ชื่อ-นามสกุล นาย [illegible] รหัสประจำตัวนักศึกษา [illegible] ปีการศึกษา 2567

แบบฝึกหัดปฏิบัติการคาบที่ 11: Thread

คำสั่ง ให้ศึกษาหลักการของเธรดต่อไปนี้

1 เธรด จะแตกต่างจาก Process ที่ทำงานภายใต้ระบบปฏิบัติการแบบ multi-tasking ตรงที่ process แต่ละ process จะมีความเป็นอิสระจากกัน แต่เธรดแต่ละเธรดอาจใช้ข้อมูลร่วมกัน คลาสแบบเธรดในภาษาจาวาคือคลาสที่สืบทอดมาจากคลาสที่ชื่อ Thread หรือคลาสที่ implements อินเตอร์เฟสชื่อ Runable ภายในคลาสแบบเธรดจะต้องมีเมธอด run() ที่ไม่มีการรับอาร์กูเม้นต์ใด ๆ เข้ามา สถานะของ เธรดอาจจะเป็น New runnable running blocked หรือ dead

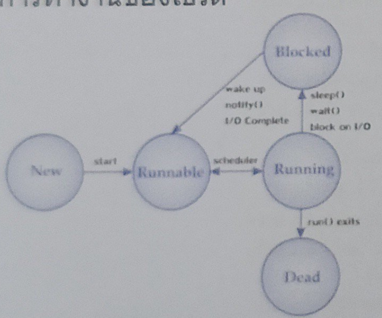
วงจรชีวิตของแต่ละเธรดเริ่มต้นที่ขั้น new เมื่อถูกเริ่มสร้างและจะอยู่ที่ขั้นนี้จนกระทั่งเมธอด start ถูกเรียกประมวลผล ซึ่งจะทำให้เธรดนั้นๆ อยู่ที่ขั้น ready โดยที่เธรดที่มี priority สูงที่สุดจะถูก ทำงานก่อนโดยเขาสู่ขั้น running แต่ละเธรดจะเข้าสู่ขั้นสุดท้ายของวงจรชีวิตคือ dead เมื่อเธรดนั้น ทำงานในเมธอด run เสร็จเรียบร้อยหรือเมื่อถูกหยุดการทำงานด้วยเหตุผลใดๆ ก็ตาม ออปเจ็คแบบเธรดเมื่อลงทะเบียนไว้กับตัวตารางเวลาแล้ว อาจยังไม่มีการรันโปรแกรมโดยทันที แต่ทั้งนี้จะขึ้นอยู่กับสถานะของออปเจ็ค

ตัวอย่าง

```
class PrintName implements Runnable {
    String name;
    public PrintName(String n) {
        name = n;
    }
    public void run() {
        for(int i=0; i<100; i++) {
            System.out.println(name);
        }
    }
}
public class PrintNameThread {
  public static void main(String args[]) {
            PrintName p1 = new PrintName("Thana");
            PrintName p2 = new PrintName("Somchai");
            Thread t1 = new Thread(p1);
            Thread t2 = new Thread(p2);
            t1.start();
            t2.start();
  }
}
```

```
class PrintName extends Thread {
    String name;
    public PrintName(String n) {
  name = n;
    }
    public void run() {
        for(int i=0; i<100; i++) {
        System.out.println(name);
  }
    }
}
public class PrintNameThread {
  public static void main(String args[]) {
  PrintName p1 = new PrintName();
  PrintName p2 = new PrintName();
  p1.run();
  p2.run();
  }
}
```

วงจรการทำงานของเธรด



2.จงอธิบายผลลัพธ์ของการทำงานโดยการใช้ Thread ต่อไปนี้

```
class Thread1 extends Thread{
   Thread1(String name){
      super(name);
   }
   public void run(){
      for(int i=0;i<10;i++){
         System.out.println(getName()+" ");
      }
   }
}
class TestThread1{
   public static void main(String args[]){
      new Thread1("A").start();
      new Thread1("B").start();
   }
}
```

ผลลัพธ์รันในแต่ละรอบจะไม่เหมือนกันแต่จะมีการแสดงชื่อของ Thread นั้นจนครบ 10 ครั้ง

ชื่อ-นามสกุล นาย กฤษธาวร ขมภิเดช รหัสประจำตัวนักศึกษา 6604062620091 ปีการศึกษา 2567

```java
68          // Draw the box
69          g.setColor(Color.BLACK);
70          g.fillRect(0, 0, BOX_WIDTH, BOX_HEIGHT);
71  
72          // Draw the ball
73          g.setColor(Color.BLUE);
74          g.fillOval((int) (ballX - ballRadius), (int) (ballY - ballRadius),
75                (int)(2 * ballRadius), (int)(2 * ballRadius));
76  
77          // Display the ball's information
78          g.setColor(Color.WHITE);
79          g.setFont(new Font("Courier New", Font.PLAIN, 12));
80          StringBuilder sb = new StringBuilder();
81          Formatter formatter = new Formatter(sb);
82          formatter.format("Ball @(%3.0f,%3.0f) Speed=(%2.0f,%2.0f)", ballX, ballY,
83                ballSpeedX, ballSpeedY);
84          g.drawString(sb.toString(), 20, 30);
85       }
86  
87       /** main program (entry point) */
88       public static void main(String[] args) {
89          // Run GUI in the Event Dispatcher Thread (EDT) instead of main thread.
90          javax.swing.SwingUtilities.invokeLater(new Runnable() {
91             public void run() {
92                // Set up main window (using Swing's Jframe)
93                JFrame frame = new JFrame("A Bouncing Ball");
94                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
95                frame.setContentPane(new BouncingBallSimple());
96                frame.pack();
97                frame.setVisible(true);
98             }
99          });
100      }
101 }
```

จงอธิบายผลลัพธ์ของการทำงานโดยที่มีการใช้ Thread ยกตัวอย่างพร้อมอธิบายบรรทัดที่มีการใช้

บรรทัดที่ 28 - 60

=> เริ่มโดยการสร้าง Thread แบบ Anonymous จึงไม่ต้องมี Overwrite และสามารถเขียนที่นี่ได้เลย ในส่วนของ method run ข้างในจะมีการคำนวณ สำหรับลูกบอลแล้ว repaint และThread นี้จะหยุดทุก 1 วินาที / UpdateRate ซึ่ง UpdateRate ของลูกบอลจะเป็นตัวปรับไว ผลลัพธ์ของโปรแกรม จะได้ลูกบอลที่มี Animation เด้งไปมา

4.ให้เปลี่ยนการทำงานของ Clock Animation จากที่ใช้ Timer ให้เปลี่ยนไปใช้ Thread จากคลาส Thread แทน