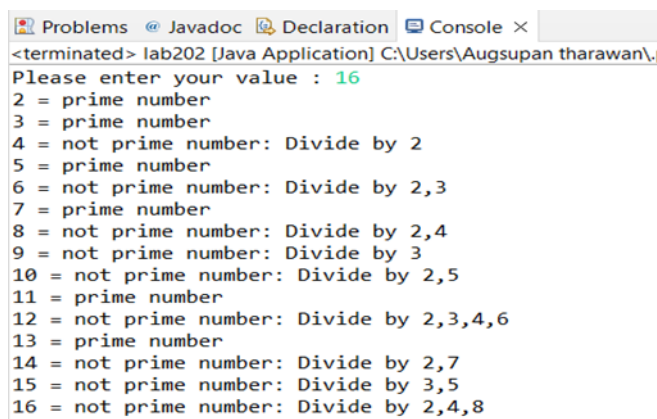
ในข้อ4.1 โจทย์ให้แสดงเลข 1 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้และแสดงคำตอบบรรทัดละ 5 ตัวเลขเท่านั้น สรุปผลที่ได้ก็ตามภาพนี้เลยครับ



```

Problems Javadoc Declaration Console ×
<terminated> lab2 (1) [Java Application] C:\Users\Augsupan tha
Please enter your value : 21
-----
1 ,2 ,3 ,4 ,5
6 ,7 ,8 ,9 ,10
11 ,12 ,13 ,14 ,15
16 ,17 ,18 ,19 ,20
21 ,
    
```

ข้อ4.2 โจทย์ให้เขียนโปรแกรมเพื่อแสดงจำนวนเฉพาะที่อยู่ระหว่าง เลข 2 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้ สรุปผลที่ได้ก็คือ



```

Problems Javadoc Declaration Console ×
<terminated> lab202 [Java Application] C:\Users\Augsupan tharawan\
Please enter your value : 16
2 = prime number
3 = prime number
4 = not prime number: Divide by 2
5 = prime number
6 = not prime number: Divide by 2,3
7 = prime number
8 = not prime number: Divide by 2,4
9 = not prime number: Divide by 3
10 = not prime number: Divide by 2,5
11 = prime number
12 = not prime number: Divide by 2,3,4,6
13 = prime number
14 = not prime number: Divide by 2,7
15 = not prime number: Divide by 3,5
16 = not prime number: Divide by 2,4,8
    
```

6. คำถามท้ายการทดลอง

6.1. คำสั่งต่างๆ ระหว่างภาษา C และภาษา Java มีความเหมือนหรือแตกต่างกันอย่างไร ?

ภาษา C	ภาษา Java
พัฒนาโปรแกรมแบบ Procedural	พัฒนาโปรแกรมแบบเชิงวัตถุ (OOP)
ได้รับการพัฒนาโดย Dennis M. Ritchie	ได้รับการพัฒนาโดย James Gosling
เป็นภาษา middle-level language	เป็นภาษา high-level language
ไม่รองรับ threading และ overloading function	รองรับ threading และ method overloading
รองรับ pointers	ไม่รองรับ pointers
ใช้ malloc บริหาร และจัดการหน่วยความจำ	ใช้คำสั่ง new บริหาร และจัดการหน่วยความจำ
Garbage collector ต้องทำเอง	Garbage collector จัดการเองแบบอัตโนมัติ
รองรับการเข้าถึงข้อมูลผ่าน value และ reference	รองรับการเข้าถึงข้อมูลผ่าน value

6.2. จงระบุข้อควรระวังในการใช้คำสั่ง Switch Case

ควรวิเคราะห์โจทย์ให้ละเอียดเพราะคำสั่ง switch case เป็นคำสั่งควบคุมเพื่อให้โปรแกรมทำงานแบบหลายทางเลือกคล้ายกับคำสั่ง if else-if แต่การใช้งานของมันจะเรียบง่ายและจำกัดกว่ามาก

6.3. การเลือกใช้คำสั่ง For เหมาะกับสถานการณ์ใดมากที่สุด

คำสั่ง For เหมาะกับสถานการณ์ที่จำนวนซ้ำที่ใช้สำหรับควบคุมให้โปรแกรมทำงานซ้ำๆ ภายใต้งื่อนไขที่กำหนด

6.4. คำสั่ง Casting ควรถูกนำมาใช้งานในสถานการณ์ใดที่สุด

ใช้ในสถานการณ์ที่เราต้องการแปลงประเภทข้อมูลหนึ่งไปเป็นอีกประเภทหนึ่งเรียกว่าการแคสต์ประเภทหรือการแปลงประเภทนั่นเองครับ