
デザインパターン

22.Command

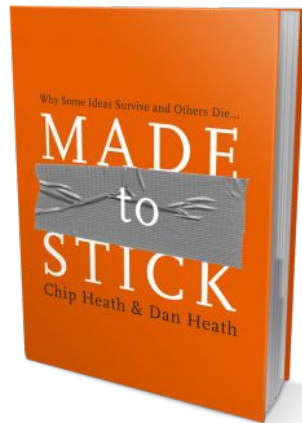
どんなもの？ 使い道は？

あるオブジェクトに対して要求を送る = そのオブジェクトのメソッドを呼び出すこと

`Object.doSomething1()`, `Object.doSomething2()` ...

要求自体をオブジェクトにして、

そのオブジェクトを引数に渡すのがcommandパターン！



ビーカーに食塩や水を入れたりして、食塩水に関する実験をするシーン

```
public class Beaker {  
    private double water = 0; //水  
    private double salt = 0; //食塩  
    private boolean melted; //食塩がすべて溶けたときtrue、溶け残ったときfalse  
  
    public static final int ADD_SALT = 1; //食塩を加えて、かき混ぜる場合  
    public static final int ADD_WATER = 2; //水を加えて、かき混ぜる場合  
  
    public void addSalt(double salt) {  
        this.salt += salt;  
    }  
  
    public void addWater(double water) {  
        this.water += water;  
    }  
  
    public void mix() {}  
  
    public void experiment(int param) {  
        if (param == ADD_SALT) {  
  
        } else if (param == ADD_WATER) {  
        }  
    }  
}
```

実行する生徒クラス

```
public class Student {  
    public static void main(String[] args) {  
        //水100gに食塩を1gずつ加えて飽和食塩水を作る実験  
        Beaker beaker = new Beaker(100,0);  
        beaker.experiment(Beaker.ADD_SALT);  
  
        //食塩10gに水を10gずつ加えて飽和食塩水を作る実験  
        Beaker beaker2 = new Beaker(0,10);  
        beaker2.experiment(Beaker.ADD_WATER);  
    }  
}
```

処理を追加するときに厄介 修正が2クラスに必要！

```
public class Beaker {  
    private double water = 0; //水  
    private double salt = 0; //食塩  
    private boolean melted; //食塩がすべて溶けたときtrue、溶け残ったときfalse  
  
    public static final int ADD_SALT = 1; //食塩を加えて、かき混ぜる場合  
    public static final int ADD_WATER = 2; //水を加えて、かき混ぜる場合  
    public static final int MAKE_SALT_WATER = 3; //食塩水を作る場合  
  
    public void addSalt(double salt) {  
        this.salt += salt;  
    }  
    public void addWater(double water) {  
        this.water += water;  
    }  
    public void mix() {}  
  
    public void experiment(int param) {  
        if (param == ADD_SALT) {  
        } else if (param == ADD_WATER) {  
        } else if (param == MAKE_SALT_WATER) {  
        }  
    }  
}
```

実行する生徒クラス

```
public class Student {  
    public static void main(String[] args) {  
        //水100gに食塩を1gずつ加えて飽和食塩水を作る実験  
        Beaker beaker = new Beaker(100,0);  
        beaker.experiment(Beaker.ADD_SALT);  
  
        //食塩10gに水を10gずつ加えて飽和食塩水を作る実験  
        Beaker beaker2 = new Beaker(0,10);  
        beaker2.experiment(Beaker.ADD_WATER);  
  
        Beaker beaker3 = new Beaker(90,10);  
        beaker3.experiment(Beaker.MAKE_SALT_WATER);  
    }  
}
```

実験内容をオブジェクトとして実行するぞ！

具体的にどうやって実装する？

→実験内容についてのスーパークラスを作る！

→そこから派生して、いろんな実験内容のオブジェクトを作る

//実験コマンドのスーパークラス

```
public abstract class Command {
```

```
    //ビーカー
```

```
    protected Beaker beaker;
```

```
    //ビーカーをセットするメソッド
```

```
    public void setBeaker(Beaker beaker) {
```

```
        this.beaker = beaker;
```

```
    }
```

```
    //要求内容を実行する抽象メソッド
```

```
    public abstract void execute();
```

```
}
```

実際の実行処理

実際のオブジェクト内容(実験内容)

```
public class AddSaltCommand extends Command {  
    public void execute() {  
        //実際の処理..長いので省略  
    }  
}
```

```
public class AddWaterCommand extends Command {  
    public void execute() {  
        //実際の処理..長いので省略  
    }  
}
```

```
public class MakeSaltWaterCommand extends Command {  
    public void execute() {  
        //実際の処理..長いので省略  
    }  
}
```

```
public class Student {  
    public static void main(String[] args) {  
        //実験内容(コマンドオブジェクト)を用意する  
        Command addSalt = new AddSaltCommand();  
        Command addWater = new AddWaterCommand();  
        Command makeSaltWater = new MakeSaltWaterCommand();  
  
        //実験セットを実験内容にセットする  
        addSalt.setBeaker(new Beaker(100,0));  
        //水100gの入ったビーカーをセットする  
        addWater.setBeaker(new Beaker(0,10));  
        //食塩10gの入ったビーカーをセットする  
        makeSaltWater.setBeaker(new Beaker(90,10));  
        //水90g、食塩10gの入ったビーカーをセットする  
  
        //実験を行う  
        addSalt.execute(); //食塩を加えて飽和食塩水を作る実験  
        addWater.execute(); //水を加えて飽和食塩水を作る実験  
        makeSaltWater.execute(); //10%の食塩水100gを作る実験  
    }  
}
```