Bereich: Threads (2)

Synchronisation	**Musterlösung**
Package: `de.dhbwka.java.exercise.threads.buffer`	**Klasse:** `MyBuffer`

```java
package de.dhbwka.java.exercise.threads.buffer;

import java.util.LinkedList;
import java.util.List;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016-2018 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.1
 */
public class MyBuffer {

   private final static int MAXSIZE = 3;
   private List<Integer> values = new LinkedList<>();

   /**
    * The producer routine
    *
    * @param value
    *           value to put in buffer
    */
   public synchronized void put( int value ) {
      if ( !(this.values.size() < MyBuffer.MAXSIZE) ) {
         System.out.println( "Buffer full - wait!" );
         try {
            this.wait();
         } catch ( InterruptedException e ) {
         }
      } // buffer is empty, fill it!
      this.values.add( new Integer( value ) );
      // notify waiting consumer via notify()
      this.notify();
      System.out.println( "Put: " + value );
   }


   // Continued on next page
```

```
   /**
    * The consumer routine
    *
    * @return value from buffer
    */
   public synchronized int get() {
      // consumer must wait until buffer is not empty
      if ( this.values.size() == 0 ) {
         try {
            System.out.println( "Buffer empty - wait!" );
            this.wait();
         } catch ( InterruptedException e ) {
         }
      }
      // buffer has at least one Element, remove it!
      int value = this.values.remove( 0 );
      // notify waiting producer via notify()
      this.notify();
      System.out.println( "Get: " + value );
      return value;
   }

}
```

```
package de.dhbwka.java.exercise.threads.buffer;

import java.util.Random;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016-2018 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.1
 */
public class BufferDemo {

   private static final int TESTDATASIZE = 7;
   private static final int MAXDELAY = 500;
   private static final Random RND = new Random();

   /**
    * Inner class for Producer-Thread
    */
   static class ProducerThread implements Runnable {
      private MyBuffer buffer;

      public ProducerThread( MyBuffer b ) {
         this.buffer = b;
      }

      // Continued on next page
```

```
      @Override
      public void run() {
         for ( int i = 0; i < BufferDemo.TESTDATASIZE; i++ ) {
            this.buffer.put( i );
            // simulate data calculation time by random waiting
            try {
               Thread.sleep( BufferDemo.RND.nextInt( BufferDemo.MAXDELAY ) );
            } catch ( InterruptedException ex ) {
            }
         }
      }
   }

   /**
    * Inner class for Consumer-Thread
    */
   static class ConsumerThread implements Runnable {
      private MyBuffer buffer;

      public ConsumerThread( MyBuffer b ) {
         this.buffer = b;
      }

      @Override
      public void run() {
         for ( int i = 0; i < BufferDemo.TESTDATASIZE; i++ ) {
            this.buffer.get();
            // Simulate data processing time by random waiting
            try {
               Thread.sleep( BufferDemo.RND.nextInt( BufferDemo.MAXDELAY ) );
            } catch ( InterruptedException ex ) {
            }
         }
      }
   }

   public static void main( String args[] ) {
      MyBuffer buf = new MyBuffer();
      // Create one producer and one consumer
      new Thread( new ProducerThread( buf ) ).start();
      new Thread( new ConsumerThread( buf ) ).start();
   }

}
```

Bereich: Threads (2)

Suchmaschine	**Musterlösung**
Package: `de.dhbwka.java.exercise.threads.search`	**Klasse:** `SearchEngine`

```
package de.dhbwka.java.exercise.threads.search;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016-2018 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.1
 */
public class PageLoader implements Runnable {

   private final static String ENCODING = "UTF-8"; // e.g. "ISO-8859-1", "UTF-8"
   public String url;
   private String content;

   public PageLoader( String url ) {
      this.url = url;
      this.content = null;
   }

   /**
    * Check if page was loaded
    *
    * @return <code>true</code> if content has been loaded (is not
    *         <code>null</code>), <code>false</code> otherwise
    */
   public boolean pageLoaded() {
      return this.content != null;
   }

   /**
    * Get the page content, but first test if there is any with
    * {@link #pageLoaded()}
    *
    * @return page content in a single line or <code>null</code> if page was no
    *         {@link #pageLoaded() loaded}
    */
   public String getPageContent() {
      return this.content != null
            ? this.content.replaceAll( "(\\r|\\n)+", "##" ) : null;
   }

   @Override
   public void run() {
      this.content =
         ReadURLExample.getStringContentFromUrl( this.url, PageLoader.ENCODING );
   }
}
```

```java
package de.dhbwka.java.exercise.threads.search;

import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

/**
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg Cooperative
 * State University.
 *
 * (C) 2016-2019 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 *
 * @author DHBW lecturer
 * @version 1.2
 */
public class SearchEngine {

    public static final int MAXTHREADS = 3;

    private List<PageLoader> pageLoader = new ArrayList<>();

    public void crawl(String... urls) {

        ExecutorService exec = Executors.newFixedThreadPool(MAXTHREADS);

        for (String url : urls) {
            PageLoader pl = new PageLoader(url);
            this.pageLoader.add(pl);
            exec.submit(pl);
            System.out.println("Gestartet: " + url);
        }

        // Keine weiteren Threads mehr zulassen
        exec.shutdown();

        while (!pageLoader.isEmpty()) {
            for (PageLoader pl : pageLoader) {
                if (pl.pageLoaded()) {
                    System.out.println("Loaded: " + pl.url);
                    System.out.println("  Content: " +
                        pl.getPageContent().substring(0, 40));
                    this.pageLoader.remove(pl);
                    break;
                }
            }
        }
    }

    public static void main(String[] args) {
        new SearchEngine().crawl("https://www.tagesschau.de",
                                 "https://www.sueddeutsche.de",
                                 "https://www.spiegel.de",
                                 "https://www.kit.edu");
    }
}
```