05506007 Operating Systems (1/2568) ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
รหัสนักศึกษา.....ชื่อ-สกุล.....

ปฏิบัติการ ครั้งที่ 7 – Java Thread

วัตถุประสงค์ของการปฏิบัติการ

- ใช้งาน Java Thread ด้วยวิธี
  - Extends จาก Thread Class โดยตรง
  - หรือ Implement จาก Runnable Interface
- ใช้งาน Java Thread Cancellation ได้
- ใช้งาน Java Thread Pools ได้
- นำ Java Thread ไปประยุกต์ใช้งานได้

1. Java Thread using Thread Class and Runnable Interface

แก้ไข Code ที่ 1 ให้สามารถ Compile ได้ ทดสอบรัน ศึกษาผลลัพธ์และตอบคำถาม 1.1 - 1.5

Code ที่ 1

No.	File Name: OSLab7_01.java
1	class Worker1 extends Thread {
2	public void run() {
3	System.out.println("I'm a worker1");
4	}
5	}
6	class Worker2 **.....1.1.....** {
7	public void run() {
8	System.out.println("I'm a worker2");
9	}
10	}
11	public class OSLab7_01 {
12	public static void main(String [] args) {

```
13                  .........1.2.........
14                  wk1.start();
15                  Worker2 wk2 = new Worker2();
16                  Thread thd = new Thread(wk2);
17                  thd.start();
18                  try {
19                          .........1.3.........
20                  } catch (InterruptedException ie) {
21                          System.out.println(ie);
22                  }
23                  System.out.println("I'm a master");
24          }
25  }
```

1.1. .....

1.2. .....

1.3. .....

1.4. หากต้องการเปลี่ยน Code ในบรรทัดที่ 15 และ 16 ให้เหลือเพียงหนึ่งบรรทัด จะต้องเขียนอย่างไร .....

.....

1.5. แสดงผลลัพธ์ของโปรแกรม .....

.....

.....

.....

2. Java Thread Cancellation (Deferred Cancellation)

แก้ไข Code ที่ 2 ให้สามารถ Compile ได้ ทดสอบรัน ศึกษาผลลัพธ์และตอบคำถาม 2.1 - 2.5

Code ที่ 2

No.	File Name: OSLab7_02.java
1	public class .....2.1..... {
2	public static void main(String argv[]) {
3	Task task = new Task();
4	task.start();
5	try {

```
6                       Thread.sleep(1000);
7                 } catch (InterruptedException ie) {
8                 }
9                 task.interrupt();
10        }
11  }
12  class Task ..........2.2..........  {
13        int count = 0;
14        public void ..........2.3..........  {
15                while (true) {
16                        count += 1;
17                        System.out.println("I'm doing work#" + count);
18                        System.out.println("I'm done for work#" + count);
19                        if (Thread.currentThread().isInterrupted()) {
20                                System.out.println("I'm interrupted !!!");
21                                ..........2.4..........
22                        }
23                }
24        }
25  }
```

2.1. .....

2.2. .....

2.3. .....

2.4. .....

2.5. ในบรรทัดที่ 6 หากใช้การเรียก sleep() method จาก task object แทน Thread Class สามารถทำได้หรือไม่ เพราะอะไร (จงอธิบาย) .....

.....

.....

.....

.....

.....

3. Java Thread Pool

แก้ไข Code ที่ 3 ให้สามารถ Compile ได้ ทดสอบรัน ศึกษาผลลัพธ์และตอบคำถาม 3.1 - 3.5

Code ที่ 3

No.	File Name: OSLab7_03.java
1	class OSLab7_03 {
2	public static void main(String [] args) {
3	int numTask = Integer.parseInt(.....3.1.....);
4	ExecutorService pool = Executors.newCachedThreadPool();
5	for (int i = 0; i < numTask; i++) {
6	Runner runner = new Runner(i);
7	pool.execute(runner);
8	}
9	pool.shutdown();
10	}
11	}
12	class .....3.2..... implements Runnable {
13	private int num;
14	Runner(int num) {
15	.....3.3.....
16	}
17	public void run() {
18	System.out.println("I'm working on task #" + num + ", thread id: " + Thread.currentThread().getId());
19	}
20	}

หมายเหตุ: ขณะรันโปรแกรมให้ใส่ argument เพื่อระบุจำนวน Thread ด้วย เช่น "java OSLab8_03 **10**" เป็นต้น

3.1. .....

3.2. .....

3.3. .....

3.4. หากต้องการกำหนดจำนวน Thread Pool ให้เท่ากับตัวเลขที่ระบุผ่าน argument ขณะรันโปรแกรม จะต้องกำหนด Statement อย่างไรในบรรทัดที่ 4 .....

.....

3.5. จงอธิบายความแตกต่างระหว่างผลลัพธ์ของโปรแกรมเมื่อใช้ Single thread executor และ Cached thread pool .....
.....
.....
.....
.....
.....

4. More Understand Java Thread

แก้ไข Code ที่ 4 ให้สามารถ Compile ได้ ทดสอบรัน ศึกษาผลลัพธ์และตอบคำถาม 4.1 - 4.5

Code ที่ 4

No.	File Name: OSLab7_04.java
1	public class OSLab7_04 {
2	public static void main(String [] args) {
3	int num = 10;
4	MyObject obj = new MyObject();
5	Thread thd01 = new Thread(new Task01(obj, 20));
6	Thread thd02 = new Thread(new Task02(obj, 30));
7	.....4.1.....
8	try {thd01.join(); thd02.join();} catch (Exception e){}
9	obj.calculated(num);
10	System.out.println("From master#0 thread ID: " + Thread.currentThread().getId() + ", value is " + obj.getValue());
11	}
12	}
13	class Task01 .....4.2..... {
14	private int num;
15	private MyObject obj;
16	Task01(MyObject obj, int num) {
17	this.num = num;
18	this.obj = obj;
19	}

```
20      public void run() {
21          obj.calculated(num);
22          System.out.println("From worker#1 thread ID: "
                    + Thread.currentThread().getId()
                    + ", value is " + obj.getValue());
23      }
24  }
25  class Task02 implements Runnable {
26      private int num;
27      private MyObject obj;
28      Task02(MyObject obj, int num) {
29          this.num = num;
30          this.obj = obj;
31      }
32      public void run() {
33          obj.calculated(num);
34          System.out.println("From worker#2 thread ID: "
                    + Thread.currentThread().getId()
                    + ", value is " + obj.getValue());
35      }
36  }
37  class MyObject {
38      private int value;
39      MyObject() {
40          setValue(0);
41      }
42      .........4.3..... void calculated(int num) {
43          value += num;
44      }
45      private void setValue(int value) {
46          this.value = value;
47      }
48      public int getValue() {
49          return value;
50      }
51  }
```

4.1. .....

4.2. .....

4.3. .....

4.4. รันโปรแกรมจำนวน 10 ครั้ง ได้ผลลัพธ์จากการรันโปรแกรมกี่รูปแบบ และจงแสดงผลลัพธ์ดังกล่าว .....

.....

.....

.....

.....

.....

4.5. หากใส่ Comment ในบรรทัดที่ 8 - Compile Code และทดสอบรันโปรแกรมจำนวน 10 ครั้ง จงอธิบายผลลัพธ์จากการรันโปรแกรมทั้ง 10 ครั้ง ว่ามีผลลัพธ์เหมือนหรือแตกต่างกันอย่างไร เพราะเหตุใด .....

.....

.....

.....

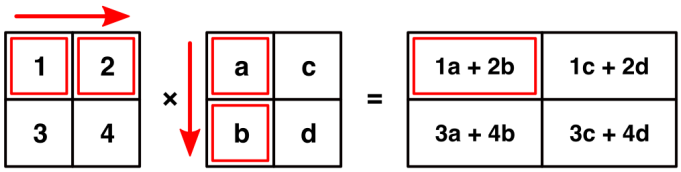
.....

.....

5. Java Thread Application (Matrix Multiplication)

จงพัฒนาโปรแกรม Matrix Multiplication ผ่าน Java Source Code ที่กำหนดให้ ดัง Code ที่ 5 โดยประยุกต์ใช้ความรู้ของ Java Thread ทำให้โปรแกรมสามารถคำนวณได้เร็วขึ้น

ข้อกำหนดที่สำคัญของการคูณ Matrix คือ Column ของ Matrix ตัวตั้งจะต้องเท่ากันกับ Row ของ Matrix ตัวคูณ และมีหลักการคูณโดยผลลัพธ์ที่ตำแหน่งที่พิจารณาอยู่จะเป็นผลรวมของการคูณกันระหว่าง Row กับ Column ของ Matrix ตัวตั้งและตัวคูณตามลำดับ แสดงดังรูปที่ 1



รูปที่ 1 หลักการคูณ Matrix

แสดงตัวอย่างผลลัพธ์จากการคูณ Matrix ของโปรแกรม ที่กำหนด Matrix ตัวตั้งและตัวคูณ (Input) ดังรูปที่ 2 (a) และให้ค่าผลลัพธ์ (Output) ดังรูปที่ 2 (b)

```
matrixA[][] = {{5, 10, 13}, {12, 22, 9}};
matrixB[][] = {{11, 21}, {51, 12}, {20, 18}};
```

(a)

```
[[825, 459], [1434, 678]]
```

(b)

รูปที่ 2 แสดงตัวอย่าง Input และ Output ของโปรแกรม

Code ที่ 5

No.	File Name: OSLab7_MatrixMulThread.java
1	import java.util.Arrays;
2	public class Lab7_05 {
3	public static void main(String args[]) {
4	int matrixA[][] = {{5, 10, 13}, {12, 22, 9}};
5	int matrixB[][] = {{11, 21}, {51, 12}, {20, 18}};
6	MyData dataA = new MyData(matrixA);
7	MyData dataB = new MyData(matrixB);
8	int matrixC_row = dataA.data.length;
9	int matrixC_col = dataB.data[0].length;
10	MyData dataC = new MyData(matrixC_row, matrixC_col);
11	
12	<< You Code Here >>
13	
14	dataC.show();
15	}
16	}
17	class MatrixMulThread implements Runnable {
18	private int processingRow;
19	private int processingCol;
20	private MyData dataA;
21	private MyData dataB;
22	private MyData dataC;
23	MatrixMulThread(int targetRow, int targetCol, MyData dataA, MyData dataB, MyData dataC) {
24	processingRow = targetRow;
25	processingCol = targetCol;
26	this.dataA = dataA;
27	this.dataB = dataB;

```
28                  this.dataC = dataC;
29          }
30          public void run() {
31                  int sum = 0;
32
33                          << You Code Here >>
34
35          }
36  }
37  class MyData {
38          public int data[][];
39          MyData(int data[][]) {
40                  this.data = data;
41          }
42          MyData(int row, int col) {
43                  data = new int[row][col];
44                  for (int aRow[]: data) {
45                          Arrays.fill(aRow, 0);
46                  }
47          }
48          public void show() {
49                  System.out.println(Arrays.deepToString(data));
50          }
51  }
```

จงเขียน/พิมพ์/แปะ Java Code ของโปรแกรม Matrix Multiplication ที่สมบูรณ์ลงในช่องว่างด้านล่าง