```
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 SendDataObserver extends Observable {

       //data to send to platform///

    //platform movement commands//
     int thrusterSpeed=0;
     boolean fwdMotion = false;
     boolean bckMotion = false;
     boolean rightMotion = false;
     boolean leftMotion = false;
     boolean clockWMotion = false;


     boolean counterClockWMotion = false;
     boolean enableLight=false;
     boolean enableFlute=false;
     boolean platformEnable=false;
     boolean enableAuto=false;
     boolean enableManual=false;


     //DpMode commands//
     boolean dpModeEnable=false;
     float latitude=0;
     float longitude=0;



     //winch commands//
     boolean winchUp=false;
     boolean winchDown=false;
     int winchSpeed=0;
```

```
     boolean winchLockOn=false;
     boolean winchLockOff=false;

     //Pump commands//
     boolean startPump = false;
     //rov data for logging
     float dpt=0;
     float tmpInrov=0;
     float tmpInSea=0;
     float headrov=0;
     float oxygen=0;









    public SendDataObserver() {



    }

    /**
     * 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.
    }

    //set functions for variables that the observer hodlds;
    public  void setThrusterSpeed(int thrusterSpeed) {
         this.thrusterSpeed = thrusterSpeed;

    }
```

```
public void setEnableLight(boolean enableLight) {
    this.enableLight = enableLight;
}

public void setEnableFlute(boolean enableFlute) {
    this.enableFlute = enableFlute;
}

public void setPlatformEnable(boolean platformEnable) {
    this.platformEnable = platformEnable;
}

public void setEnableAuto(boolean enableAuto) {
    this.enableAuto = enableAuto;
}

public void setEnableManual(boolean enableManual) {
    this.enableManual = enableManual;
}

public void setDpModeEnable(boolean dpModeEnable) {
    this.dpModeEnable = dpModeEnable;
}

public void setLatitude(float latitude) {
    this.latitude = latitude;
}

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

public void setWinchUp(boolean winchUp) {
    this.winchUp = winchUp;
}

public void setWinchDown(boolean winchDown) {
    this.winchDown = winchDown;
}

public void setWinchSpeed(int winchSpeed) {
    this.winchSpeed = winchSpeed;
}
```

```
public void setWinchLockOn(boolean winchLockOn) {
    this.winchLockOn = winchLockOn;
}


public void setWinchLockOff(boolean winchLockOff) {
    this.winchLockOff = winchLockOff;
}


///functions to get the observer variables///

public int getThrusterSpeed() {
    return thrusterSpeed;
}


public boolean isEnableLight() {
    return enableLight;
}

public boolean isEnableFlute() {
    return enableFlute;
}

public boolean isPlatformEnable() {
    return platformEnable;
}

public boolean isEnableAuto() {
    return enableAuto;
}

public boolean isEnableManual() {
    return enableManual;
}

public boolean isDpModeEnable() {
    return dpModeEnable;
}

public float getLatitude() {
    return latitude;
}
```

```
public float getLongitude() {
    return longitude;
}

public boolean isWinchUp() {
    return winchUp;
}

public boolean isWinchDown() {
    return winchDown;
}

public int getWinchSpeed() {
    return winchSpeed;
}

public boolean isWinchLockOn() {
    return winchLockOn;
}

public boolean isWinchLockOff() {
    return winchLockOff;
}

    public boolean isFwdMotion() {
    return fwdMotion;
}

public void setFwdMotion(boolean fwdMotion) {
    this.fwdMotion = fwdMotion;
}

public boolean isBckMotion() {
    return bckMotion;
}

public void setBckMotion(boolean bckMotion) {
    this.bckMotion = bckMotion;
}

public boolean isRightMotion() {
    return rightMotion;
}
```

```
public void setRightMotion(boolean rightMotion) {
    this.rightMotion = rightMotion;
}

public boolean isLeftMotion() {
    return leftMotion;
}

public void setLeftMotion(boolean leftMotion) {
    this.leftMotion = leftMotion;
}

public boolean isClockWMotion() {
    return clockWMotion;
}

public void setClockWMotion(boolean clockWMotion) {
    this.clockWMotion = clockWMotion;
}

public boolean isCounterClockWMotion() {
    return counterClockWMotion;
}

public void setCounterClockWMotion(boolean counterClockWMotion) {
    this.counterClockWMotion = counterClockWMotion;
}

public boolean isStartPump() {
    return startPump;
}

public void setStartPump(boolean startPump) {
    this.startPump = startPump;
}

//rovdata

 public float getDpt() {
    return dpt;
}

public void setDpt(float dpt) {
```

```
    this.dpt = dpt;
}

public float getTmpInrov() {
    return tmpInrov;
}

public void setTmpInrov(float tmpInrov) {
    this.tmpInrov = tmpInrov;
}

public float getTmpInSea() {
    return tmpInSea;
}

public void setTmpInSea(float tmpInSea) {
    this.tmpInSea = tmpInSea;
}

public float getHeadrov() {
    return headrov;
}

public void setHeadrov(float headrov) {
    this.headrov = headrov;
}

public float getOxygen() {
    return oxygen;
}


public void setOxygen(float oxygen) {
    this.oxygen = oxygen;
}



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

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


}
```