

ใบงาน

รหัสวิชา 31901-2006

วิชาการพัฒนาซอฟต์แวร์เชิงวัตถุ

(Object-Oriented Software Development)



ครูผู้สอน

นางสุปรียา รัตนวิทยาพันธ์



แผนกวิชาเทคโนโลยีสารสนเทศ



วิทยาลัยเทคนิคชลบุรี

ใช้เพื่อการศึกษา ห้ามจำหน่าย



หลักสูตร : หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง พุทธศักราช 2567
ระดับ : ปวส.
สาขาวิชา : สาขาวิชาเทคโนโลยีสารสนเทศ
รายวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
รหัสวิชา : 31901-2006

คำอธิบายฯ

หน้าที่

แผ่นที่ :

31901-2006 การพัฒนาซอฟต์แวร์เชิงวัตถุ

1-4-3

Object-Oriented Software Development

อ้างอิงมาตรฐาน

—

ผลลัพธ์การเรียนรู้ระดับรายวิชา

พัฒนาซอฟต์แวร์ตามหลักการเชิงวัตถุด้วยความรับผิดชอบ ความละเอียด รอบคอบ และถูกต้อง

จุดประสงค์รายวิชา เพื่อให้

1. เข้าใจเกี่ยวกับหลักการพัฒนาซอฟต์แวร์เชิงวัตถุ
2. มีทักษะในการพัฒนาซอฟต์แวร์เชิงวัตถุ
3. มีเจตคติและกิจนิสัยที่ดีในการปฏิบัติงานด้วยความรับผิดชอบ ความละเอียด รอบคอบ และถูกต้อง
4. มีความสามารถประยุกต์ใช้ภาษาคอมพิวเตอร์เชิงวัตถุเพื่อพัฒนาซอฟต์แวร์

สมรรถนะรายวิชา

1. ประมวลความรู้เกี่ยวกับการพัฒนาซอฟต์แวร์ตามหลักการเชิงวัตถุ
2. พัฒนาซอฟต์แวร์ด้วยภาษาคอมพิวเตอร์เชิงวัตถุ
3. ประยุกต์ใช้ภาษาคอมพิวเตอร์เชิงวัตถุเพื่อพัฒนาซอฟต์แวร์

คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับหลักการพื้นฐานของการโปรแกรมเชิงวัตถุด้วยภาษาคอมพิวเตอร์เชิงวัตถุ การติดตั้งและกำหนดการทำงานของโปรแกรมที่เกี่ยวข้องกับการพัฒนาโปรแกรมเชิงวัตถุ ลำดับขั้นตอนการ เขียนโปรแกรมเชิงวัตถุ การสร้างและเรียกใช้งาน Classes, Objects, Attributes, Methods, Constructors, Modifiers, Encapsulation, Packages, Inheritance, Polymorphism, Abstraction, Interface, Exceptions Handlings, Threads และเชื่อมต่อกับฐานข้อมูลในการพัฒนาซอฟต์แวร์เชิงวัตถุในงานอาชีพ

หน่วยการเรียนรู้

หน่วยที่	ชื่อหน่วยการเรียนรู้	จำนวน ชั่วโมง	สัปดาห์ที่
1	หลักการพื้นฐานของการโปรแกรมเชิงวัตถุ	5	1
2	การติดตั้งและกำหนดการทำงานของโปรแกรม	5	2
3	Classes และ Objects	10	3-4
4	Constructors และ Modifiers	5	5
5	Encapsulation และ Packages	10	6-7
6	Inheritance และ Polymorphism	10	8-9
7	Abstraction และ Interface	10	10-11
8	Exception Handling และ Threads	10	12-13
9	การเชื่อมต่อฐานข้อมูลและการประยุกต์ใช้ในงานอาชีพ	5	14
10	สอบปลายภาค	5	15



หลักสูตร : ประกาศนียบัตรวิชาชีพชั้นสูง พุทธศักราช 2567

ระดับ : ประกาศนียบัตรวิชาชีพชั้นสูง

สาขาวิชา : เทคโนโลยีสารสนเทศ

รายวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ

รหัสวิชา : 31901-2006

ใบวิเคราะห์ผัง

สมรรถนะ

แผนที่ :

หน้าที่

การพัฒนาซอฟต์แวร์เชิงวัตถุ

พัฒนาซอฟต์แวร์ด้วยภาษาคอมพิวเตอร์

- การติดตั้งโปรแกรม Edit Plus
- การติดตั้งโปรแกรม jdk
- การสร้าง คลาส และ ออบเจกต์
- งานการซ่อนข้อมูล(Encapsulation)
- การสืบทอดคุณสมบัติ(Inheritance)
- การมีหลายรูปแบบ(Polymorphism)
- งานการซ่อนรายละเอียด(Abstraction)
- งานการจัดการข้อผิดพลาด Exception Handling
- งานการทำงานหลายอย่างพร้อมกัน(Threads)
- งานการเขียนโปรแกรมเชื่อมต่อด้านข้อมูล



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การติดตั้งโปรแกรม jdk

ใบสั่งงานที่

1

หน้าที่

แผ่นที่ : 1

คำสั่ง จงทำติดตั้งและกำหนดค่าเริ่มต้นโปรแกรม JDK

ORACLE

Products Industries Resources Customers Partners Developers Company

Q

🇹🇭

🔒 View Accounts

📄 Contact Sales

Java downloads Tools and resources Java archive

Java 22, Java 21, and Java 17 available now

JDK 21 is the latest long-term support release of Java SE Platform.

Learn about Java SE Subscription

JDK 22 JDK 21 JDK 17 GraalVM for JDK 22 GraalVM for JDK 21 GraalVM for JDK 17

JDK Development Kit 22.0.2 downloads

JDK 22 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

JDK 22 will receive updates under these terms, until September 2024, when it will be superseded by JDK 23.

Linux macOS Windows

Product/file description	File size	Download
x64 Compressed Archive	184.16 MB	https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.zip (sha256)
x64 Installer	164.35 MB	https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.exe (sha256)

รูปภาพ แสดงเว็บไซต์ของ Oracle เพื่อดาวน์โหลดไฟล์

ลำดับขั้นตอนการปฏิบัติงาน

1. เตรียมเครื่องคอมพิวเตอร์
2. ขั้นที่ 1: ดาวน์โหลดโปรแกรม

เข้าสู่เว็บไซต์ของ Oracle หรือ OpenJDK (ถ้าคุณไม่ต้องการใช้ JDK จาก Oracle) ที่

<https://www.oracle.com/java/technologies/javase-downloads.html> (สำหรับ Oracle) หรือ

<https://jdk.java.net/> (สำหรับ OpenJDK).

- 3.ดับเบิลคลิกที่ไฟล์ดาวน์โหลดเพื่อเริ่มกระบวนการติดตั้ง.
- 4.ยอมรับข้อตกลงการใช้งาน (License Agreement) และเลือกโฟลเดอร์ปลายทางที่คุณต้องการติดตั้ง JDK.
- 5.เลือกส่วนที่คุณต้องการติดตั้ง (JDK และ JRE มักมีให้เลือก).
- 6.ติดตั้ง JDK โดยคลิกที่ปุ่ม "Install" หรือ "Next" ตามคำแนะนำบนหน้าต่างติดตั้ง.
- 7.รอให้กระบวนการติดตั้งเสร็จสิ้น.



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การติดตั้งโปรแกรม jdk

ใบสั่งงานที่

1

หน้าที่

แผ่นที่ : 2

เครื่องมือและอุปกรณ์

1. เครื่องคอมพิวเตอร์
2. โปรแกรม JDK 21

เวลาในการปฏิบัติงาน

30 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การติดตั้งโปรแกรม Edit Plus

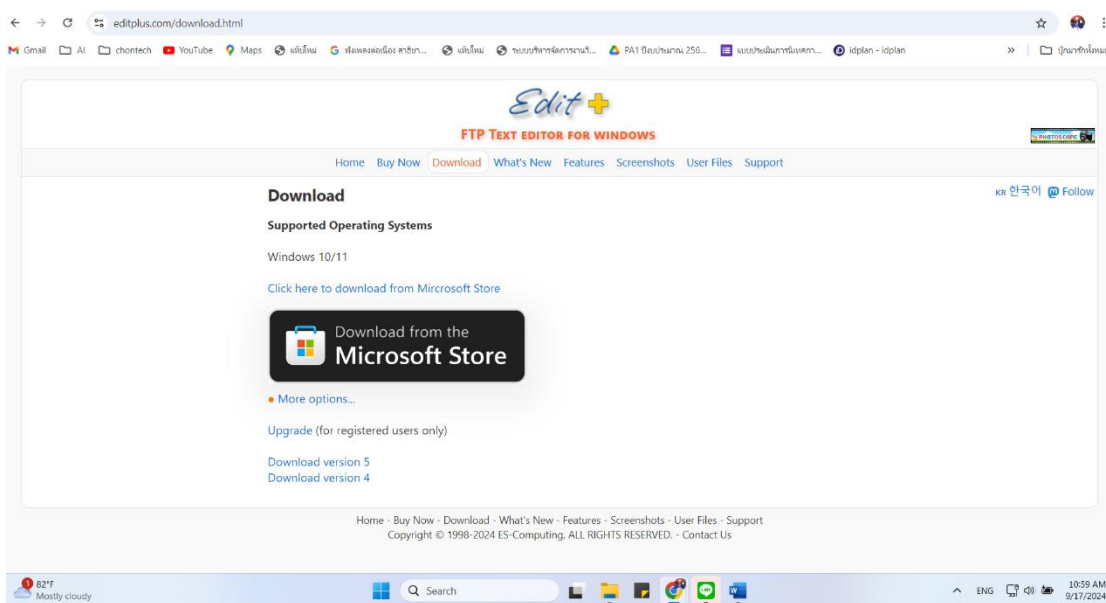
ใบสั่งงานที่

1_1

หน้าที่

แผ่นที่ : 1

คำสั่ง จงทำการ Download และติดตั้งโปรแกรม Edit Plus



รูปภาพแสดง หน้า เว็บไซต์การ Download โปรแกรม

ลำดับขั้นตอนการปฏิบัติงาน

1. เตรียมเครื่องคอมพิวเตอร์
2. ขั้นที่ 1: ดาวน์โหลดโปรแกรม Edit Plus
 - 2.1 เข้าสู่เว็บไซต์ของ Edit Plus ที่ <https://www.editplus.com/>
 - 2.2 บนเมนูหลักของเว็บไซต์ คลิกที่ "Download."
 - 2.3 เลือกรุ่นของ Edit Plus ที่คุณต้องการดาวน์โหลด (เวอร์ชันทดลองใช้งานฟรีหรือเวอร์ชันจ่ายเงิน ตามที่คุณต้องการ).
 - 2.4 คลิกที่ลิงก์ดาวน์โหลดเพื่อดาวน์โหลดไฟล์ติดตั้ง ไฟล์ .exe
3. ขั้นที่ 2. ติดตั้ง Edit Plus



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การติดตั้งโปรแกรม Edit Plus

ใบสั่งงานที่

1_1

แผ่นที่ : 2

หน้า
ที่

- 3.1 เมื่อดาวน์โหลดเสร็จสิ้น ดับเบิลคลิกที่ไฟล์ติดตั้ง (.exe) เพื่อเริ่มกระบวนการติดตั้ง.
- 3.2 ที่หน้าต่างติดตั้ง คลิกที่ "Next" เพื่อเริ่มต้น.
- 3.3 ยอมรับข้อกำหนดและเงื่อนไขการใช้งานโปรแกรม Edit Plus โดยกดปุ่ม "I Agree."
- 3.4 เลือกโฟลเดอร์ที่คุณต้องการติดตั้ง Edit Plus หากคุณไม่ต้องการใช้โฟลเดอร์เริ่มต้น.
- 3.5 เลือกส่วนเสริม (Extensions) ที่คุณต้องการติดตั้ง (สามารถปล่อยว่างไว้ก็ได้) แล้วคลิก "Next."
- 3.6 ปรับแต่งตัวเลือกการติดตั้งต่าง ๆ ตามความต้องการของคุณ และคลิก "Next."
- 3.7 ตรวจสอบการตั้งค่าของคุณอีกครั้งในหน้าสรุป และคลิก "Install" เพื่อเริ่มกระบวนการติดตั้ง.
- 3.8 รอให้กระบวนการติดตั้งเสร็จสิ้น หลังจากนั้นคลิก "Finish" เพื่อปิดหน้าต่างติดตั้ง.
4. ขั้นตอนที่ 3 ใช้งาน Edit Plus
 - 4.1 เมื่อ Edit Plus ติดตั้งเสร็จสิ้น เริ่มใช้โปรแกรมได้โดยคลิกที่ไอคอน Edit Plus บนเดสก์ท็อปหรือในเมนูเริ่มของ Windows.
5. ตรวจสอบความสมบูรณ์ของ โปรแกรม

เครื่องมือและอุปกรณ์

1. เครื่องคอมพิวเตอร์
2. โปรแกรม Edit Plus

เวลาในการปฏิบัติงาน

30 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การสร้าง คลาส และ ออบเจ็กต์

ใบสั่งงานที่

2

หน้า
ที่

แผ่นที่ : 1

วัตถุประสงค์:

1. เข้าใจพื้นฐานการสร้างคลาส (Class) ในภาษาจาวา
2. สามารถสร้างและใช้ออบเจ็กต์ (Object) จากคลาสได้
3. เรียนรู้การใช้งานคอนสตรัคเตอร์ (Constructor) และเมธอด (Method) ในคลาส

ตอนที่ 1 จงตอบคำถามต่อไปนี้

1. คลาส (Class) คืออะไร

.....

.....

.....

.....

2. ออบเจ็กต์ (Object) คืออะไร

.....

.....

.....

.....

3. ให้เขียนโครงสร้างของคลาส

.....

.....

.....

.....



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การสร้าง คลาส และ ออบเจกต์

ใบสั่งงานที่

2

หน้า
ที่

แผ่นที่ : 2

4. รูปแบบการสร้าง ออบเจกต์จากคลาส

.....

.....

.....

.....

ตอนที่ 2 ให้เขียนโปรแกรมภาษาจาวาดังต่อไปนี้

2.1 เขียนโปรแกรมสร้างคลาส ชื่อ Student

2.1.1 มีคุณสมบัติดังนี้

- 1.รหัส String
2. ชื่อ-สกุล String
3. อายุ int
4. เพศ String

2.1.2 สร้างคอนสตรัคเตอร์ที่รับพารามิเตอร์ ทั้ง 4 ตัว

2.1.3 สร้าง เมธอด displayInfo() เพื่อแสดงข้อมูลของ นักเรียน

2.2 สร้างและใช้ออบเจกต์

2.2.1 สร้างคลาส Main

2.2.2 สร้างออบเจกต์ของคลาส Student และแสดงข้อมูลของนักเรียนแต่ละคน

3.1 สร้างคลาสใหม่ชื่อ Book ที่มีคุณสมบัติ:

- title (ชื่อหนังสือ)
- author (ชื่อผู้เขียน)
- price (ราคา)

3.2 สร้างคอนสตรัคเตอร์และเมธอด displayInfo() เพื่อแสดงข้อมูลของหนังสือ

3.3 สร้างออบเจกต์จากคลาส Book และแสดงข้อมูล



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การสร้าง คลาส และ ออบเจ็กต์

ใบสั่งงานที่

2

หน้า
ที่

แผ่นที่ : 3

เครื่องมือและอุปกรณ์

1. เครื่องคอมพิวเตอร์
2. โปรแกรม Edit Plus
- 3 โปรแกรม JDK

เวลาที่ใช้ในการปฏิบัติงาน

โปรแกรมละ 30 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Encapsulation

ใบสั่งงานที่

3

หน้าที่

แผ่นที่ : 1

วัตถุประสงค์:

1. เข้าใจหลักการของการซ่อนข้อมูล (Encapsulation) ในการเขียนโปรแกรมเชิงวัตถุ (OOP)
2. สามารถใช้ตัวเข้าถึงข้อมูล (Access Modifiers) เพื่อควบคุมการเข้าถึงคุณสมบัติของคลาส
3. เรียนรู้การใช้งานเมธอด getter และ setter เพื่อเข้าถึงและแก้ไขข้อมูลภายในคลาส

ทฤษฎีเบื้องต้น

1.1 Encapsulation คืออะไร?

Encapsulation คือหลักการที่ใช้ในการซ่อนข้อมูล (Data Hiding) โดยการทำให้คุณสมบัติ (Attributes) ของออบเจกต์เป็นแบบ private และควบคุมการเข้าถึงข้อมูลเหล่านั้นผ่านเมธอด getter และ setter

1.2 ตัวเข้าถึงข้อมูล (Access Modifiers)

- **private:** จำกัดการเข้าถึงเฉพาะภายในคลาสเท่านั้น
- **public:** สามารถเข้าถึงได้จากภายนอกคลาส
- **protected** และ **default:** สำหรับการควบคุมการเข้าถึงที่ซับซ้อนขึ้น

1.3 การใช้งาน getter และ setter

- **Getter:** เป็นเมธอดที่ใช้ในการเข้าถึงค่าของคุณสมบัติ
- **Setter:** เป็นเมธอดที่ใช้ในการเปลี่ยนแปลงค่าของคุณสมบัติ

โครงสร้างของ getter และ setter มีดังนี้:

```
java

// Getter
public ชนิดข้อมูล getชื่อคุณสมบัตินี้() {
    return ชื่อคุณสมบัตินี้;
}

// Setter
public void setชื่อคุณสมบัตินี้(ชนิดข้อมูล ชื่อคุณสมบัตินี้) {
    this.ชื่อคุณสมบัตินี้ = ชื่อคุณสมบัตินี้;
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การสร้าง คลาส และ ออบเจ็ค

ใบสั่งงานที่
3

แผ่นที่ : 2

หน้า
ที่

ตัวอย่างการใช้งาน Encapsulation

```
class Person {  
    // กำหนดคุณสมบัติเป็น private  
    private String name;  
    private int age;  
    // สร้างคอนสตรักเตอร์  
    public Person(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
    // Getter สำหรับ name  
    public String getName() {  
        return name;  
    }  
    // Setter สำหรับ name  
    public void setName(String name) {  
        this.name = name;  
    }  
    // Getter สำหรับ age  
    public int getAge() {  
        return age;  
    }  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การสร้าง คลาส และ ออบเจ็ค

ใบสั่งงานที่
3
แผ่นที่ : 3

หน้า
ที่

// Setter สำหรับ age

```
public void setAge(int age) {  
    if (age > 0) {  
        this.age = age;  
    } else {  
        System.out.println("Age must be positive.");  
    }  
}
```

การใช้งานคลาส Person ที่ใช้ Encapsulation

```
public class Main {  
    public static void main(String[] args) {  
        // สร้างออบเจ็กต์ Person  
        Person person1 = new Person("Anan", 25);  
        // ใช้เมธอด getter และ setter  
        System.out.println("Name: " + person1.getName());  
        System.out.println("Age: " + person1.getAge());  
        // เปลี่ยนชื่อและอายุโดยใช้ setter  
        person1.setName("Somsak");  
        person1.setAge(30);  
        // แสดงข้อมูลที่แก้ไขแล้ว  
        System.out.println("Updated Name: " + person1.getName());  
        System.out.println("Updated Age: " + person1.getAge());  
    }  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การสร้าง คลาส และ ออบเจกต์

ใบสั่งงานที่

3

หน้าที่

แผ่นที่ : 4

คำสั่ง เขียนโปรแกรมภาษา จาวา ดังต่อไปนี้

1.1: การสร้างคลาส "Product"

- 1.สร้างคลาสชื่อ Product ที่มีคุณสมบัติเป็นแบบ private:
 - o name (ชื่อสินค้า) ชนิดข้อมูลเป็น String
 - o price (ราคา) ชนิดข้อมูลเป็น double
- 2.สร้างเมธอด getter และ setter สำหรับทุกคุณสมบัติ
- 3.สร้างคอนสตรัคเตอร์สำหรับตั้งค่าคุณสมบัติเริ่มต้น

1.2: การสร้างและใช้ออบเจกต์จากคลาส "Product"

- 1.สร้างออบเจกต์จากคลาส Product และแสดงค่าของคุณสมบัติ name และ price
2. ใช้ setter เพื่อเปลี่ยนค่า name และ price แล้วแสดงผลใหม่

2.1สร้างคลาส Employee ที่มีคุณสมบัติเป็น private:

- o id (รหัสพนักงาน)
- o name (ชื่อพนักงาน)
- o salary (เงินเดือน)

2.2 สร้างเมธอด getter และ setter สำหรับทุกคุณสมบัติ

2.3 สร้างคอนสตรัคเตอร์สำหรับตั้งค่าข้อมูลพนักงาน

2.4 สร้างออบเจกต์จากคลาส Employee และทดลองแก้ไขข้อมูลผ่าน setter

เครื่องมือที่ใช้ในการปฏิบัติงาน

- 1.เครื่องคอมพิวเตอร์
- 2.โปรแกรม Edit Plus
- 3 โปรแกรม JDK

เวลาในการปฏิบัติงาน

60 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Inheritance

ใบสั่งงานที่

4

หน้า
ที่

แผ่นที่ : 1

วัตถุประสงค์:

1. เข้าใจหลักการสืบทอดคุณสมบัติ (Inheritance) ในการเขียนโปรแกรมเชิงวัตถุ (OOP)
2. สามารถสร้างคลาสซูเปอร์ (Superclass) และคลาสซับ (Subclass) ได้
3. เรียนรู้การใช้งานคีย์เวิร์ด extends เพื่อสืบทอดคุณสมบัติและพฤติกรรม
4. เข้าใจการใช้เมธอดโอเวอร์ไรด์ (Method Override) เพื่อแก้ไขพฤติกรรมของซูเปอร์คลาสในซับคลาส

ทฤษฎีเบื้องต้น

1.1 Inheritance คืออะไร?

- Inheritance คือกระบวนการที่คลาสหนึ่ง (เรียกว่า Subclass) สืบทอดคุณสมบัติและพฤติกรรมจากคลาสอื่น (เรียกว่า Superclass) ทำให้สามารถนำโค้ดที่ซ้ำซ้อนมาใช้งานใหม่ได้โดยไม่ต้องเขียนใหม่

โครงสร้างของการสืบทอด

ในภาษาจาวา เราสามารถใช้คีย์เวิร์ด extends เพื่อระบุว่า Subclass จะสืบทอดจาก Superclass

```
java

class Superclass {
    // คุณสมบัติและเมธอดของ Superclass
}

class Subclass extends Superclass {
    // คุณสมบัติและเมธอดของ Subclass
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : Inheritance

ใบสั่งงานที่

4

หน้าที่

แผ่นที่ : 2

1.2 Method Overriding

- การ **Override** เมธอดคือการเขียนเมธอดใหม่ใน Subclass ที่มีชื่อเดียวกันกับเมธอดใน Superclass เพื่อเปลี่ยนพฤติกรรมหรือปรับปรุงให้เหมาะสม

โครงสร้างของการ Override

```
java

@Override
public void ชื่อเมธอด() {
    // การทำงานที่ถูกแก้ไขใน Subclass
}
```

ตัวอย่างการใช้งาน Inheritance

// Superclass: Vehicle

```
class Vehicle {

    public String brand;

    public Vehicle(String brand) {

        this.brand = brand;

    }

    public void startEngine() {

        System.out.println("Engine is starting...");

    }

}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : Inheritance

ใบสั่งงานที่
4
แผ่นที่ : 3

หน้า
ที่

// Subclass: Car

```
class Car extends Vehicle {
```

```
    public int speed;
```

```
    public Car(String brand, int speed) {
```

```
        super(brand); // เรียกคอนสตรัคเตอร์ของ Superclass
```

```
        this.speed = speed;
```

```
    }
```

// Override เมธอด startEngine

@Override

```
public void startEngine() {
```

```
    System.out.println(brand + " car engine is starting at speed: " + speed);
```

```
}
```

```
}
```

การสร้างออบเจกต์และใช้งาน Subclass

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        // สร้างออบเจกต์ Car
```

```
        Car car1 = new Car("Toyota", 120);
```

```
        // เรียกใช้เมธอด startEngine ของ Car (Override)
```

```
        car1.startEngine();
```

```
    }
```

```
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : : Inheritance

ใบสั่งงานที่

4

หน้าที่

แผ่นที่ : 4

คำสั่ง เขียนโปรแกรมภาษา จาวา ดังต่อไปนี้

1.1: การสร้าง Superclass และ Subclass

- สร้าง Superclass ชื่อ Animal ที่มีคุณสมบัติ:
 - name (ชื่อสัตว์)
 - age (อายุสัตว์)
 - สร้างเมธอด makeSound() ที่พิมพ์ข้อความ "Animal is making a sound"
- สร้าง Subclass ชื่อ Dog ที่สืบทอดจาก Animal:
 - เพิ่มคุณสมบัติ breed (สายพันธุ์สุนัข)
 - Override เมธอด makeSound() ให้พิมพ์ข้อความ "Dog is barking"

1.2: การสร้างและใช้ออบเจกต์

- สร้างออบเจกต์จากคลาส Dog
- ใช้เมธอด makeSound() ของคลาส Dog และแสดงผล

2.1 สร้าง Superclass ชื่อ Person ที่มีคุณสมบัติ:

- name (ชื่อ)
- age (อายุ)
- สร้างเมธอด introduce() เพื่อแนะนำตัว

2.2 สร้าง Subclass ชื่อ Student ที่สืบทอดจาก Person:

- เพิ่มคุณสมบัติ studentId (รหัสนักเรียน)
- Override เมธอด introduce() เพื่อแนะนำตัวนักเรียนพร้อมระบุรหัสนักเรียน

2.3 สร้าง Subclass ชื่อ Teacher ที่สืบทอดจาก Person:

- เพิ่มคุณสมบัติ subject (วิชาที่สอน)
- Override เมธอด introduce() เพื่อแนะนำตัวครูพร้อมระบุวิชาที่สอน

2.4 สร้างออบเจกต์จากคลาส Student และ Teacher และแสดงผลการแนะนำตัว



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : : Inheritance

ใบสั่งงานที่

4

หน้าที่

แผ่นที่ : 5

เครื่องมือที่ใช้ในการปฏิบัติงาน

- 1.เครื่องคอมพิวเตอร์
- 2.โปรแกรม Edit Plus
- 3 โปรแกรม JDK

เวลาในการปฏิบัติงาน

60 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Polymorphism

ใบสั่งงานที่

5

หน้าที่

แผ่นที่ : 1

วัตถุประสงค์:

1. เข้าใจหลักการของการเขียนโปรแกรมเชิงวัตถุ (OOP) เรื่อง Polymorphism
2. สามารถใช้งานการโอเวอร์โหลดเมธอด (Method Overloading) และการโอเวอร์ไรด์เมธอด (Method Overriding)
3. เข้าใจการใช้ Polymorphism ในการสืบทอดคลาสและการเรียกใช้งานเมธอดในลักษณะที่หลากหลาย
4. เรียนรู้การใช้ Polymorphism ผ่านอินเทอร์เฟซ (Interface)

ทฤษฎีเบื้องต้น

1.1 Polymorphism คืออะไร?

Polymorphism คือความสามารถในการใช้งานวัตถุที่มีพฤติกรรมที่หลากหลายได้โดยใช้โครงสร้างเดียวกัน ซึ่งในภาษาจาวาจะมี 2 ประเภทหลักๆ ได้แก่:

- Compile-time Polymorphism (หรือ Method Overloading)
- Run-time Polymorphism (หรือ Method Overriding)

1.2 Compile-time Polymorphism (Method Overloading)

การโอเวอร์โหลดเมธอด (Method Overloading) เกิดขึ้นเมื่อมีเมธอดหลายตัวในคลาสเดียวกันที่มีชื่อเดียวกันแต่ต่างกันในการพารามิเตอร์ (เช่น จำนวนหรือชนิดของพารามิเตอร์)

```
java

class Calculator {
    // Overloading เมธอด add()
    public int add(int a, int b) {
        return a + b;
    }

    public double add(double a, double b) {
        return a + b;
    }
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Polymorphism

ใบสั่งงานที่
5
แผ่นที่ : 2

หน้า
ที่

1.3 Run-time Polymorphism (Method Overriding)

การโอเวอร์ไรด์เมธอด (Method Overriding) คือการที่ Subclass ทำการเขียนทับเมธอดที่สืบทอดมาจาก Superclass โดยใช้ชื่อเมธอดและพารามิเตอร์แบบเดียวกัน

```
java

class Animal {
    public void sound() {
        System.out.println("Animal is making a sound");
    }
}

class Dog extends Animal {
    @Override
    public void sound() {
        System.out.println("Dog is barking");
    }
}
```

1.4 Polymorphism ผ่านอินเทอร์เฟซ

อินเทอร์เฟซช่วยให้สามารถใช้ Polymorphism ได้มากขึ้น เพราะอินเทอร์เฟซสามารถมีหลายคลาสที่อิมพลีเมนต์ (implement) และแต่ละคลาสสามารถมีพฤติกรรมที่แตกต่างกัน

```
java

interface Shape {
    void draw();
}

class Circle implements Shape {
    public void draw() {
        System.out.println("Drawing a circle");
    }
}

class Rectangle implements Shape {
    public void draw() {
        System.out.println("Drawing a rectangle");
    }
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ

ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented Software Development)

รหัสวิชา : 31901-2006

งาน : เรื่อง Polymorphism

ใบสั่งงานที่

5

แผ่นที่ : 4

หน้าที่

ตัวอย่างการใช้งาน Polymorphism

1.ตัวอย่างการใช้ Compile-time Polymorphism (Method Overloading)

```
class Calculator {
```

```
    public int add(int a, int b) {
```

```
        return a + b;
```

```
    }
```

```
    public double add(double a, double b) {
```

```
        return a + b;
```

```
    }
```

```
}
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Calculator calc = new Calculator();
```

```
        System.out.println("Addition of integers: " + calc.add(5, 10));
```

```
        System.out.println("Addition of doubles: " + calc.add(5.5, 10.5));
```

```
    }
```

```
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Polymorphism

ใบสั่งงานที่

5

แผ่นที่ : 4

หน้าที่

2.ตัวอย่างการใช้ Run-time Polymorphism (Method Overriding)

```
class Animal {  
    public void sound() {  
        System.out.println("Animal is making a sound");  
    }  
}  
  
class Dog extends Animal {  
    @Override  
    public void sound() {  
        System.out.println("Dog is barking");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Animal animal = new Dog(); // ใช้ Polymorphism  
        animal.sound(); // เรียกใช้เมธอดของ Dog  
    }  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Polymorphism

ใบสั่งงานที่

5

แผ่นที่ : 4

หน้าที่

3.ตัวอย่างการใช้ Polymorphism ผ่านอินเทอร์เฟซ

```
interface Shape {
```

```
    void draw();
```

```
}
```

```
class Circle implements Shape {
```

```
    public void draw() {
```

```
        System.out.println("Drawing a circle");
```

```
    }
```

```
}
```

```
class Rectangle implements Shape {
```

```
    public void draw() {
```

```
        System.out.println("Drawing a rectangle");
```

```
    }
```

```
}
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Shape shape1 = new Circle();
```

```
        Shape shape2 = new Rectangle();
```

```
        shape1.draw(); // เรียกใช้เมธอด draw ของ Circle
```

```
        shape2.draw(); // เรียกใช้เมธอด draw ของ Rectangle
```

```
    }
```

```
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Polymorphism

ใบสั่งงานที่
5
แผ่นที่ : 4

หน้าที่

คำสั่ง จงเขียนโปรแกรมภาษาจาวา ดังต่อไปนี้

1

1.1: Method Overloading

1. สร้างคลาส MathOperation ที่มีเมธอด multiply() สำหรับการคูณ:
 - o สร้างเมธอด multiply(int a, int b)
 - o สร้างเมธอด multiply(double a, double b)
2. เขียนโค้ดทดสอบการใช้งานเมธอด multiply() ทั้งสองแบบ

1.2: Method Overriding

1. สร้าง Superclass Employee ที่มีเมธอด work()
2. สร้าง Subclass Manager ที่โอเวอร์ไรด์เมธอด work() เพื่อเปลี่ยนการทำงานของพนักงานประเภท Manager
3. สร้างออบเจกต์ของ Subclass และเรียกใช้เมธอด work()

1.3: Polymorphism ผ่านอินเทอร์เฟซ

1. สร้างอินเทอร์เฟซ Vehicle ที่มีเมธอด start()
2. สร้างคลาส Car และ Bike ที่อิมพลีเมนต์อินเทอร์เฟซ Vehicle และสร้างเมธอด start() ที่แตกต่างกัน
3. สร้างออบเจกต์ของ Car และ Bike แล้วเรียกใช้เมธอด start() ผ่าน Polymorphism

2.1 สร้าง Superclass Person ที่มีเมธอด introduce() เพื่อแนะนำตัว

2.2 สร้าง Subclass Teacher และ Student ที่โอเวอร์ไรด์เมธอด introduce() เพื่อให้เหมาะสมกับการแนะนำตัวของแต่ละบทบาท

2.3 ใช้ Polymorphism เพื่อเรียกใช้เมธอด introduce() ของ Teacher และ Student ผ่านออบเจกต์ของ Person

เครื่องมือที่ใช้ในการปฏิบัติงาน

1. เครื่องคอมพิวเตอร์
2. โปรแกรม Edit Plus
- 3 โปรแกรม JDK

เวลาในการปฏิบัติงาน

60 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Abstraction

ใบสั่งงานที่

6

หน้าที่

แผ่นที่ : 1

วัตถุประสงค์:

1. สามารถอธิบายแนวคิดของ Abstraction ในภาษาจาวาได้
2. สามารถใช้งาน Abstract Class และ Interface ได้
3. สามารถประยุกต์ใช้ Abstraction ในการออกแบบโปรแกรมอย่างง่ายได้

แนวคิดของ Abstraction

Abstraction คืออะไร:

Abstraction หมายถึง การซ่อนรายละเอียดบางส่วนจากระบบ และแสดงเฉพาะข้อมูลที่จำเป็น เพื่อช่วยให้โปรแกรมอ่านง่ายและเข้าใจได้ง่ายขึ้น โดยมักใช้ Abstract Class และ Interface ในการทำ Abstraction ในภาษาจาวา

ประโยชน์ของ Abstraction:

- ช่วยลดความซับซ้อนของระบบ
- ส่งเสริมการใช้งานซ้ำและการดูแลรักษาโค้ด
- ช่วยให้โค้ดมีความยืดหยุ่นมากขึ้น

1 Abstract Class

ลักษณะการใช้งาน:

- abstract class คือคลาสที่ไม่สามารถสร้างวัตถุได้โดยตรง ต้องให้คลาสลูกสืบทอดและนิยามเมธอดที่เป็นแบบ abstract ด้วยตัวเอง
- ใช้เมื่อมีฟังก์ชันบางส่วนที่ต้องการให้คลาสลูกเขียนเอง



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Abstraction

ใบสั่งงานที่

6

แผ่นที่ : 2

หน้า
ที่

ตัวอย่าง

```
abstract class Vehicle {  
    // Abstract method (ไม่มีเนื้อหา)  
    public abstract void move();  
  
    // Concrete method (มีเนื้อหา)  
    public void stop() {  
        System.out.println("The vehicle has stopped.");  
    }  
}  
  
class Car extends Vehicle {  
    public void move() {  
        System.out.println("The car is moving.");  
    }  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Abstraction

ใบสั่งงานที่
6
แผ่นที่ : 3

หน้า
ที่

2 Interface

ลักษณะการใช้งาน:

- interface ใช้เมื่อคลาสหลายคลาสต้องการใช้โครงสร้างแบบเดียวกัน โดยทุกคลาสที่ implement อินเตอร์เฟซต้องนิยามเมธอดเองทั้งหมด
- ใช้ในกรณีที่ต้องการกำหนด "สัญญา" ให้กับคลาสลูกในการสร้างเมธอดต่างๆ

ตัวอย่าง:

```
interface Appliance {
```

```
    void turnOn();
```

```
    void turnOff();
```

```
}
```

```
class Fan implements Appliance {
```

```
    public void turnOn() {
```

```
        System.out.println("The fan is on.");
```

```
    }
```

```
    public void turnOff() {
```

```
        System.out.println("The fan is off.");
```

```
    }
```

```
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Abstraction

ใบสั่งงานที่

6

หน้าที่

แผ่นที่ : 4

คำสั่ง จงเขียนโปรแกรมภาษาจาวา ดังต่อไปนี้

1. จงเขียนโปรแกรมสร้าง Abstract Class ชื่อ Animal ที่มีเมธอด makeSound() และให้สร้างคลาส Dog และ Cat ที่สืบทอดจากคลาส Animal แล้วให้พิมพ์เสียงของสัตว์แต่ละชนิดเมื่อเรียกใช้เมธอด makeSound()
2. จงเขียนโปรแกรมสร้าง Interface ชื่อ ElectronicDevice ที่มีเมธอด powerOn() และ powerOff() และสร้างคลาส TV และ Radio ที่ implement อินเตอร์เฟซนี้ จากนั้นให้เขียนการทำงานของแต่ละเมธอด
3. จงเขียนโปรแกรมสร้าง Abstract Class ชื่อว่า Shape ที่มีเมธอดชื่อ area() (ไม่มีเนื้อหาในเมธอด) และให้สร้างคลาส Circle และ Rectangle ที่สืบทอดจากคลาส Shape และทำการคำนวณพื้นที่ของรูปวงกลมและสี่เหลี่ยม
4. จงเขียนโปรแกรมสร้าง Interface ชื่อว่า Vehicle ที่มีเมธอด startEngine() และ stopEngine() และให้สร้างคลาส Car และ Motorcycle ที่ implement อินเตอร์เฟซ Vehicle และเขียนการทำงานของแต่ละเมธอด

เครื่องมือที่ใช้ในการปฏิบัติงาน

1. เครื่องคอมพิวเตอร์
2. โปรแกรม Edit Plus
- 3 โปรแกรม JDK

เวลาในการปฏิบัติงาน

60 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Exception Handling

ใบสั่งงานที่

7

หน้าที่

แผ่นที่ : 1

วัตถุประสงค์:

1. เข้าใจแนวคิดและความสำคัญของการจัดการข้อผิดพลาด (Exception Handling) ในการเขียนโปรแกรม
2. สามารถใช้คำสั่ง try-catch และ finally เพื่อจัดการข้อผิดพลาดในโปรแกรมได้
3. สามารถเขียนโปรแกรมที่ใช้การโยนและจับข้อยกเว้น (Exception) ได้อย่างถูกต้อง

ทฤษฎี: Exception Handling ในภาษาจาวา

Exception Handling คืออะไร:

ในการเขียนโปรแกรม อาจมีการเกิดข้อผิดพลาดหรือ "exception" ขึ้น เช่น การแบ่งตัวเลขด้วย 0 หรือการพยายามเข้าถึงไฟล์ที่ไม่มีอยู่ การจัดการข้อผิดพลาด (Exception Handling) ในภาษาจาวาช่วยให้โปรแกรมไม่หยุดการทำงานทันทีเมื่อเกิดข้อผิดพลาดขึ้น แต่สามารถจับและแก้ไขข้อผิดพลาดนั้นได้อย่างมีประสิทธิภาพ ประเภทของ Exception:

1. Checked Exception: เป็นข้อผิดพลาดที่เกิดขึ้นในระหว่างการคอมไพล์ เช่น IOException
2. Unchecked Exception: เป็นข้อผิดพลาดที่เกิดขึ้นระหว่างการทำงานของโปรแกรม เช่น ArithmeticException
3. Error: ข้อผิดพลาดที่ร้ายแรงซึ่งโปรแกรมไม่สามารถกู้คืนได้ เช่น OutOfMemoryError

คำสั่งที่ใช้ใน Exception Handling

1. try-catch: ใช้เพื่อจับข้อผิดพลาดที่อาจเกิดขึ้นในบล็อก try และดำเนินการแก้ไขในบล็อก catch
2. finally: เป็นบล็อกที่สามารถใช้หลังจากบล็อก try-catch ซึ่งจะถูกระบุว่าข้อผิดพลาดจะเกิดขึ้นหรือไม่
3. throw: ใช้เพื่อโยน Exception ด้วยตนเอง
4. throws: ใช้ในการประกาศว่าเมธอดนั้นอาจโยน Exception ออกมาได้



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : Exception Handling

ใบสั่งงานที่
7

หน้า
ที่

แผ่นที่ : 2

โครงสร้างทั่วไป

```
java

try {
    // โค้ดที่อาจเกิดข้อยกเว้น
} catch (ExceptionType e) {
    // จัดการข้อยกเว้น
} finally {
    // โค้ดที่ต้องทำเสมอไม่ว่าจะเกิดข้อยกเว้นหรือไม่
}
```

ตัวอย่างการใช้งาน การจับข้อยกเว้นโดยใช้ try-catch

```
public class Example {

    public static void main(String[] args) {

        try {

            int result = 10 / 0; // ข้อยกเว้น:หารด้วยศูนย์

        } catch (ArithmeticException e) {

            System.out.println("Cannot divide by zero!");

        } finally {

            System.out.println("This block is always executed.");

        }

    }

}
```

ผลลัพธ์

```
csharp
```

```
Cannot divide by zero!
This block is always executed.
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Exception Handling

ใบสั่งงานที่

7

หน้า
ที่

แผ่นที่ : 3

ตัวอย่างการใช้ throw เพื่อโยนข้อผิดพลาด

```
public class ThrowExample {  
    public static void validateAge(int age) {  
        if (age < 18) {  
            throw new ArithmeticException("Age must be 18 or above.");  
        } else {  
            System.out.println("Access granted.");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    validateAge(16); // ข้อผิดพลาด: อายุไม่ถึง 18  
}  
}
```

ผลลัพธ์

arduino

คัดลอกโค้ด

```
Exception in thread "main" java.lang.ArithmeticException: Age must be 18 or above.
```



สาขาวิชา : เทคโนโลยีสารสนเทศ

ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)

รหัสวิชา : 31901-2006

งาน : เรื่อง Exception Handling

ใบสั่งงานที่

7

หน้าที่

แผ่นที่ : 4

คำสั่ง จงเขียนโปรแกรมภาษาจาวาดังต่อไปนี้

1. จงเขียนโปรแกรมที่อ่านตัวเลข 2 ตัวจากผู้ใช้ และทำการหารตัวเลขทั้งสอง โดยให้จัดการข้อยกเว้นที่การหารด้วยศูนย์ และแสดงข้อความเตือนที่เหมาะสม

2. จงเขียนโปรแกรมที่ตรวจสอบอายุของผู้ใช้ โดยให้โยนข้อยกเว้น IllegalArgument Exception หากอายุที่กรอกน้อยกว่า 18 และให้พิมพ์ข้อความว่า "ไม่สามารถเข้าใช้บริการได้" พร้อมทั้งใช้บล็อก finally เพื่อแสดงข้อความ "จบการตรวจสอบ"

เครื่องมือที่ใช้ในการปฏิบัติงาน

1. เครื่องคอมพิวเตอร์
2. โปรแกรม Edit Plus
- 3 โปรแกรม JDK

เวลาในการปฏิบัติงาน

60 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Threads

ใบสั่งงานที่

8

หน้าที่

แผ่นที่ : 1

วัตถุประสงค์:

1. เข้าใจแนวคิดของ Threads และการทำงานแบบหลายเธรด (Multithreading)
2. สร้างและจัดการ Threads ในภาษาจาวาได้
3. สามารถใช้คำสั่งต่างๆ ที่เกี่ยวข้องกับ Threads เช่น start(), run(), sleep(), และ join() ได้
4. ประยุกต์ใช้ Threads เพื่อแก้ปัญหาโปรแกรมอย่างง่ายได้
5. แก้ปัญหาเกี่ยวกับการซิงโครไนซ์ (Synchronization) ได้

ทฤษฎี: Threads

Threads คืออะไร:

Thread คือหน่วยการทำงานย่อยของโปรแกรมที่สามารถทำงานควบคู่กันในเวลาเดียวกัน หรือเรียกว่า Multithreading ซึ่งช่วยให้โปรแกรมทำงานหลายอย่างพร้อมกันได้ ทำให้มีประสิทธิภาพมากขึ้น โดยเฉพาะในงานที่มีความซับซ้อน เช่น การประมวลผลกราฟิก หรือการโหลดข้อมูลจากเครือข่าย

การสร้าง Threads ในภาษาจาวา:

สามารถสร้าง Thread ได้สองวิธีหลัก:

1. การสืบทอดจากคลาส Thread
2. การใช้ Runnable interface

1. การสร้าง Threads ด้วยการสืบทอดจากคลาส Thread

```
class MyThread extends Thread {
```

```
    public void run() {
```

```
        for(int i = 0; i < 5; i++) {
```

```
            System.out.println("Thread is running: " + i);
```

```
        }
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        MyThread t = new MyThread();
```

```
        t.start(); // เริ่มทำงานของเธรด
```

```
    }
```

```
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : Threads

ใบสั่งงานที่
8

แผ่นที่ : 2

หน้า
ที่

2.การสร้าง Threads จากการใช้ Runnable Interface

```
class MyRunnable implements Runnable {  
    public void run() {  
        for(int i = 0; i < 5; i++) {  
            System.out.println("Runnable thread is running: " + i);  
        }  
    }  
  
    public static void main(String[] args) {  
        MyRunnable myRunnable = new MyRunnable();  
        Thread t = new Thread(myRunnable);  
        t.start(); // เริ่มทำงานของเธรด  
    }  
}
```

คำสั่งที่สำคัญในการจัดการ Threads

1. start(): ใช้เพื่อเริ่มทำงานของเธรด
2. run(): ใช้ในการกำหนดงานที่จะทำภายในเธรด
3. sleep(long milliseconds): หยุดเธรดชั่วคราวเป็นเวลาที่กำหนด
4. join(): รอให้เธรดอื่นทำงานเสร็จสมบูรณ์ก่อนที่จะดำเนินการต่อ
5. synchronized: ใช้เพื่อจัดการการชิงโครไนซ์ของหลายเธรด เพื่อป้องกันการเข้าถึงข้อมูลพร้อมกันในเธรดหลายตัว



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Threads

ใบสั่งงานที่
8
แผ่นที่ : 3

หน้า
ที่

ตัวอย่างการใช้งาน Threads

ตัวอย่าง 1: การใช้ sleep() เพื่อหน่วงเวลาการทำงานของเธรด

```
class SleepExample extends Thread {  
    public void run() {  
        for (int i = 0; i < 5; i++) {  
            try {  
                Thread.sleep(1000); // หน่วงเวลา 1 วินาที  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
            System.out.println(i);  
        }  
    }  
}  
  
public static void main(String[] args) {  
    SleepExample t1 = new SleepExample();  
    t1.start();  
}  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Threads

ใบสั่งงานที่

8

หน้าที่

แผ่นที่ : 4

ตัวอย่าง 2: การใช้ join() เพื่อรอให้เธรดอื่นทำงานเสร็จ

```
class JoinExample extends Thread {  
    public void run() {  
        for (int i = 1; i <= 3; i++) {  
            System.out.println("Thread " + Thread.currentThread().getId() + " is running");  
        }  
    }  
  
    public static void main(String[] args) {  
        JoinExample t1 = new JoinExample();  
        JoinExample t2 = new JoinExample();  
  
        t1.start();  
        try {  
            t1.join(); // รอให้ t1 ทำงานเสร็จก่อน  
        } catch (InterruptedException e) {  
            System.out.println(e);  
        }  
  
        t2.start();  
    }  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Threads

ใบสั่งงานที่
8
แผ่นที่ : 5

หน้าที่

คำสั่ง จงเขียนโปรแกรมภาษาจาวาดังต่อไปนี้

ข้อที่ 1:

จงเขียนโปรแกรมที่สร้างเธรด 2 ตัว โดยการใช้การสืบทอดจากคลาส Thread ให้แต่ละเธรดพิมพ์ตัวเลขตั้งแต่ 1 ถึง 5 และหน่วงเวลา 1 วินาทีระหว่างการพิมพ์แต่ละครั้ง

ข้อที่ 2:

จงเขียนโปรแกรมที่สร้างเธรด 2 ตัว โดยใช้ Runnable interface และให้แต่ละเธรดพิมพ์ข้อความว่า "เธรดกำลังทำงาน" 3 ครั้ง โดยใช้ sleep() หน่วงเวลาครั้งละ 500 มิลลิวินาที

ข้อที่ 3:

จงเขียนโปรแกรมที่สร้างเธรด 2 ตัว ให้เธรดแรกทำงานก่อน เมื่อทำงานเสร็จแล้วให้เธรดที่สองเริ่มทำงาน โดยใช้คำสั่ง join()

ข้อที่ 4:

จงเขียนโปรแกรมที่ใช้ synchronized เพื่อชิงโครโนซ์การเข้าถึงตัวแปรร่วมกันระหว่าง 2 เธรด โดยให้เธรดแต่ละตัวเพิ่มค่าในตัวแปรร่วมกันทีละ 1 ครั้ง

ข้อที่ 5:

จงเขียนโปรแกรมที่ใช้ synchronized ในการจำลองการถอนเงินจากบัญชีธนาคาร โดยมีหลายเธรดที่พยายามถอนเงินจากบัญชีเดียวกัน โดยต้องจัดการไม่ให้ถอนเงินเกินยอดคงเหลือในบัญชี

เครื่องมือที่ใช้ในการปฏิบัติงาน

1. เครื่องคอมพิวเตอร์
2. โปรแกรม Edit Plus
- 3 โปรแกรม JDK

เวลาในการปฏิบัติงาน

60 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Threads

ใบสั่งงานที่

8

หน้าที่

แผ่นที่ : 6

เฉลยใบงานที่ 1

```
class MyThread extends Thread {  
    public void run() {  
        for (int i = 1; i <= 5; i++) {  
            try {  
                Thread.sleep(1000); // หน่วงเวลา 1 วินาที  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
            System.out.println(i);  
        }  
    }  
}  
  
public static void main(String[] args) {  
    MyThread t1 = new MyThread();  
    MyThread t2 = new MyThread();  
  
    t1.start();  
    t2.start();  
}  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Threads

ใบสั่งงานที่

8

หน้าที่

แผ่นที่ : 7

เฉลยใบงานที่ 2

```
class MyRunnable implements Runnable {  
    public void run() {  
        for (int i = 0; i < 3; i++) {  
            try {  
                Thread.sleep(500); // หน่วงเวลา 500 มิลลิวินาที  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
            System.out.println("เธรดกำลังทำงาน");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    Thread t1 = new Thread(new MyRunnable());  
    Thread t2 = new Thread(new MyRunnable());  
  
    t1.start();  
    t2.start();  
}  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Threads

ใบสั่งงานที่
8
แผ่นที่ : 8

หน้าที่

เฉลยใบงานที่ 3

```
class JoinExample extends Thread {  
    public void run() {  
        for (int i = 1; i <= 3; i++) {  
            System.out.println("Thread " + Thread.currentThread().getId() + " is running");  
        }  
    }  
  
    public static void main(String[] args) {  
        JoinExample t1 = new JoinExample();  
        JoinExample t2 = new JoinExample();  
  
        t1.start();  
        try {  
            t1.join(); // รอให้ t1 ทำงานเสร็จก่อน  
        } catch (InterruptedException e) {  
            System.out.println(e);  
        }  
  
        t2.start();  
    }  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ

ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)

รหัสวิชา : 31901-2006

งาน : เรื่อง Threads

ใบสั่งงานที่

8

แผ่นที่ : 9

หน้าที่

เฉลยใบงานที่ 4

```
class Counter {
```

```
    int count = 0;
```

```
    public synchronized void increment() {
```

```
        count++;
```

```
    }
```

```
}
```

```
class MyThread extends Thread {
```

```
    Counter counter;
```

```
    MyThread(Counter counter) {
```

```
        this.counter = counter;
```

```
    }
```

```
    public void run() {
```

```
        for (int i = 0; i < 1000; i++) {
```

```
            counter.increment();
```

```
        }
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Counter counter = new Counter();
```

```
        MyThread t1 = new MyThread(counter);
```

```
        MyThread t2 = new MyThread(counter);
```

```
        t1.start();
```

```
        t2.start();
```

```
        try {
```

```
            t1.join();
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง Threads

ใบสั่งงานที่
8
แผ่นที่ : 10

หน้าที่

เฉลยใบงานที่ 5

```
class BankAccount {  
    private int balance = 1000;  
    public synchronized void withdraw(int amount) {  
        if (balance >= amount) {  
            System.out.println(Thread.currentThread().getName() + " is withdrawing " + amount);  
            balance -= amount;  
            System.out.println("Remaining balance: " + balance);  
        } else {  
            System.out.println("Not enough balance for " + Thread.currentThread().getName());  
        }  
    }  
}  
  
class WithdrawThread extends Thread {  
    BankAccount account;  
    WithdrawThread(BankAccount account) {  
        this.account = account;  
    }  
    public void run() {  
        account.withdraw(600);  
    }  
}  
  
public class BankApp {  
    public static void main(String[] args) {  
        BankAccount account = new BankAccount();  
        WithdrawThread t1 = new WithdrawThread(account);  
        WithdrawThread t2 = new WithdrawThread(account);  
  
        t1.start();  
        t2.start();  
    }  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่

9

หน้าที่

แผ่นที่ : 1

วัตถุประสงค์:

1. เข้าใจแนวคิดพื้นฐานของการเชื่อมต่อฐานข้อมูล (Database Connection) ในภาษาจาวา
2. ใช้ JDBC (Java Database Connectivity) ในการเชื่อมต่อฐานข้อมูลได้
3. เขียนโปรแกรมเพื่อเชื่อมต่อกับฐานข้อมูล MySQL หรือ SQLite ได้
4. ใช้คำสั่ง SQL เช่น SELECT, INSERT, UPDATE, และ DELETE ผ่านโปรแกรมจาวาได้
5. จัดการกับข้อผิดพลาดที่เกิดขึ้นจากการเชื่อมต่อฐานข้อมูลได้อย่างถูกต้อง

ทฤษฎี: การเชื่อมต่อฐานข้อมูลในภาษาจาวา

การเชื่อมต่อฐานข้อมูลคืออะไร:

การเชื่อมต่อฐานข้อมูลหมายถึงการที่โปรแกรมเชื่อมต่อกับฐานข้อมูลภายนอกเพื่อดึงข้อมูล เพิ่มข้อมูล ลบ หรืออัปเดตข้อมูลได้ โดยในภาษาจาวา เราจะใช้ JDBC (Java Database Connectivity) ซึ่งเป็น API ที่ช่วยในการติดต่อระหว่างโปรแกรมจาวาและฐานข้อมูลประเภทต่างๆ

ขั้นตอนการเชื่อมต่อฐานข้อมูล:

1. โหลดไดรเวอร์ฐานข้อมูล
2. สร้างการเชื่อมต่อ (Connection)
3. สร้างคำสั่ง (Statement หรือ PreparedStatement)
4. ดำเนินการคำสั่ง SQL
5. ปิดการเชื่อมต่อ



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่

9

แผ่นที่ : 2

หน้า
ที่

โครงสร้างทั่วไปในการเชื่อมต่อฐานข้อมูล:

```
import java.sql.*;

public class DatabaseConnectionExample {

    public static void main(String[] args) {

        // ข้อมูลการเชื่อมต่อฐานข้อมูล

        String url = "jdbc:mysql://localhost:3306/mydatabase";

        String username = "root";

        String password = "password";

        try {

            // โหลดไดรเวอร์

            Class.forName("com.mysql.cj.jdbc.Driver");

            // สร้างการเชื่อมต่อ

            Connection connection = DriverManager.getConnection(url, username, password);

            System.out.println("เชื่อมต่อฐานข้อมูลสำเร็จ");

            // ปิดการเชื่อมต่อ

            connection.close();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่

9

หน้า
ที่

แผ่นที่ : 3

.การใช้ SELECT ดึงข้อมูลจากฐานข้อมูล

```
import java.sql.*;
```

```
public class SelectExample {
```

```
    public static void main(String[] args) {
```

```
        String url = "jdbc:mysql://localhost:3306/mydatabase";
```

```
        String username = "root";
```

```
        String password = "password";
```

```
        try {
```

```
            Connection connection = DriverManager.getConnection(url, username, password);
```

```
            String query = "SELECT * FROM students";
```

```
                Statement statement = connection.createStatement();
```

```
                ResultSet resultSet = statement.executeQuery(query);
```

```
                    while (resultSet.next()) {
```

```
                        System.out.println("ID: " + resultSet.getInt("id"));
```

```
                        System.out.println("Name: " + resultSet.getString("name"));
```

```
                        System.out.println("Age: " + resultSet.getInt("age"));
```

```
                    }
```

```
                connection.close();
```

```
            } catch (Exception e) {
```

```
                e.printStackTrace();
```

```
            }
```

```
        }
```

```
    }
```



สาขาวิชา : เทคโนโลยีสารสนเทศ

ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)

รหัสวิชา : 31901-2006

งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่

9

หน้าที่

แผ่นที่ : 4

การใช้ INSERT เพิ่มข้อมูลเข้าสู่ฐานข้อมูล

```
import java.sql.*;
```

```
public class InsertExample {
```

```
    public static void main(String[] args) {
```

```
        String url = "jdbc:mysql://localhost:3306/mydatabase";
```

```
        String username = "root";
```

```
        String password = "password";
```

```
        try {
```

```
            Connection connection = DriverManager.getConnection(url, username, password);
```

```
            String query = "INSERT INTO students (name, age) VALUES (?, ?)";
```

```
            PreparedStatement preparedStatement = connection.prepareStatement(query);
```

```
            preparedStatement.setString(1, "John Doe");
```

```
            preparedStatement.setInt(2, 20);
```

```
            int rows = preparedStatement.executeUpdate();
```

```
            if (rows > 0) {
```

```
                System.out.println("ข้อมูลถูกเพิ่มเรียบร้อยแล้ว");
```

```
            }
```

```
                connection.close();
```

```
        } catch (Exception e) {
```

```
            e.printStackTrace();
```

```
        }
```

```
    }
```

```
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่
9
แผ่นที่ : 5

หน้าที่

การใช้ UPDATE อัปเดตข้อมูลในฐานข้อมูล

```
import java.sql.*;

public class UpdateExample {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydatabase";
        String username = "root";
        String password = "password";

        try {
            Connection connection = DriverManager.getConnection(url, username, password);
            String query = "UPDATE students SET age = ? WHERE name = ?";

            PreparedStatement preparedStatement = connection.prepareStatement(query);
            preparedStatement.setInt(1, 25); // อัปเดตอายุใหม่
            preparedStatement.setString(2, "John Doe");

            int rows = preparedStatement.executeUpdate();
            if (rows > 0) {
                System.out.println("ข้อมูลถูกอัปเดตเรียบร้อยแล้ว");
            }

            connection.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ

ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)

รหัสวิชา : 31901-2006

งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่

9

แผ่นที่ : 6

หน้าที่

การใช้ DELETE ลบข้อมูลจากฐานข้อมูล

```
import java.sql.*;
```

```
public class DeleteExample {
```

```
    public static void main(String[] args) {
```

```
        String url = "jdbc:mysql://localhost:3306/mydatabase";
```

```
        String username = "root";
```

```
        String password = "password";
```

```
        try {
```

```
            Connection connection = DriverManager.getConnection(url, username, password);
```

```
            String query = "DELETE FROM students WHERE name = ?";
```

```
            PreparedStatement preparedStatement = connection.prepareStatement(query);
```

```
            preparedStatement.setString(1, "John Doe");
```

```
            int rows = preparedStatement.executeUpdate();
```

```
            if (rows > 0) {
```

```
                System.out.println("ข้อมูลถูกลบเรียบร้อยแล้ว");
```

```
            }
```

```
            connection.close();
```

```
        } catch (Exception e) {
```

```
            e.printStackTrace();
```

```
        }
```

```
    }
```

```
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่
9
แผ่นที่ : 7

หน้าที่

คำสั่ง จงเขียนโปรแกรมภาษาจาวาดังต่อไปนี้

ข้อที่ 1:

จงเขียนโปรแกรมที่เชื่อมต่อกับฐานข้อมูล MySQL และแสดงผลข้อความ "เชื่อมต่อฐานข้อมูลสำเร็จ" เมื่อการเชื่อมต่อสำเร็จ

ข้อที่ 2:

จงเขียนโปรแกรมเพื่อดึงข้อมูลจากตาราง students โดยใช้คำสั่ง SELECT และแสดงข้อมูลทั้งหมดในคอนโซล

ข้อที่ 3:

จงเขียนโปรแกรมเพื่อเพิ่มข้อมูลนักเรียนใหม่เข้าสู่ตาราง students โดยใช้คำสั่ง INSERT โดยข้อมูลให้นักเรียนกรอกจากคีย์บอร์ด

ข้อที่ 4:

จงเขียนโปรแกรมเพื่ออัปเดตข้อมูลอายุของนักเรียนในตาราง students โดยใช้คำสั่ง UPDATE โดยให้ผู้ใช้กรอกชื่อของนักเรียนและอายุใหม่จากคีย์บอร์ด

ข้อที่ 5:

จงเขียนโปรแกรมเพื่อทำการลบข้อมูลนักเรียนออกจากตาราง students โดยใช้คำสั่ง DELETE โดยให้ผู้ใช้กรอกชื่อของนักเรียนที่ต้องการลบจากคีย์บอร์ด

เครื่องมือที่ใช้ในการปฏิบัติงาน

1. เครื่องคอมพิวเตอร์
2. โปรแกรม Edit Plus
- 3 โปรแกรม JDK

เวลาในการปฏิบัติงาน

60 นาที



สาขาวิชา : เทคโนโลยีสารสนเทศ

ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)

รหัสวิชา : 31901-2006

งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่

9

แผ่นที่ : 8

หน้าที่

เฉลยใบงานที่ 1

```
import java.sql.*;
```

```
public class DatabaseConnectionTest {  
    public static void main(String[] args) {  
        String url = "jdbc:mysql://localhost:3306/mydatabase";  
        String username = "root";  
        String password = "password";  
  
        try {  
            Connection connection = DriverManager.getConnection(url, username, password);  
            System.out.println("เชื่อมต่อฐานข้อมูลสำเร็จ");  
            connection.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ

ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)

รหัสวิชา : 31901-2006

งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่

9

แผ่นที่ : 9

หน้าที่

เฉลยใบงานที่ 2

```
import java.sql.*;
```

```
public class SelectStudents {
```

```
    public static void main(String[] args) {
```

```
        String url = "jdbc:mysql://localhost:3306/mydatabase";
```

```
        String username = "root";
```

```
        String password = "password";
```

```
        try {
```

```
            Connection connection = DriverManager.getConnection(url, username, password);
```

```
            String query = "SELECT * FROM students";
```

```
            Statement statement = connection.createStatement();
```

```
            ResultSet resultSet = statement.executeQuery(query);
```

```
            while (resultSet.next()) {
```

```
                System.out.println("ID: " + resultSet.getInt("id"));
```

```
                System.out.println("Name: " + resultSet.getString("name"));
```

```
                System.out.println("Age: " + resultSet.getInt("age"));
```

```
            }
```

```
            connection.close();
```

```
        } catch (Exception e) {
```

```
            e.printStackTrace();
```

```
        }
```

```
    }
```

```
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่
9
แผ่นที่ : 10

หน้าที่

เฉลยใบงานที่ 3

```
import java.sql.*;
import java.util.Scanner;

public class InsertStudent {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydatabase";
        String username = "root";
        String password = "password";

        Scanner scanner = new Scanner(System.in);
        System.out.println("กรอกชื่อนักเรียน: ");
        String name = scanner.nextLine();
        System.out.println("กรอกอายุ: ");
        int age = scanner.nextInt();
        try {
            Connection connection = DriverManager.getConnection(url, username, password);
            String query = "INSERT INTO students (name, age) VALUES (?, ?)";
            PreparedStatement preparedStatement = connection.prepareStatement(query);
            preparedStatement.setString(1, name);
            preparedStatement.setInt(2, age);

            int rows = preparedStatement.executeUpdate();
            if (rows > 0) {
                System.out.println("ข้อมูลนักเรียนถูกเพิ่มเรียบร้อยแล้ว");
            }

            connection.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)
รหัสวิชา : 31901-2006
งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่
9
แผ่นที่ : 11

หน้าที่

เฉลยใบงานที่ 4

```
import java.sql.*;
import java.util.Scanner;

public class UpdateStudentAge {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydatabase";
        String username = "root";
        String password = "password";
        Scanner scanner = new Scanner(System.in);
        System.out.println("กรอกชื่อนักเรียนที่ต้องการอัปเดตอายุ: ");
        String name = scanner.nextLine();
        System.out.println("กรอกอายุใหม่: ");
        int newAge = scanner.nextInt();
        try {
            Connection connection = DriverManager.getConnection(url, username, password);
            String query = "UPDATE students SET age = ? WHERE name = ?";
            PreparedStatement preparedStatement = connection.prepareStatement(query);
            preparedStatement.setInt(1, newAge);
            preparedStatement.setString(2, name);

            int rows = preparedStatement.executeUpdate();
            if (rows > 0) {
                System.out.println("อายุของนักเรียนถูกอัปเดตเรียบร้อยแล้ว");
            }
            connection.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ

ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ (Object-oriented
Software Development)

รหัสวิชา : 31901-2006

งาน : เรื่อง การเชื่อมต่อฐานข้อมูล

ใบสั่งงานที่

9

แผ่นที่ : 12

หน้าที่

เฉลยใบงานที่ 5

```
import java.sql.*;
```

```
import java.util.Scanner;
```

```
public class DeleteStudent {
```

```
    public static void main(String[] args) {
```

```
        String url = "jdbc:mysql://localhost:3306/mydatabase";
```

```
        String username = "root";
```

```
        String password = "password";
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("กรอกชื่อนักเรียนที่ต้องการลบ: ");
```

```
        String name = scanner.nextLine();
```

```
        try {
```

```
            Connection connection = DriverManager.getConnection(url, username, password);
```

```
            String query = "DELETE FROM students WHERE name = ?";
```

```
            PreparedStatement preparedStatement = connection.prepareStatement(query);
```

```
            preparedStatement.setString(1, name);
```

```
            int rows = preparedStatement.executeUpdate();
```

```
            if (rows > 0) {
```

```
                System.out.println("ข้อมูลนักเรียนถูกลบเรียบร้อยแล้ว");
```

```
            }
```

```
            connection.close();
```

```
        } catch (Exception e) {
```

```
            e.printStackTrace();
```

```
        }
```

```
    }
```

```
}
```



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การติดตั้งโปรแกรม Edit plus
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 1_1

แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1. ดาวน์โหลดโปรแกรมได้ถูกต้อง 2. ติดตั้งโปรแกรมได้ถูกต้อง 3. กำหนดค่าการ Compile ได้ถูกต้อง 4. กำหนดค่าการ Run ได้ถูกต้อง 5. กำหนดค่ารูปแบบการใช้งานโปรแกรม Edit plus ได้ถูกต้อง			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
1. ความถูกต้อง 2. ความรับผิดชอบ 3. ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การติดตั้งโปรแกรม jdk
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 1

แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1.ดาวน์โหลดโปรแกรมได้ถูกต้อง 2.ติดตั้งโปรแกรมได้ถูกต้อง 3.กำหนดค่ารูปแบบการใช้งานโปรแกรม ได้ถูกต้อง			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
1.ความถูกต้อง 2.ความรับผิดชอบ 3.ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การสร้าง คลาส และ ออบเจกต์
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 2

แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1.สร้างคลาส Student ได้ถูกต้อง 2.สร้างคอนสแตนท์ตัวแปรพารามิเตอร์ ได้ถูกต้อง 3.สร้างเมธอด display() ได้ถูกต้อง 4.สร้างออบเจกต์ ได้ถูกต้อง 5.สร้างคลาส Main ได้ถูกต้อง 6. สร้างออบเจกต์ของคลาส Student และแสดงผลข้อมูลนักเรียนได้ถูกต้อง 7. คอมไพล์โปรแกรมเพื่อตรวจสอบข้อผิดพลาด 8. รันโปรแกรมเพื่อให้ได้ผลรันตามที่กำหนด			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
1. ความถูกต้อง 2. ความรับผิดชอบ 3. ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : Encapsulation
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 3
แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1.สร้างคลาส Product ได้ถูกต้อง 2.สร้างเมธอด getter ได้ถูกต้อง 3.สร้างเมธอด setter ได้ถูกต้อง 4.สร้างคอนสตรัคเตอร์สำหรับตั้งค่าคุณสมบัติเริ่มต้น ได้ถูกต้อง 5.สร้างออบเจ็กต์จากคลาส Product และแสดงค่าของคุณสมบัติ name และ price ได้ถูกต้อง 6.ใช้ setter เพื่อเปลี่ยนค่า name และ price แล้วแสดงผลใหม่ ได้ถูกต้อง 7.คอมไพล์โปรแกรมเพื่อตรวจสอบข้อผิดพลาด 8.รันโปรแกรมเพื่อให้ได้ผลลัพธ์ตามที่กำหนด			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
4. ความถูกต้อง 5. ความรับผิดชอบ 6. ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การสืบทอดคุณสมบัติ(Inheritance)
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 4

แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1.สร้าง Superclass ชื่อ Animal ได้ถูกต้อง 2.สร้างเมธอด makeSound() ได้ถูกต้อง 3.สร้าง Subclass ชื่อ Dog ที่สืบทอดจาก Animal: ได้ถูกต้อง 4.สร้างออบเจกต์จากคลาส Dog ถูกต้อง 5. Override เมธอด makeSound() ให้พิมพ์ข้อความ "Dog is barking" ได้ถูกต้อง 6.สร้างออบเจกต์จากคลาส Dog 7.ใช้เมธอด makeSound() ของคลาส Dog และแสดงผล 8. คอมไพล์โปรแกรมเพื่อตรวจสอบข้อผิดพลาด 9. รันโปรแกรมเพื่อให้ได้ผลรันตามที่กำหนด			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
7. ความถูกต้อง 8. ความรับผิดชอบ 9. ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : Polymorphism
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 5
แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1.สร้างคลาส MathOperation ได้ถูกต้อง 2.สร้างเมธอด multiply(int a, int b) ได้ถูกต้อง 3.สร้างเมธอด multiply(double a, double b) ได้ถูกต้อง 4. สร้าง Superclass Employee ที่มีเมธอด work() ได้ถูกต้อง 5.สร้าง Subclass Manager ที่โอเวอร์ไรด์เมธอด work() เพื่อเปลี่ยนการทำงาน ของพนักงานประเภท Manager ได้ถูกต้อง 6.สร้างออบเจกต์ของ Subclass และเรียกใช้เมธอด work() ได้ถูกต้อง 7. คอมไพล์โปรแกรมเพื่อตรวจสอบข้อผิดพลาด 8. รันโปรแกรมเพื่อให้ได้ผลรันตามที่กำหนด			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
1.ความถูกต้อง 2.ความรับผิดชอบ 3.ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : งานการซ่อนรายละเอียด(Abstraction)
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 6
แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1.สร้าง Abstract Class ชื่อ Animal ได้ถูกต้อง 2.สร้าง เมธอด makeSound() ได้ถูกต้อง 3.สร้างคลาส Dog และ Cat ที่สืบทอดจากคลาส Animal แล้วให้พิมพ์เสียงของสัตว์แต่ละชนิดเมื่อเรียกใช้เมธอด makeSound()ได้ถูกต้อง 4. คอมไพล์โปรแกรมเพื่อตรวจสอบข้อผิดพลาด 5. รันโปรแกรมเพื่อให้ได้ผลรันตามที่กำหนด			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
1.ความถูกต้อง 2.ความรับผิดชอบ 3.ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การจัดการข้อผิดพลาด Exception Handling
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 7

แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1.ประกาศ ตัวแปร ได้ถูกต้อง 2.กำหนดค่า ให้กับตัวแปร ได้ถูกต้อง 3.เขียนคำสั่งจัดการข้อผิดพลาดกรณีที่มีการหารด้วยศูนย์ และแสดงข้อความเตือนที่เหมาะสมได้ถูกต้อง 4. คอมไพล์โปรแกรมเพื่อตรวจสอบข้อผิดพลาด 5. รันโปรแกรมเพื่อให้ได้ผลรันตามที่กำหนด			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
1.ความถูกต้อง 2.ความรับผิดชอบ 3.ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การทำงานหลายอย่างพร้อมกัน(Threads)
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 8
แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1.ประกาศ ตัวแปร ได้ถูกต้อง 2.กำหนดค่า ให้กับตัวแปร ได้ถูกต้อง 3.เขียนคำสั่ง ประมวลผล ได้ถูกต้อง 4. ตรวจสอบเงื่อนไขได้ถูกต้อง 5.เขียนคำสั่งแสดงผล ได้ถูกต้อง 6. คอมไพล์โปรแกรมเพื่อตรวจสอบข้อผิดพลาด 7. รันโปรแกรมเพื่อให้ได้ผลรันตามที่กำหนด			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
1.ความถูกต้อง 2.ความรับผิดชอบ 3.ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)



สาขาวิชา : เทคโนโลยีสารสนเทศ
ชื่อวิชา : การพัฒนาซอฟต์แวร์เชิงวัตถุ
(Object-oriented Software Development)
รหัสวิชา : 31901-2006
งาน : การเขียนโปรแกรมเชื่อมต่อฐานข้อมูล
ชื่อ-สกุล.....เลขที่.....

ใบประเมินผล
การปฏิบัติงาน
ที่ 9

แผ่นที่ :

หน้าที่

จุดประเมิน	ผลการประเมิน		หมายเหตุ
	ผ่าน	ไม่ผ่าน	
1.เขียนโปรแกรมที่เชื่อมต่อกับฐานข้อมูล MySQL และแสดงผลข้อความ "เชื่อมต่อฐานข้อมูลสำเร็จ" เมื่อการเชื่อมต่อสำเร็จได้ถูกต้อง 2.เขียนโปรแกรมเพื่อดึงข้อมูลจากตาราง students โดยใช้คำสั่ง SELECT และแสดงข้อมูลทั้งหมดในคอนโซล ได้ถูกต้อง 3.เขียนโปรแกรมเพื่อเพิ่มข้อมูลนักเรียนใหม่เข้าสู่ตาราง students โดยใช้คำสั่ง INSERT โดยข้อมูลให้นักเรียนกรอกจากคีย์บอร์ดได้ถูกต้อง 4.เขียนโปรแกรมเพื่ออัปเดตข้อมูลอายุของนักเรียนในตาราง students โดยใช้คำสั่ง UPDATE โดยให้ผู้ใช้กรอกชื่อของนักเรียนและอายุใหม่จากคีย์บอร์ดได้ถูกต้อง 5.เขียนโปรแกรมเพื่อทำการลบข้อมูลนักเรียนออกจากตาราง students โดยใช้คำสั่ง DELETE โดยให้ผู้ใช้กรอกชื่อของนักเรียนที่ต้องการลบจากคีย์บอร์ดได้ถูกต้อง 6.คอมไพล์โปรแกรมเพื่อตรวจสอบข้อผิดพลาด 7.รันโปรแกรมเพื่อให้ได้ผลลัพธ์ตามที่กำหนด			

จุดประเมิน	ผลการประเมิน				หมายเหตุ
	ดีมาก	ดี	พอใช้	แก้ไข	
1.ความถูกต้อง 2.ความรับผิดชอบ 3.ความสะอาด					

เวลาเริ่ม น. เวลาเสร็จ น. รวม เวลา ชม. นาที

สรุปผลการประเมิน

ผู้ประเมิน

(นางสุปรียา รัตนวิทยาพันธุ์)