

● より良い暮らしを提供する ECC ホーム！（Before）

「より良い暮らしを提供する」ECC ホームでは、様々な電化製品をボタンで操作するリモコンを開発している。とりあえずエアコンと加湿器と除湿器を操作できるように業者に作成してもらったところ（リスト1）、特定のエアコンと特定の加湿器と特定の除湿器しか操作できないシロモノになってしまった！様々な電化製品を操作するためにはリモコンを改造しなければならない！！操作できる製品を追加・変更したり、ボタンの設定をカスタマイズしたりできるように設計を見直せ。

業者の作ったリモコン（Controller クラス）の仕様

ボタン	操作	操作する製品
0	冷房	TypeA エアコン (AirconTypeA クラス)
1	暖房	
2	加湿器オン	TypeA 加湿器 (HumidifierTypeA クラス)
3	加湿器オフ	
4	除湿器オン	TypeA 除湿器 (DehumidifierTypeA クラス)
5	除湿器オフ	

TypeA エアコン、TypeA 加湿器、TypeA 除湿器  
にしか対応していない。

リスト1：業者の作ったリモコン（ファイル「before.Controller.java」）

```
class Controller {
    private AirconTypeA ac = null; // TypeA エアコン
    private HumidifierTypeA hmd = null; // TypeA 加湿器
    private DehumidifierTypeA dehmd = null; // TypeA 除湿器

    public void setAircon(AirconTypeA ac) { this.ac = ac; }
    public void setHumidifier(HumidifierTypeA hmd) { this.hmd = hmd; }
    public void setDehumidifier(DehumidifierTypeA dehmd) { this.dehmd = dehmd; }
    public void showCommand() {
        System.out.println("[0]冷房");
        System.out.println("[1]暖房");
        System.out.println("[2]加湿 ON");
        System.out.println("[3]加湿 OFF");
        System.out.println("[4]除湿 ON");
        System.out.println("[5]除湿 OFF");
    }
    public void pushButton(int i) {
        switch(i) {
            case 0: ac.setCool(); break; // 冷房
            case 1: ac.setWarm(); break; // 暖房
            case 2: hmd.hmdOn(); break; // 加湿 ON
            case 3: hmd.hmdOff(); break; // 加湿 OFF
            case 4: dehmd.dehmdOn(); break; // 除湿 ON
            case 5: dehmd.dehmdOff(); break; // 除湿 OFF
        }
    }
}
```

## リスト2：リモコンの操作（ファイル「before.Main.java」）

```
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        Controller rc = new Controller();           // リモコンの生成
        rc.setAircon(new AirconTypeA());           // TypeA エアコンをセット
        rc.setHumidifier(new HumidifierTypeA());    // TypeA 加湿器をセット
        rc.setDehumidifier(new DehumidifierTypeA()); // TypeA 除湿器をセット

        while(true) {
            rc.showCommand();                       // 操作コマンドの表示
            System.out.print("どのボタンを押しますか？>");
            int n = Integer.parseInt(in.next());
            if (n < 0) break;
            rc.pushButton(n);                       // 指定したコマンドの実行
        }
    }
}
```

## 実行時の画面

[0]冷房  
[1]暖房  
[2]加湿 ON  
[3]加湿 OFF  
[4]除湿 ON  
[5]除湿 OFF  
どのボタンを押しますか？>0  
【エアコンA】冷房にしました！

[0]冷房  
[1]暖房  
[2]加湿 ON  
[3]加湿 OFF  
[4]除湿 ON  
[5]除湿 OFF  
どのボタンを押しますか？>1  
【エアコンA】暖房にしました！

[0]冷房  
[1]暖房  
[2]加湿 ON  
[3]加湿 OFF  
[4]除湿 ON  
[5]除湿 OFF  
どのボタンを押しますか？>2  
【加湿器A】ONにしました！

## (続き)

[0]冷房  
[1]暖房  
[2]加湿 ON  
[3]加湿 OFF  
[4]除湿 ON  
[5]除湿 OFF  
どのボタンを押しますか？>3  
【加湿器A】OFFにしました！

[0]冷房  
[1]暖房  
[2]加湿 ON  
[3]加湿 OFF  
[4]除湿 ON  
[5]除湿 OFF  
どのボタンを押しますか？>4  
【除湿器A】ONにしました！

[0]冷房  
[1]暖房  
[2]加湿 ON  
[3]加湿 OFF  
[4]除湿 ON  
[5]除湿 OFF  
どのボタンを押しますか？>5  
【除湿器A】OFFにしました！

## ● J2Kad26D 「TypeB エアコンの操作 (Aircon インターフェイス)」

ECC ホームでは快適でエコな冷房・暖房を行う TypeB エアコン (AirconTypeB クラス) も取り扱っている。(加湿器と除湿器は横に置いて) とりあえず TypeA と TypeB の両方に対応するように修正せよ。

## リスト1: TypeB エアコンの操作 (ファイル「pac26d.Main.java」)

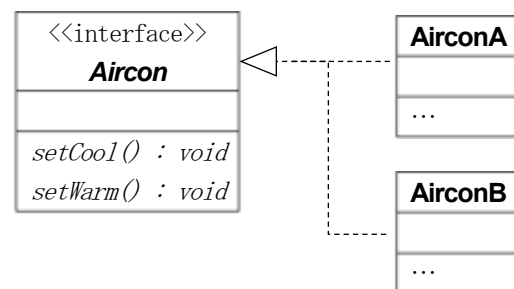
```
public class Main {  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
        Controller rc = new Controller();  
        rc.setAircon(new AirconTypeA());  
  
        while(true) {  
            rc.showCommand();  
            System.out.print("どのボタンを押しますか? >");  
            int cmd = Integer.parseInt(in.next());  
            if (cmd < 0) break;  
            rc.pushButton(cmd);  
            System.out.println();  
        }  
    }  
}
```

AirconTypeA を AirconTypeB に変更しても動くようにすること

## 課題完成時の画面

```
[0]冷房  
[1]暖房  
どのボタンを押しますか? >0  
【エアコンB】快適エコな冷房にしました!  
  
[0]冷房  
[1]暖房  
どのボタンを押しますか? >1  
【エアコンB】快適エコな暖房にしました!  
  
[0]冷房  
[1]暖房  
どのボタンを押しますか? >-1
```

## 課題完成時のクラス図



● J2Kad26C「旧式エアコンの操作 (Adapter パターン)」※pac26d のファイルをコピーすること

すでに製品として市場に出回っている旧式エアコン (OldAircon クラス) も操作できるようにせよ。ただし、OldAircon クラスは修正不可 (Aircon インターフェイスを実装できないもの) とする。

OldAircon クラス

メンバ	説明
private int mode	現在のモード (COOL : 冷房、WARM : 暖房)
public void showMode()	現在のモードを表示する。
public void changeMode()	モードを切り替える (冷房→暖房、暖房→冷房)。
public int getMode()	現在のモードを返す。

OldAirconAdapter クラス (Aircon インターフェイスを実装)

メンバ	説明
private OldAircon ac = new OldAircon();	操作する OldAircon への参照
public void setCool()	現在のモードを表示したのち、モードが暖房のときは冷房へ切り替える。
public void setWarm()	現在のモードを表示したのち、モードが冷房のときは暖房へ切り替える。

課題完成時の画面

```
[0]冷房
[1]暖房
どのボタンを押しますか？>0
【旧式エアコン】ただいま冷房中です！

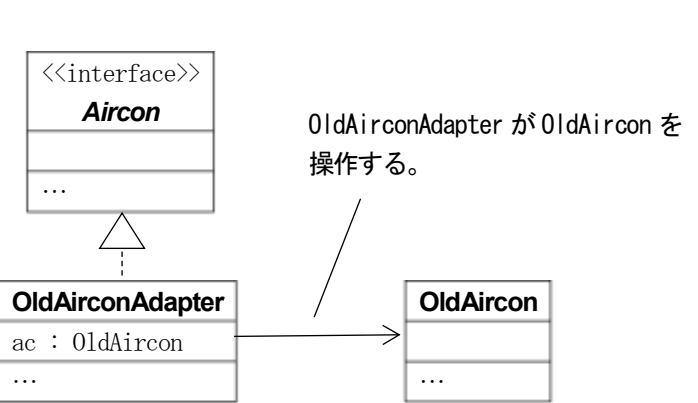
[0]冷房
[1]暖房
どのボタンを押しますか？>1
【旧式エアコン】ただいま冷房中です！
【旧式エアコン】暖房に切り換えました！

[0]冷房
[1]暖房
どのボタンを押しますか？>1
【旧式エアコン】ただいま暖房中です！

[0]冷房
[1]暖房
どのボタンを押しますか？>0
【旧式エアコン】ただいま暖房中です！
【旧式エアコン】冷房に切り換えました！

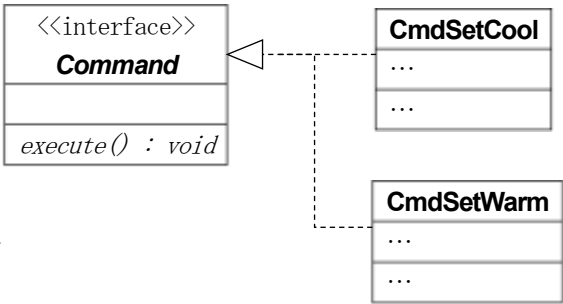
[0]冷房
[1]暖房
どのボタンを押しますか？>-1
```

課題完成時のクラス図



● J2Kad26B「エアコン操作コマンド (Command パターン)」※pac26c のファイルをコピー

エアコンを操作するコマンドをクラスとして作成し、リモコン (Controller クラス) はこのコマンドを実行するように処理を修正せよ。



新規ファイル「Command.java」を追加し、Command インターフェイスと CmdSetCool クラス、CmdSetWarm クラスを作成する。

CmdSetCool クラス (Command インターフェイスを実装)

メンバ	説明
private Aircon ac	操作するエアコンへの参照
コンストラクタ	引数として、操作するエアコンへの参照を受け取り、ac に設定する。
public void execute()	操作を実行する (CmdSetCool : 冷房にする、CmdSetWarm : 暖房) にする。
public String toString()	コマンドの名前を返す (CmdSetCool : 冷房、CmdSetWarm : 暖房)。

Controller クラスの修正

メンバ	説明
private ArrayList<Command> button = new ArrayList<>();	コマンドを格納する配列
public Controller addCommand(Command cmd)	配列 button にコマンド (cmd) を追加する。
public void showCommand()	配列 button に登録されているコマンドの名前を表示する。
public void pushButton(int i)	配列 button の i 番目のコマンドを実行する。

リスト 1 : TypeB エアコンの操作 (ファイル「pac26b.Main.java」)

```
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        Controller rc = new Controller();

        Aircon ac = new OldAirconAdapter();
        rc.addCommand(new CmdSetCool(ac)).addCommand(new CmdSetWarm(ac));

        while(true) {
            :
        }
    }
}
```

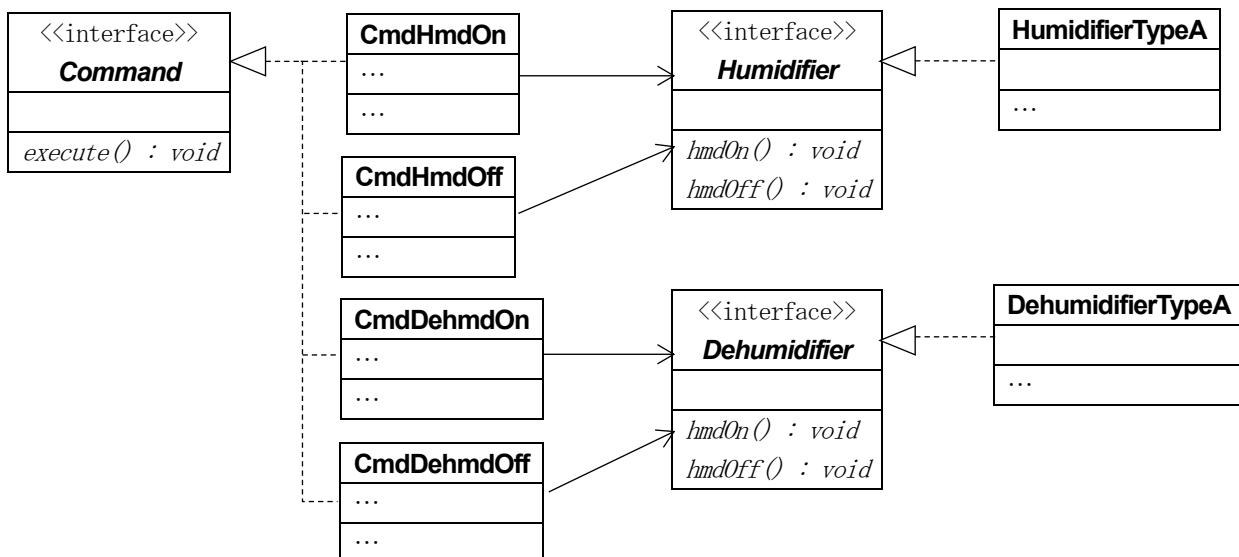
エアコンを設定するのではなく、エアコンを操作するコマンドを追加する。

課題完成時の画面  
(J2Kad26C と同じ)

## ● J2Kad26A 「完成！ECC リモコン！！」※pac26b のファイルをコピー

リモコンに TypeA 加湿器 (HumidifierTypeA クラス) と TypeA 除湿器 (DehumidifierTypeA クラス) の操作も追加せよ。ただし今後、異なるタイプの加湿器・除湿器が接続できるようにインターフェイス (Humidifier インターフェイスと Dehumidifier インターフェイス) を通して操作するようにすること。

## 課題完成時のクラス図



ヒント：Controller クラスは修正しない。修正しなくてもコマンドの追加や変更が可能になっている。

## 課題完成時の画面

```

[0]冷房
[1]暖房
[2]加湿 ON
[3]加湿 OFF
[4]除湿 ON
[5]除湿 OFF
どのボタンを押しますか？>2
【加湿器 A】ON にしました！

[0]冷房
[1]暖房
[2]加湿 ON
[3]加湿 OFF
[4]除湿 ON
[5]除湿 OFF
どのボタンを押しますか？>3
【加湿器 A】OFF にしました！
  
```

## (続き)

```

[0]冷房
[1]暖房
[2]加湿 ON
[3]加湿 OFF
[4]除湿 ON
[5]除湿 OFF
どのボタンを押しますか？>4
【除湿器 A】ON にしました！

[0]冷房
[1]暖房
[2]加湿 ON
[3]加湿 OFF
[4]除湿 ON
[5]除湿 OFF
どのボタンを押しますか？>5
【除湿器 A】OFF にしました！
  
```

● J2Kad26S「オールインワン！」※pac26a をコピーすること

「より良い暮らしを提供する」ECC ホームのライバル会社、「より便利な暮らしを提供する」FDD エレクトロニクスから冷暖房も加湿も除湿もできる新製品が発売された！その名も「オールインワン」(AllInOne クラス)。様々な電化製品を操作する ECC リモコンもこの製品に対応させたい。これまでつちかった技術を総動員して対応できるようにせよ。なお、「オールインワン」は他社製品なので、もちろん修正や変更を加えることはできない。

AllInOne
setCool() : void
setWarm() : void
hmdOn() : void
hmdOff() : void
dehmdOn() : void
dehmdOff() : void

FDD エレクトロニクスの製品はすべての機能が詰まっている！

課題完成時の画面

[0]冷房

[1]暖房

[2]加湿 ON

[3]加湿 OFF

[4]除湿 ON

[5]除湿 OFF

どのボタンを押しますか？>0

【オールインワン】もっと快適エコな冷房にしました！

[0]冷房

[1]暖房

[2]加湿 ON

[3]加湿 OFF

[4]除湿 ON

[5]除湿 OFF

どのボタンを押しますか？>1

【オールインワン】もっと快適エコな暖房にしました！

[0]冷房

[1]暖房

[2]加湿 ON

[3]加湿 OFF

[4]除湿 ON

[5]除湿 OFF

どのボタンを押しますか？>2

【オールインワン】加湿機能をオンにしました！

(続き)

[0]冷房

[1]暖房

[2]加湿 ON

[3]加湿 OFF

[4]除湿 ON

[5]除湿 OFF

どのボタンを押しますか？>3

【オールインワン】加湿機能をオフにしました！

[0]冷房

[1]暖房

[2]加湿 ON

[3]加湿 OFF

[4]除湿 ON

[5]除湿 OFF

どのボタンを押しますか？>4

【オールインワン】除湿機能をオンにしました！

[0]冷房

[1]暖房

[2]加湿 ON

[3]加湿 OFF

[4]除湿 ON

[5]除湿 OFF

どのボタンを押しますか？>5

【オールインワン】除湿機能をオフにしました！

## ● J2Kad26X 「マクロコマンド」※pac26s のファイルをコピーすること

複数のコマンドをまとめて実行するマクロコマンドを実装せよ。追加するマクロは以下の2つ。

- ・夏モード 冷房 ON・加湿 OFF・除湿 ON にする。
- ・冬モード 暖房 ON・加湿 ON・除湿 OFF にする。

## 課題完成時の画面（オールインワンで操作した場合）

```
[0]冷房
[1]暖房
[2]加湿 ON
[3]加湿 OFF
[4]除湿 ON
[5]除湿 OFF
[6]冷房/加湿 OFF/除湿 ON/
[7]暖房/加湿 ON/除湿 OFF/
どのボタンを押しますか？>6
【オールインワン】 もっと快適エコな冷房にしました！
【オールインワン】 加湿機能をオフにしました！
【オールインワン】 除湿機能をオンにしました！

[0]冷房
[1]暖房
[2]加湿 ON
[3]加湿 OFF
[4]除湿 ON
[5]除湿 OFF
[6]冷房/加湿 OFF/除湿 ON/
[7]暖房/加湿 ON/除湿 OFF/
どのボタンを押しますか？>7
【オールインワン】 もっと快適エコな暖房にしました！
【オールインワン】 加湿機能をオンにしました！
【オールインワン】 除湿機能をオフにしました！
```

← コマンド名はまとめて実行するコマンドの名前を「/」でつなげて表示する。

ヒント：Composite パターン（J2Kad23 または検索）を適用する。