

บทที่ 9 : การจัดการกับข้อผิดพลาด

วัตถุประสงค์

- เพื่อให้เข้าใจข้อผิดพลาดที่อาจเกิดขึ้นได้ในภาษาจาวา
- เพื่อให้เข้าใจคลาสที่เกี่ยวข้องกับการจัดการกับข้อผิดพลาดที่กำหนดไว้ใน Java API
- เพื่อให้เข้าใจวิธีการเรียกใช้คำสั่งที่ใช้ในการจัดการกับข้อผิดพลาดคือ คำสั่ง try, catch และ finally
- เพื่อให้สามารถจัดการกับข้อผิดพลาด และสร้างคลาสประเภทข้อผิดพลาดขึ้นมาใหม่ได้

แบบฝึกหัดเชิงปฏิบัติการ

แบบฝึกหัดที่ 1 การจัดการกับข้อผิดพลาด

วัตถุประสงค์

1. เพื่อให้เข้าใจหลักการของการจัดการกับข้อผิดพลาดในภาษาจาวา
2. เพื่อให้เข้าใจการใช้คำสั่ง try/catch

ขั้นตอนการปฏิบัติการ

- 1) จงเขียนคลาสที่ชื่อ ExceptionDemo โดยมีเมธอด main() อยู่ภายในคลาส และมีคำสั่งต่างๆ ดังนี้
 - (a) ให้เขียนคำสั่งเพื่อรับข้อมูลที่เป็นเลขจำนวนจริง 3 ค่าผ่านทาง command line เพื่อกำหนดเป็นค่าสัมประสิทธิ์ของสมการ $ax^2 + bx + c = 0$ โดยใช้เมธอด Double.parseDouble() ในการแปลงชนิดข้อมูล string ให้เป็นชนิด double
 - (b) ให้เขียนคำสั่งกำหนดค่าเพื่อคำนวณหาคำตอบของ x จากสูตร
 - (c) ให้เขียนคำสั่งเพื่อแสดงผลลัพธ์ของ x_1 และ x_2 ออกทางจอภาพ
- 2) คอมไพล์โปรแกรม ExceptionDemo.java
- 3) ทดลองรันโปรแกรมโดยป้อนสัมประสิทธิ์ที่มีค่าต่างๆ
- 4) ปรับปรุงโปรแกรมในคลาส ExceptionDemo เพื่อให้สามารถจัดการกับข้อผิดพลาดประเภท ArithmeticException และ NumberFormatException โดยใช้คำสั่ง try/catch
- 5) คอมไพล์และทดลองรันโปรแกรมใหม่

แบบฝึกหัดที่ 2 การสร้างคลาสประเภท Exception

วัตถุประสงค์

- 1) เพื่อให้เข้าใจขั้นตอนการสร้างคลาสประเภท Exception
- 2) เพื่อให้เข้าใจคำสั่งในการส่งอ็อบเจกต์ประเภท Exception ขึ้นมาเมื่อเกิดข้อผิดพลาด

ขั้นตอนการปฏิบัติการ

- 1) คัดลอกไฟล์ Account.java และ CheckingAccount.java จากบทที่ 6
- 2) จงเขียนคลาสที่ชื่อ WithdrawException โดยมีขั้นตอนดังนี้
 - (a) ประกาศคลาสให้สืบทอดมาจากคลาส Exception

- (b) กำหนด constructor แบบ default โดยมีคำสั่ง `super()` อยู่ภายใน
- (c) กำหนด constructor ที่มี argument ที่ชื่อ `s` ซึ่งเป็น `String` และคำสั่ง `super(s)` อยู่ภายใน
- 3) จงปรับปรุงเมธอด `withdraw()` ที่อยู่ในคลาส `Account` และ `CheckingAccount` ให้มีรูปแบบเมธอดดังนี้
- ```
public void withdraw(double amount)
 throws WithdrawException {
}
```
- (a) ให้ปรับปรุงคำสั่ง `if` ภายในเมธอด `withdraw()` ในกรณีที่เงินในบัญชีน้อยกว่ายอดเงินที่ต้องการจะถอน ให้ส่งอีอบเจกต์ของคลาส `WithdrawException` คืนมา โดยใช้คำสั่ง
- ```
throw new WithdrawException()
```
- 4) จงปรับปรุงโปรแกรมในคลาสที่ชื่อ `Teller.java` เพื่อให้สามารถจัดการกับข้อผิดพลาดที่เกิดขึ้นโดยใช้คำสั่ง `try/catch`
- 5) คอมไพล์และรันโปรแกรม `Teller.java`

แบบฝึกหัดทบทวน

1. จงปรับปรุงโปรแกรมต่อไปนี้เพื่อให้สามารถคอมไพล์ผ่านได้เขียนโปรแกรมดังนี้

```
import java.io.*;
public class Ex9_1 {

    public static void main(String args[]) {

        String file1 = args[0];
        String file2 = args[1];

        FileInputStream fin = new FileInputStream(file1);
        FileOutputStream fout = new FileOutputStream(file2);
        byte b[] = new byte[1024];

        int num;
        do {
            num = fin.read(b);
            if (num != -1) {
                fout.write(b, 0, num);
            }
        } while (num != -1);
        fin.close();
        fout.close();
    }
}
```

2. จงปรับปรุงโปรแกรมในข้อ 1 เพื่อไม่ให้เกิดข้อผิดพลาดตอนรันโปรแกรม