

```

package modem.model;

import java.util.Observable;

//Ich bestaetige, dass ich diese Pruefung selbstaendig geloest habe.
//Ich weiss, dass bei Zuwiederhandlung die Note 1 erteilt wird.
//
//Name:
//Vorname:

public class Model extends Observable implements SymbolListener {
    // 33
    public Delay delay;
    private FIRFilter transmitFilter;
    private FIRFilter kanalFilter;
    private LMSFilter fffilter;
    private LMSFilter fbFilter;

    protected SymbolQuelle symbolSource = new SymbolQuelle(16 * 1024, this);
    private double[] fAxis = new double[1024];
    private double[] tAxis = new double[512];

    private Random random = new Random();

    private double schrittFF = 1e-5, schrittFB = 1e-4, fbOut = 0.0;
    private double tonFrequenz = 5e6;
    private double fb = 50e6, fs = 2 * fb, Ts = 1 / fs;
    private long n = 0;

    private double awgnAmplitude = 0.0;
    private double tonAmplitude = 0.0;

    /**
     * <pre>
     * - Erzeugt mit den Argumenten (16 * 1024, this) die SymbolQuelle.
     * - Erzeugt die Verzögerungsleitung delay mit einer Verzögerung von 32.
     * - Erzeugt das Transmit-Filter und das Kanal-Filter mit den Koeffizienten
     *   TxFlt.TxFltBCoeffs und Kanal.BCoeffs[0].
     * - Erzeugt die adaptiven LMS-Filter fffilter und fbFilter mit den Längen 64 und 16.
     * - Erzeugt mittels linspace() von Matlab die fAxis und die tAxis entsprechender
     *   Längen und den Bereichen von Null bis fs/2 resp. von 0 bis Länge der tAxis minus 1.
     * </pre>
     */
    public Model() {

        public double[] getfAxis() {

        public double[] getImpulsFBFilter() {

        public double[] getImpulsFFFfilter() {

        public double[] getImpulsKanalFilter() {

        public double[] getImpulsTransmitFilter() {

        public double[] gettAxis() {

        public void processSymbol(double[] symbol) {

        /**
         * <pre>
         * - Erzeugt ein neues fffilter und fbFilter ursprünglicher Länge
         * </pre>
         */
        public void resetFilter() {

        /**
         * <pre>

```

Model.java

```
* - Erzeugt ein neues Kanal-Filter mit entsprechenden Kanal.BCoeffs[Index] Koeffizienten.  
* - Setzt die Attribute entsprechend den Werten im Objekt parameter.  
* </pre>  
*  
* @param parameter  
*/  
public void setParameter(Parameter parameter) {  
public void notifyObservers() {  
}
```