Übungen – Threads (I)

1.

a.) Schreiben sie eine Konsolenanwendung, die sekündlich einen Punkt '.' auf der Konsole ausgibt und parallel dazu Eingaben von der Konsole einliest und wieder ausgibt. Probieren sie beide Möglichkeiten zum Starten eines Thread (eigene Thread-Klasse bzw. Runnable-Implementierung) aus.

b.) Sobald ein ‚q' eingegeben wird, soll sich die Anwendung (also auch der Punkt-Ausgabe-Thread) beenden. Welche Ideen haben sie?

2.

Es soll ein grafisches Ziehungsgerät der (Zusatz-) Lotterie „Spiel 77" simuliert werden:

- **Nach Betätigen des Start-Buttons „rotieren" alle 7 Zahlen im Wertebereich 0-9, wobei jede Zahl initial mit einer Zufallszahl beginnt. Die „rotierenden" Zahlen werden rot dargestellt. Der Start-Button wird gesperrt, der Beenden-Button aktiv.**

- **Beim Beenden-Button wird im Zeitabstand von 1,5s die Rotation je einer Zahl beendet. Die jeweilige Gewinnzahl wird in blau dargestellt.**
- **Realisieren sie Ziehung einer Zahl mit jeweils einem eigenen Thread (also insgesamt 7). Zwischen der Weiterschaltung einer Zahl soll eine zufällige Zeitspanne von 20-100ms „gewartet" werden.**
- **Der Start-Button wird erst wieder aktiv, wenn alle 7 Zahlen feststehen, d.h. alle Threads beendet sind.**
- **Bei Betätigen des Close-Buttons soll eine laufende Ziehung sauber abgebrochen werden, so dass sich alle Threads sofort vorzeitig beenden.**

Verwenden sie die Vorlagedatei „Spiel77.java" und ergänzen diese um obige Anforderungen.

3.

Schreiben sie eine kleine AWT-Applikation mit einem Frame-Objekt in dem ein Kreis sekündlich abwechselnd blau bzw. gelb gefüllt „blinkt". Starten sie dazu einen eigenen Thread.

Zur Erinnerung: Neuzeichnen eines Fensterbereichs kann durch aufruf der Methode „repaint" erreicht werden. Zum Zeichnen von 2D-Grafik muss die Methode paint in der eigenen Klasse überladen werden. Die Zeichenmethoden stellt das übergebene Graphics-Objekt bereit.

4.

Schreiben sie eine AWT-Applikation mit einem Frame-Objekt, in dem ein „Lauftext" permanent „durchläuft". Hilfreich kann das folgende Codefragment sein, das demonstriert, wie ein spezieller Font erzeugt, in einem Frame gesetzt und dessen Maße abgefragt werden kann. Lesen sie ggf. in der Java-Doku nach.


```
String text = " ... Java Thread ...";
int fs = 28;
Font lFont = new Font("Verdana", Font.BOLD, fs);
setFont(lFont);
FontMetrics fm = getFontMetrics(lFont);
lt = fm.stringWidth(text);
```

5.
Beurteilen sie nachfolgende Varianten der Synchronisation (a.-f.). Welche Varianten führen zu einer ununterbrochenen Ausführung der Methode `write`?

```
public class Letters extends Thread {

    private String name;

    public Letters(String name) { this.name = name; }

    public void write() {
        System.out.print(name);
        try{ Thread.sleep(100); } catch (InterruptedException ie) {}
        System.out.print(name);
    }

    public static void main(String[] args) {
        new Letters("X").start();
        new Letters("Y").start();
    }
/*
    // a.)
    public void run() {
        write();
    }

    // b.)
    public synchronized void run() {
        write();
    }

    // c.)
    public static synchronized void run() {
        write();
    }

    // d.)
    public void run() {
        synchronized(this) {
            write();
        }
    }

    // e.)
    public void run() {
        synchronized(System.out) {
            write();
        }
    }

    // f.)
    public void run() {
        synchronized(Letters.class) {
            write();
        }
    }
*/
}
```