

```

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package seafarm;

//import java.nio.ByteBuffer;
import seafarm.RovDatahandler;
import java.util.TimerTask;

/**
 * runs on timertask. checs the functions each run
 *
 * @author Jørgen
 */
public class RovGUIController extends TimerTask {

    private RovDatahandler datahandler;

    public RovGUIController() {

    }

    /**
     * prints the statement :Reques data:"
     */
    @Override
    public void run() {
        System.out.println("Request data");
        this.datahandler.incrementRequestCode();
    }

    /**
     * setting the datahandler to the timertask
     *
     * @param datahandler
     */
    public void setDatahandler(RovDatahandler datahandler) {
        this.datahandler = datahandler;
    }

    /**

```

```

    * checs value if true, and ends the boolean value to datahandler
    *
    * @param value
    */
public void setFwd(boolean value) {

    if (value) {
        System.out.println("Command : Fwd:, Aktivated thrusters");
        this.datahandler.goFwd();

    } else {
        System.out.println("Command : Released FWD, deAktivated
thrusters");
        this.datahandler.releaseGoFwd();

    }

}

/**
 * checs value if true, and ends the boolean value to datahandler
 *
 * @param value
 */
public void setLeft(boolean value) {
    if (value) {
        System.out.println("Command : LEFT, Aktivated thrusters");
        this.datahandler.goLeft();
    } else {
        System.out.println("Command : Released LEFT, deAktivated
thrusters");
        this.datahandler.releaseGoLeft();
    }
}

/**
 * checs value if true, and ends the boolean value to datahandler
 *
 * @param value
 */
public void setRev(boolean value) {
    if (value) {
        System.out.println("Command : Rev, Aktivated thrusters");
        this.datahandler.goRew();
    }
}

```

```

        } else {
            System.out.println("Command : released Rev, deAktivated
thrusters");
            this.datahandler.releaseGoRew();
        }
    }

/**
 * checs value if true, and ends the boolean value to datahandler
 *
 * @param value
 */
public void setRight(boolean value) {
    if (value) {
        System.out.println("Command : RIGHT, Aktivated thrusters");
        this.datahandler.goRight();
    } else {
        System.out.println("Command : released RIGHT, deAktivated
thrusters");
        this.datahandler.releaseGoRight();
    }
}

/**
 * checs value if true, and ends the boolean value to datahandler
 *
 * @param value
 */
public void setSlideLeft(boolean value) {
    if (value) {
        System.out.println("Command : SLIDE LEFT, Aktivated
thrusters");
        this.datahandler.setSlideLeft();
    } else {
        System.out.println("Command :released SLIDE LEFT, deAktivated
thrusters");
        this.datahandler.resetSlideLeft();
    }
}

/**
 * checs value if true, and ends the boolean value to datahandler
 *

```

```

    * @param value
    */
    public void setSlideRight(boolean value) {
        if (value) {
            System.out.println("Command : SLIDE RIGHT, Activated
thrusters");
            this.datahandler.setSlideRight();
        } else {
            System.out.println("Command : releasae SLIDE RIGHT, deActivated
thrusters");
            this.datahandler.resetSlideRight();
        }
    }

/**
 * checs value if true, and ends the boolean value to datahandler
 *
 * @param value
 */
    public void setStart(boolean value) {
        if (value) {
            System.out.println("Command : STARTING ROV");
            this.datahandler.enableROV();
        } else if (!value) {
            System.out.println("Command : STOP ROV");
            this.datahandler.disableROV();
        }
    }

/**
 * checs value if true, and ends the boolean value to datahandler
 *
 * @param value
 */
    public void setLightValue(int sens) {
        System.out.println("New lightValue = " + sens);
        this.datahandler.setLightValue((byte) sens);
    }

/**
 * checs value if true, and ends the boolean value to datahandler
 *
 * @param value
 */

```

```
public void setThrusterValue(int sens) {  
    System.out.println("New thrusterValue = " + sens);  
    this.datahandler.setThrusterValue((byte) sens);  
}  
  
}
```