

```

/*
 * 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.math.BigInteger;
import java.util.Arrays;
import java.util.Observable;
import java.util.Observer;

/**
 * the observer to check the data received from Raspberry
 *
 * @author Jørgen
 */
public class RovReceiveDataObservable extends Observable {

    float temperature=0;
    float descTemp=0;
    float pressure=0;
    float descPressure=0;
    float depth=0;
    float descDepth=0;
    float oxygen=0;
    float descOxygen=0;
    int waterlevel=0;
    float roll=0;
    float descRoll;
    float pitch;
    float descPitch;
    int heading=0;
    float tempInROV=0;
    float descTempInROV;
    float xAccMeter;
    float descxAccmeter;
    float yAccMeter;
    float descYAccmeter;
    float zAccMeter;
    float desczAccmeter;
    float tensiometer;

```

```

float descTensiometer;

//private final Datahandler datahandler;
public RovReceiveDataObservable() {

}

/**
 * adds the observer to the object
 *
 * @param o
 */
@Override
public synchronized void addObserver(Observer o) {
    super.addObserver(o); //To change body of generated methods, choose
Tools | Templates.
}

/**
 * Sets the receiving data to the fields, and notifies observers.
 *
 * @param data
 */
public void setData(byte[] data) {
    // check if the array is of the same length and the requestcode has
changed
    if (data.length == 25) {

        descTemp = data[2];
        temperature = (data[1] + (descTemp / 100));
        descPressure = data[4];
        pressure = (data[3] + (descPressure / 100));
        descDepth = data[6];
        depth = (data[5] + (descDepth / 100));
        descOxygen = data[8];
        oxygen = (data[7] + (descOxygen / 100));
        waterlevel = data[9];
        descRoll = data[11];
        roll = (data[10] + (descRoll / 100));
        descPitch = data[13];
        pitch = (data[12] + (descPitch / 100));
        heading = data[14];
        descTempInROV = data[16];
        tempInROV = (data[15] + (descTempInROV / 100));
    }
}

```

```

        descxAccmeter = data[18];
        xAccMeter = (data[17] + (descxAccmeter / 100));
        descYAccmeter = data[20];
        yAccMeter = (data[19] + (descYAccmeter / 100));
        desczAccmeter = data[22];
        zAccMeter = (data[21] + (desczAccmeter / 100));
        descTensiometer = data[24];
        tensiometer = (data[23] + (descTensiometer / 100));

        setChanged();
        notifyObservers();
    }
}

// Gett all values of the sensores
public float getDepth() {
    return depth;
}

public float getPressure() {
    return pressure;
}

public float getTemperature() {
    return temperature;
}

public float getOxygen() {
    return oxygen;
}

public int getWaterlevel() {
    return waterlevel;
}

public float getRoll() {
    return roll;
}

public float getPitch() {
    return pitch;
}

public int getHeading() {

```

```
        return heading;
    }

    public float getTempInROV() {
        return tempInROV;
    }

    public float getXAccMeter() {
        return xAccMeter;
    }

    public float getYAccMeter() {
        return yAccMeter;
    }

    public float getzAccMeter() {
        return zAccMeter;
    }

    public float getTensiometer() {
        return tensiometer;
    }

    public boolean shouldChildOfThisRun() {
        //return datahandler.shouldThreadRun();
        return true;
    }
}
```