

```

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 platform
 */
public class RecieveDataObserver extends Observable {

    //data to recieve from platform///
    float longitude=0;
    float latitude=0;
    float currentSpeed=0;
    float pitch=0;
    float yaw=0;
    float roll=0;
    boolean enabled=false;
    boolean dpMode = false;
    boolean autoMode = false;
    boolean manualMode = false;
    //winch variables//
    boolean rovLocked=false;
    boolean rovUpperPos=false;

    public RecieveDataObserver() {

    }

    /**
     * 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.

```
    }

    /*functions to update the platform variables*/

    public void setCurrentSpeed(float speed){
        this.currentSpeed=speed;
    }

    public void setEnabled(boolean enabl){
        this.enabled=enabl;
    }

    public void setCurrentPitch(float ptc){
        this.pitch=ptc;
    }

    public void setCurrentLongitude(float lng){
        this.longitude=lng;
    }

    public void setCurrentLatitude(float lat){
        this.latitude=lat;
    }
    public void setCurrentYaw(float yaw){
        this.yaw=yaw;
    }

    }
    public void setCurrentRoll(float roll){
        this.roll=roll;
    }

    }

    public void setRovLock(boolean lockOn){
        this.rovLocked=lockOn;
    }
}

    public void setRovUpperPos(boolean isInupper){
        this.rovUpperPos=isInupper;
    }

    }

    ///Get the current platform variables//
```

```
public float getCurrentSpeed(){
    return this.currentSpeed;
}

public boolean getEnabled(){
    return this.enabled;
}

public float gettCurrentPitch(){
    return this.pitch;
}

public float getCurrentLongitude(){
    return this.longitude;
}

public float getCurrentLatitude(){
    return this.latitude;
}

public float getCurrentYaw(){
    return this.yaw;
}

public float getCurrentRoll(){
    return this.roll;
}

public boolean getRovInupperPos(){
    return this.rovUpperPos;
}

public boolean getRovIsLocked(){
    return this.rovLocked;
}

public boolean getDpMode() {
    return dpMode;
}

public void setDpMode(boolean dpMode) {
    this.dpMode = dpMode;
}
```

```
public boolean getAutoMode() {
    return autoMode;
}

public void setAutoMode(boolean autoMode) {
    this.autoMode = autoMode;
}

public boolean getManualMode() {
    return manualMode;
}

public void setManualMode(boolean manualMode) {
    this.manualMode = manualMode;
}

//notify the observer that the data has changed///
public void notifyObs() {
    setChanged();
    notifyObservers();
}

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