

Úkol 1

- napište MergeSort
 - např. nad polem `intů`
- napište paralelní MergeSort
 - použijte vlákna
 - počet vláken bude určený jako parametr programu
 - nebo použijte `Runtime.availableProcessor()`
 - implementujte přímo pomocí vláken i pomocí `ForkJoinPool`

Úkol 2

- napište třídu RoundBuffer
 - implementuje kruhový buffer dané kapacity (určeno při vytvoření instance), do kterého se dají ukládat Stringy
 - podporuje alespoň metody
 - void put(String msg)
 - String get()
 - buffer může paralelně používat více vláken
 - pokud už není v bufferu místo příp. není co vybrat, tak se put() příp. get() zablokují
- napište program, kde více vláken zapisuje a čte z bufferu
 - vlákna mezi zápisy/čtením vždy usnou na náhodnou dobu (náhodná čísla viz java.util.Random)

Testy...

Test 1

- Co program vypíše

```
public class Test01 {
    public static synchronized void main(String[] a) {
        Thread t = new Thread() {
            public void run() {
                pong();
            }
        };
        t.start();
        System.out.println("Ping");
    }
    static synchronized void pong() {
        System.out.println("Pong");
    }
}
```

- A Ping Pong
- B Pong Ping
- C nic – nastane deadlock a program neskončí
- D vyhodí výjimku
- E pořadí Ping a Pong se může lišit běh od běhu programu

Test 2

- Co program vypíše

```
public class SelfInterruption {
    public static void main(String[] args) {

        Thread.currentThread().interrupt();

        if (Thread.interrupted()) {
            System.out.println("Interrupted: " +
                               Thread.interrupted());
        } else {
            System.out.println("Not interrupted: " +
                               Thread.interrupted());
        }
    }
}
```

- A Interrupted: true
- B Not interrupted: false
- C něco jiného



Verze prezentace PJ08.cz.2018.01

Tato prezentace podléhá licenci [Creative Commons Uveďte autora-Neužívejte komerčně 4.0 Mezinárodní License](https://creativecommons.org/licenses/by-nc/4.0/).