

## Sehr einfaches Programm, welches von Thread erbt

### Aufgabe 1

Erstelle ein Programm `MyThread`, welches einen Thread startet und einen Text auf die Konsole schreibt. (Erbe entweder von `Thread` oder implementiere `Runnable`)

Falls du Probleme hast, hier noch ein paar Tipps:

- In Java erbst du in dem du schreibst; *Unterklasse **extends** Superklasse*
- Um mit `Runnable` zu arbeiten verwende *`MyThread implements Runnable`*
- Du kannst in die Konsole schreiben indem du die Methode `System.out.println(„Hier schreibst du deinen Text“)`; aufrufst. In Eclipse gibt es einen praktischen Trick. Markiere deinen Text. Drücke `ctrl+space` und dann Pfeil nach oben und Enter.
- Kreiere einen neuen Thread indem du das keyword `new` tippst und dann `Ctrl+Space` drückst. Eclipse schlägt dir dann vor `create new object`. So erhältst du gerade eine Vorlage in welcher du mit `Tab` ins nächste Feld gelangst. Du verwendest hier einen Konstruktor, der keine Argumente braucht.
- Starte den Thread mit `mythread.start()`; Dabei passiert folgendes: `start()` bewirkt, dass der Thread ausgeführt wird. Dabei ruft die Java Virtual Machine die Methode `run()` dieses Threads auf.
- Implementiere die Methode `run`. Das Programm sollte einfach irgendeinen Text in die Konsole schreiben.

Für mehr Informationen zur Klasse `Thread` bzw. dem Interface `Runnable` kannst du hier nachschauen:

<http://java.sun.com/j2se/1.5.0/docs/api/>

### Aufgabe 2

Experimentiere ein wenig mit deinem Programm. Was passiert, wenn du mehrere Threads startest, erhältst du jedes Mal den selben Output,...

Tipps:

- Du kannst im `run()` den Namen des aktuellen Threads mit der Methode `getName()` anzeigen lassen. So kannst du beispielsweise überprüfen, ob dein Programm immer den selben Output generiert.
- Falls du die Methode `toString()` aufrufst, erhältst du nebst dem Namen auch noch Informationen zur Priorität und der Thread-Gruppe.

Auf der nächsten Seite findest du meine Lösung.

## Lösung

### Aufgabe 1

```
public class MyThread extends Thread{

    public void run() {
        System.out.println("Mein erstes Programm mit Threads
funktioniert.");
    }

    public static void main(String[] args) {
        MyThread t = new MyThread();
        t.start();
        System.out.println("start");
    }
}
```

Bemerkung:

Die 2 Zeilen

*MyThread t = new MyThread();*

*t.start();*

können auch durch folgende Zeile ersetzt werden:

*(new MyThread()).start();*

oder

```
public class MyThread implements Runnable {
    public void run() {
        System.out.println("Mein erstes Programm mit Threads
funktioniert.");
    }
    public static void main(String argFile[]) {
        (new Thread(new MyThread())).start();
        System.out.println("start");
    }
}
```

Output (egal welche Variante):

```
start
Mein erstes Programm mit Threads funktioniert.
```

## Aufgabe 2

Nur ein Beispiel.

```
public class MyThread extends Thread{

    public void run() {
        System.out.println("Ich bin der Thread: " +
getName());
    }

    public static void main(String[] args) {
        for (int i = 0; i < 6; i++) {
            MyThread t = new MyThread();
            t.start();
            System.out.println("Starte Thread Nr: " + i);
        }
    }
}
```

Output:

Starte Thread Nr: 0 Ich bin der Thread: Thread-0 Starte Thread Nr: 1 Starte Thread Nr: 2 Ich bin der Thread: Thread-2 Starte Thread Nr: 3 Starte Thread Nr: 4 Ich bin der Thread: Thread-1 Ich bin der Thread: Thread-4 Starte Thread Nr: 5 Ich bin der Thread: Thread-3 Ich bin der Thread: Thread-5	Starte Thread Nr: 0 Starte Thread Nr: 1 Starte Thread Nr: 2 Starte Thread Nr: 3 Starte Thread Nr: 4 Starte Thread Nr: 5 Ich bin der Thread: Thread-1 Ich bin der Thread: Thread-3 Ich bin der Thread: Thread-5 Ich bin der Thread: Thread-0 Ich bin der Thread: Thread-2 Ich bin der Thread: Thread-4
Starte Thread Nr: 0 Starte Thread Nr: 1 Ich bin der Thread: Thread-0 Starte Thread Nr: 2 Starte Thread Nr: 3 Starte Thread Nr: 4 Starte Thread Nr: 5 Ich bin der Thread: Thread-2 Ich bin der Thread: Thread-4 Ich bin der Thread: Thread-1 Ich bin der Thread: Thread-3 Ich bin der Thread: Thread-5	Starte Thread Nr: 0 Ich bin der Thread: Thread-0 Starte Thread Nr: 1 Starte Thread Nr: 2 Ich bin der Thread: Thread-2 Starte Thread Nr: 3 Starte Thread Nr: 4 Ich bin der Thread: Thread-4 Starte Thread Nr: 5 Ich bin der Thread: Thread-1 Ich bin der Thread: Thread-3 Ich bin der Thread: Thread-5

Das ganze läuft also ziemlich unkoordiniert ab. Es gibt aber viele Möglichkeiten etwas Ordnung in das Chaos zu bringen.