

ใบงานการทดลองที่ 2

เรื่อง ภาษาจาวากับการโปรแกรมเชิงวัตถุ

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการโปรแกรมเชิงวัตถุด้วยคำสั่งพื้นฐานในภาษาจาวา
- 1.2. รู้และเข้าใจเพื่อศึกษาและทดลองสร้างโปรแกรมเชิงวัตถุโดยใช้ภาษาการโปรแกรมเชิงวัตถุใหม่ๆ

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

- 3.1. คำสั่งที่ใช้ในการแสดงผลบนแถบ Console คือคำสั่งใด

`System.out.println("Hello World");`

- 3.2. คำสั่งการคอมเมนต์แบบบรรทัดเดียว และแบบหลายบรรทัด คืออะไร ?

คำสั่งที่ใช้คอมเมนต์แบบบรรทัดเดียวคือ `//คอมเมนต์แบบบรรทัดเดียว`

คำสั่งที่ใช้คอมเมนต์แบบหลายบรรทัดคือ `/*คอมเมนต์แบบ`

`หลายบรรทัด*/`

- 3.3. ประเภทข้อมูลแบบ Integer คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

คือประเภทข้อมูลที่เป็นชนิดข้อมูลจำนวนเต็มมีขนาดข้อมูล 32 bit (4 byte) Ex. `int age = 18;`

`System.out.println("Your age is" + age);`

- 3.4. ประเภทข้อมูลแบบ Double คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

คือประเภทข้อมูลที่เป็นชนิดข้อมูลตัวเลขในรูปแบบของทศนิยม Ex. `double money = 118.50;`

มีขนาดข้อมูล 64 bit (8 byte)

`System.out.println("Your money is" + money);`

- 3.5. ประเภทข้อมูลแบบ Float คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

คือประเภทข้อมูลที่เป็นชนิดข้อมูลตัวเลขในรูปแบบของทศนิยม Ex. `double money = 118.50;`

มีขนาดข้อมูล 32 bit (4 byte)

`System.out.println("Your money is" + money);`

- 3.6. ประเภทข้อมูลแบบ Bool คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

คือประเภทข้อมูลที่เป็นชนิดข้อมูลแบบเปรียบเทียบข้อมูล true และ false Ex. `char male = 'm';`

มีขนาดข้อมูล 1 bit

`System.out.println("The male initials are" + male);`

- 3.7. ประเภทข้อมูลแบบ Char คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

คือประเภทข้อมูลที่เป็นชนิดข้อมูลที่เก็บข้อมูลเป็นชนิดอักขระได้แก่ ตัวอักษร, ตัวเลข Ex. `String message = "Welcome to Java";`

และสัญลักษณ์ต่างๆ เก็บข้อมูลได้มากกว่า 1 ตัวอักษร

`System.out.println("Hi," + message);`

- 3.8. ประเภทข้อมูลแบบ String คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

คือประเภทข้อมูลที่เป็นชนิดข้อมูลที่เก็บข้อมูลเป็นชนิดอักขระได้แก่ ตัวอักษร, ตัวเลข Ex. `String messages = "Welcome to Java";`

และสัญลักษณ์ต่างๆ เก็บข้อมูลได้มากกว่า 1 ตัวอักษร

`System.out.println("Hi," + message);`

- 3.9. Widening Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เป็นการกำหนดค่าของ Variable ที่มี Data Type ลำดับต่ำกว่า Ex. `byte < short < int < long < float < double`
ให้กับ Variable ที่มี Data Type สูงกว่า

- 3.10. Narrowing Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เป็นการกำหนดค่าของ Variable ที่มี Data Type ลำดับสูงกว่า Ex. double < float < long < int < short < byte
ให้กับ Variable ที่มี Data Type สูงกว่า

3.11. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการทางคณิตศาสตร์ดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
+	เพิ่มค่า	x = 5 + 5
-	ลบค่า	x = 10 - 5
*	คูณค่า	y = 2 * 2
/	หารค่า	y = 4 / 2
++	การเพิ่มค่าของจำนวนเต็ม 1	x = X++
--	การลดค่าของจำนวนเต็ม 1	y = y--
%	การหารแล้วเอาแต่เศษ	z = 11 % 2

3.12. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อกำหนดค่าดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
=	เป็นการกำหนดค่าให้กับตัวแปร	a = 10
+=	นำตัวแปรด้านซ้ายมาบวกเพิ่ม ก่อนที่จะกำหนดค่าให้	a += 10 มีค่าเท่ากับ a = a + 10
-=	นำตัวแปรด้านซ้ายมาลบออก ก่อนที่จะกำหนดค่าให้	a -= 10 มีค่าเท่ากับ a = a - 10
*=	นำตัวแปรด้านซ้ายมาคูณ ก่อนที่จะกำหนดค่าให้	a *= 10 มีค่าเท่ากับ a = a * 10
/=	นำตัวแปรด้านซ้ายมาหาร ก่อนที่จะกำหนดค่าให้	a /= 10 มีค่าเท่ากับ a = a / 10
%=	นำตัวแปรด้านซ้ายมาหารเก็บเศษ ก่อนที่จะกำหนดค่าให้	a %= 10 มีค่าเท่ากับ a = a % 10
&=	นำตัวแปรด้านซ้ายมาทำการ AND ระดับบิต ก่อนที่จะกำหนดค่าให้	a &= 10 มีค่าเท่ากับ a = a & 10
=	นำตัวแปรด้านซ้ายมา OR ระดับบิต ก่อนที่จะกำหนดค่าให้	a = 10 มีค่าเท่ากับ a = a 10

3.13. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อเปรียบเทียบดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
==	การตรวจสอบค่าทั้งสองค่าว่ามีค่าเท่ากันไหมถ้าเท่ากันจะได้ค่า True	(2 == 2)
>	การตรวจสอบค่าทั้งสองค่าทางด้านซ้ายมีค่ามากกว่าค่าทางด้านขวาไหมถ้าใช่จะได้ค่า True	(2 != 3)
>=	การตรวจสอบค่าทั้งสองค่าทางด้านซ้ายมีค่ามากกว่าหรือเท่ากับค่าทางด้านขวาไหมถ้าใช่จะได้ค่า True	(5 > 4)
<	การตรวจสอบค่าทั้งสองค่าทางด้านซ้ายมีค่าน้อยกว่าค่าทางด้านขวาไหมถ้าใช่จะได้ค่า True	(1 < 5)
<=	การตรวจสอบค่าทั้งสองค่าทางด้านซ้ายมีค่าน้อยกว่าหรือเท่ากับค่าทางด้านขวาไหมถ้าใช่จะได้ค่า True	(5 >= 5)
!=	การตรวจสอบค่าทั้งสองค่ามีค่าไม่เท่ากันไหมถ้าไม่เท่ากันจะได้ค่า True	(5 <= 5)

3.14. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการตรรกศาสตร์ดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
&&	จะเป็นค่า true ก็ต่อเมื่อค่าทั้งสองค่าเป็น true	(a & b) ได้ค่า false
	จะเป็นค่า false ก็ต่อเมื่อค่าทั้งสองค่าเป็น false	(a b) ได้ค่า true
!	จะทำการกลับค่าจาก true เป็น false หรือจาก false เป็น true	!a ได้ค่า false

3.15. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if พร้อมยกตัวอย่างประกอบ

.....

```

if(20 > 18){
    System.out.println("20 is greater than 18");
}

```

3.16. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if else พร้อมยกตัวอย่างประกอบ

```

if(13%2 == 0){
    System.out.println("even number");
} else {
    System.out.println("odd number");
}

```

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

```

if(เงื่อนไขที่ 1) {
    //โค้ดที่จะดำเนินการหากเงื่อนไขเป็นจริง
} else if(เงื่อนไขที่ 2) {
    //โค้ดที่จะดำเนินการหากเงื่อนไขที่ 1 เป็นเท็จและเงื่อนไขที่ 2 เป็นจริง
} else {
    //โค้ดที่จะดำเนินการหากเงื่อนไขที่ 1 เป็นเท็จและเงื่อนไขที่ 2 เป็นเท็จ
}

```

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

```

int time = 20;
String result = (time < 18) ? "Good evening.";
System.out.println(result);

```

3.19. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง switch พร้อมยกตัวอย่างประกอบ

```
Switch () {
```

```
}
```

3.20. การวนรอบ - จงเขียน Syntax การใช้งานคำสั่ง for พร้อมยกตัวอย่างประกอบ

```
for(กำหนดค่าเงื่อนไขจะปรับค่า){
```

```
//โค้ดที่ต้องการดำเนินการ
```

```
}
```

3.21. การวนรอบ - จงเขียน Syntax การใช้งานคำสั่ง while พร้อมยกตัวอย่างประกอบ

```
while(เงื่อนไข){
```

```
//โค้ดที่ต้องการดำเนินการ
```

```
//ปรับค่าตัวแปร
```

```
}
```

3.22. การวนรอบ - จงเขียน Syntax การใช้งานคำสั่ง do while พร้อมยกตัวอย่างประกอบ

```
do{
```

```
//โค้ดที่จะดำเนินการ
```

```
/ปรับค่าตัวแปร
```

```
}while (เงื่อนไข);
```

3.23. อธิบายการทำงานของคำสั่ง break พร้อมยกตัวอย่างประกอบ

```
for(int i = 0; i < 10; i++){           ใช้เพื่อ "ออกจากคำสั่ง" ของคำสั่ง switch
```

```
if(i == 4) {
```

```
break;
```

```
}
```

```
System.out.println(i);
```

```
}
```

3.24. อธิบายการทำงานของคำสั่ง continue พร้อมยกตัวอย่างประกอบ

```
for(int i = 0; i < 10; i++){           ใช้เพื่อดำเนินการคำสั่งต่อไปในขณะลูปอยู่
```

```
if(i == 4) {
```

```
continue;
```

```
}
```

```
System.out.println(i);
```

```
}
```

3.25. อธิบายการทำงานของคำสั่ง Enum

```
enum เป็น "คลาส" พิเศษที่แสดงถึงกลุ่มของค่าคงที่หากต้องการสร้าง enum ให้ใช้คีย์เวิร์ด enum
```

```
(แทนคลาสหรืออินเทอร์เฟซ) และคั่นค่าคงที่ด้วยเครื่องหมายจุลภาคและควรเป็นตัวพิมพ์ใหญ่
```

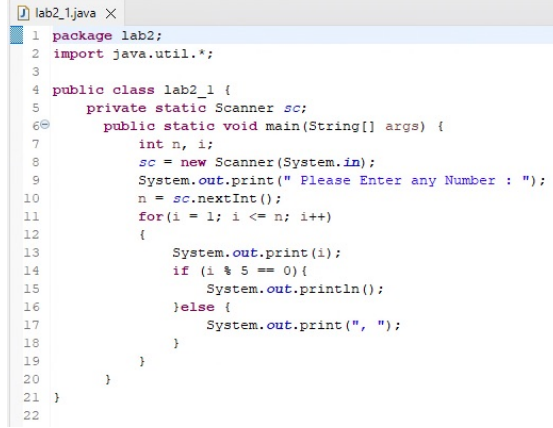
4. ลำดับขั้นการปฏิบัติการ

4.1. จงแก้โจทย์ปัญหาดังต่อไปนี้

4.1.1. จงเขียนโปรแกรมเพื่อแสดงเลข 1 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้ และแสดงคำตอบบรรทัดละ 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. ผลงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผลงาน)

ผลงาน	โค้ดโปรแกรม
	 <pre> 1 package lab2; 2 import java.util.*; 3 4 public class lab2_1 { 5 private static Scanner sc; 6 public static void main(String[] args) { 7 int n, i; 8 sc = new Scanner(System.in); 9 System.out.print(" Please Enter any Number : "); 10 n = sc.nextInt(); 11 for(i = 1; i <= n; i++) 12 { 13 System.out.print(i); 14 if (i % 5 == 0){ 15 System.out.println(); 16 }else { 17 System.out.print(", "); 18 } 19 } 20 } 21 } 22 </pre>

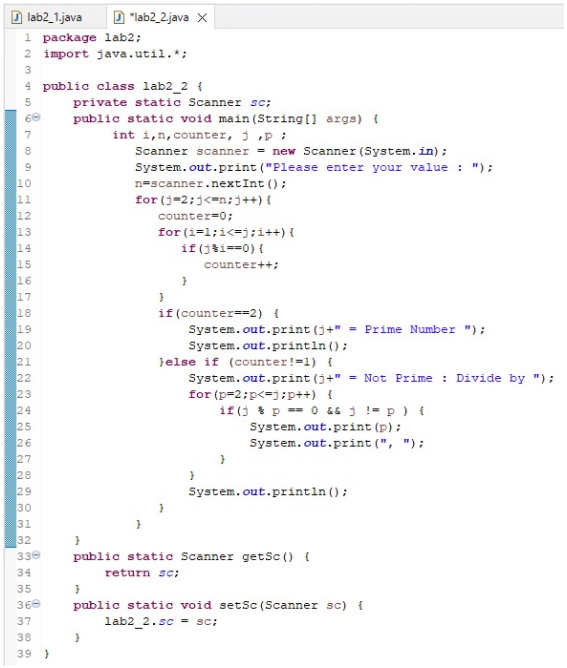
4.2. จงแก้ไขข้อปัญหาดังต่อไปนี้

4.2.1. จงเขียนโปรแกรมเพื่อแสดงจำนวนเฉพาะที่อยู่ระหว่าง เลข 2 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้

Test case 1	Test case 2
Please enter your value : 8 ----- 2 = Prime Number 3 = Prime Number 4 = Not Prime : Divide by 2 5 = Prime Number 6 = Not Prime : Divide by 2, 3 7 = Prime Number	Please enter your value : 16 ----- 2 = Prime Number 3 = Prime Number 4 = Not Prime : Divide by 2 5 = Prime Number 6 = Not Prime : Divide by 2, 3 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
--------------------------------	--

4.2.2. ผลงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผลงาน)

ผลงาน	โค้ดโปรแกรม
	 <pre> 1 package lab2; 2 import java.util.*; 3 4 public class lab2_2 { 5 private static Scanner sc; 6 public static void main(String[] args) { 7 int i,n,counter, j ,p ; 8 Scanner scanner = new Scanner(System.in); 9 System.out.print("Please enter your value : "); 10 n=scanner.nextInt(); 11 for(j=2;j<=n;j++){ 12 counter=0; 13 for(i=1;i<=j;i++){ 14 if(j%i==0){ 15 counter++; 16 } 17 } 18 if(counter==2) { 19 System.out.print(j+" = Prime Number "); 20 System.out.println(); 21 }else if (counter!=1) { 22 System.out.print(j+" = Not Prime : Divide by "); 23 for(p=2;p<=j;p++) { 24 if(j % p == 0 && j != p) { 25 System.out.print(p); 26 System.out.print(", "); 27 } 28 } 29 System.out.println(); 30 } 31 } 32 } 33 public static Scanner getSc() { 34 return sc; 35 } 36 public static void setSc(Scanner sc) { 37 lab2_2.sc = sc; 38 } 39 } </pre>

--	--

5. สรุปผลการปฏิบัติการ

ได้ทำความเข้าใจคำสั่งพื้นฐานในภาษา Java โดยเรียนรู้ในเรื่องประเภทข้อมูล, ตัวดำเนินการทางคณิตศาสตร์, ตัวดำเนินการเพื่อกำหนดค่าตัวดำเนินการเพื่อเปรียบเทียบค่า, ตัวดำเนินการทางตรรกศาสตร์ และการเขียน syntax ประโยคเงื่อนไข การเขียนโปรแกรมนับเลขจาก 1 to N โปรแกรมหาจำนวนเฉพาะและหาตัวประกอบของตัวที่ไม่เป็นจำนวนเฉพาะตั้งแต่ 2 to N

6. คำถามท้ายการทดลอง

6.1. คำสั่งต่างๆ ระหว่างภาษา C และภาษา Java มีความเหมือนหรือแตกต่างกันอย่างไร ?

ภาษา C = พัฒนาโปรแกรมแบบ Procedural ได้รับการพัฒนาโดย Dennis M. Ritchie, เป็นภาษา middle-level language, ไม่รองรับ threading และ overloading function, ไม่รองรับ pointers

ภาษา Java = พัฒนาโปรแกรมเชิงวัตถุ(OOP), ได้รับการพัฒนาโดย James Gosling, เป็นภาษา high-level language, รองรับ threading และ method overloading, ไม่รองรับ pointers

6.2. จงระบุข้อควรระวังในการใช้คำสั่ง Switch Case

ต้องกำหนด break ไว้ภายหลังโค้ดเพื่อหยุดการทำงานของ Switch Case

6.3. การเลือกใช้คำสั่ง For เหมาะกับสถานการณ์ใดมากที่สุด

ใช้กับการวนซ้ำในจำนวนรอบที่แน่นอน เนื่องจากสามารถกำหนดค่าเริ่มต้น เงื่อนไข และการเปลี่ยนแปลงค่า

6.4. คำสั่ง Casting ควรนำมาใช้งานในสถานการณ์ใดที่สุด

ใช้ต่อเมื่อเรากำหนดค่าของ Variable ที่มี Data Type ที่เราไม่ต้องการให้กับ Variable ที่มี Data Type ที่เราต้องการ เช่น เป็น Data Type ที่เก็บตัวเลขเหมือนกับลำดับของ Data Type ใน Type Casting เรียงจากต่ำไปสูง byte < short < int < long < float < double