

# Increasing Flexibility by Removing Enums and Switch Statements

---



**Zoran Horvat**

CEO AT CODING HELMET

@zoranh75 <http://codinghelmet.com>



Pluralsightsrccomcodinghelmetmoreoojavadeemo

Part.javaWarranty.javaLifetimeWarranty.javaVoidWarranty.javaArticle.javaDemo.javaTimeLimitedWarranty.java

11  
12 @  
13  
14 A  
15 A  
16 A  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33

```
private void claimWarranty(Article article) {  
    LocalDate today = LocalDate.now();  
    article.getMoneyBackGuarantee().on(today).claim(this::offerMoneyBack);  
    article.getExpressWarranty().on(today).claim(this::offerRepair);  
    article.getExtendedWarranty().on(today).claim(this::offerSensorRepair);  
    System.out.println("-----");  
}  
  
public void run() {  
    LocalDate sellingDate = LocalDate.now().minus( amountToSubtract: 40, ChronoUnit.DAYS);  
    Warranty moneyBack = new TimeLimitedWarranty(sellingDate, Duration.ofDays(60));  
    Warranty warranty = new TimeLimitedWarranty(sellingDate, Duration.ofDays(365));  
  
    Part sensor = new Part(sellingDate);  
    Warranty sensorWarranty = new TimeLimitedWarranty(sellingDate, Duration.ofDays(90));  
  
    Article item = new Article(moneyBack, warranty).install(sensor, sensorWarranty);  
  
    this.claimWarranty(item);  
}
```

Ant Build  
Maven

1  
2: Favorites

```
graph TD; A[Cube] --- B[Gear]; A --- C[Monitor]; A --- D[Signal Tower]; B --- E[Lines]; C --- F[Lines]; D --- G[Lines]
```

Demo

3: Find6: TODOTerminal4: RunEvent Log

Pluralsightsrccomcodinghelmetmoreoojavademo

Part.javaWarranty.javaLifetimeWarranty.javaVoidWarranty.javaArticle.javaDemo.javaTimeLimitedWarranty.java

11  
12 @  
13  
14 A  
15 A  
16 A  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33

```
private void claimWarranty(Article article) {  
    LocalDate today = LocalDate.now();  
    article.getMoneyBackGuarantee().on(today).claim(this::offerMoneyBack);  
    article.getExpressWarranty().on(today).claim(this::offerRepair);  
    article.getExtendedWarranty().on(today).claim(this::offerSensorRepair);  
    System.out.println("-----");  
}  
  
public void run() {  
    LocalDate sellingDate = LocalDate.now().minus( amountToSubtract: 40, ChronoUnit.DAYS);  
    Warranty moneyBack = new TimeLimitedWarranty(sellingDate, Duration.ofDays(60));  
    Warranty warranty = new TimeLimitedWarranty(sellingDate, Duration.ofDays(365));  
  
    Part sensor = new Part(sellingDate);  
    Warranty sensorWarranty = new TimeLimitedWarranty(sellingDate, Duration.ofDays(90));  
  
    Article item = new Article(moneyBack, warranty).install(sensor, sensorWarranty);  
  
    this.claimWarranty(item);  
}  
}
```

1: Project  
2: Structure  
3: Find  
4: Run

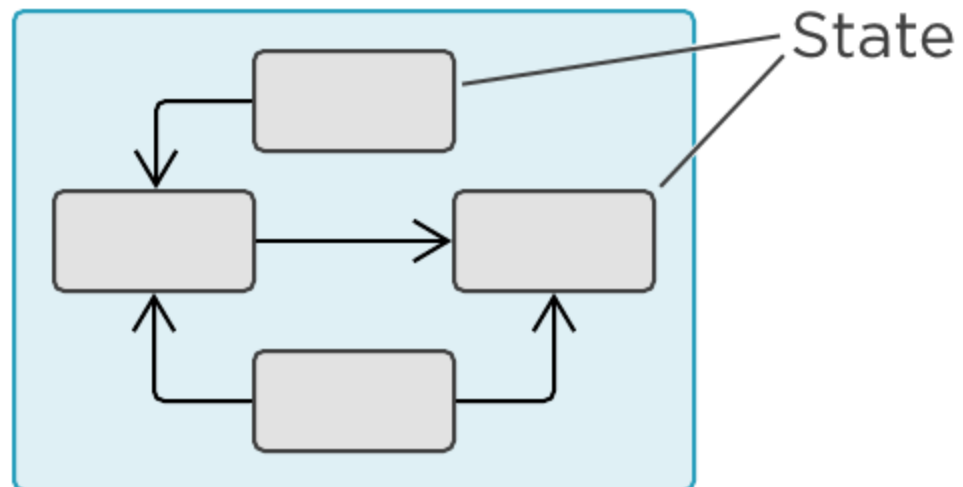
Ant Build  
Maven

Demo

Event Log

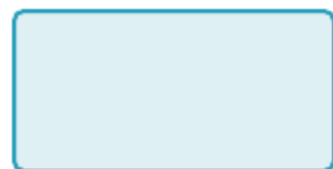
```
11
12 @
13 private void claimWarranty(Article article) {
14     LocalDate today = LocalDate.now();
15     article.getMoneyBackGuarantee().on(today).claim(this::offerMoneyBack);
16     article.getExpressWarranty().on(today).claim(this::offerRepair);
17     article.getExtendedWarranty().on(today).claim(this::offerSensorRepair);
18     System.out.println("-----");
19 }
```

## An object

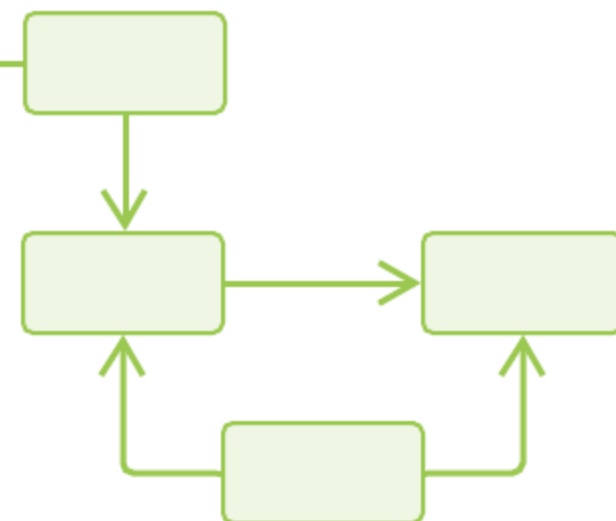


```
11
12 @
13 private void claimWarranty(Article article) {
14     LocalDate today = LocalDate.now();
15     article.getMoneyBackGuarantee().on(today).claim(this::offerMoneyBack);
16     article.getExpressWarranty().on(today).claim(this::offerRepair);
17     article.getExtendedWarranty().on(today).claim(this::offerSensorRepair);
18     System.out.println("-----");
19 }
```

An object



External state



Pluralsight

src > com > codinghelmet > moreoojava > Demo

Main

Part.java

Warranty.java

LifetimeWarranty.java

VoidWarranty.java

Article.java

Demo.java

DeviceStatus.java

TimeLimitedWarranty.java

1: Project

Ant BuildMaven

11

12 @

13

14

15

16

17 A↑

18

19

20 A↑

21 A↑

22

23

24

25

26 A↑

27 A↑

28

29

30

31

32

33

34

35

36

37

38

```
private void claimWarranty(Article article, DeviceStatus status) {  
    LocalDate today = LocalDate.now();  
  
    switch (status) {  
        case ALL_FINE:  
            article.getMoneyBackGuarantee().on(today).claim(this::offerMoneyBack);  
            break;  
        case NOT_OPERATIONAL:  
            article.getMoneyBackGuarantee().on(today).claim(this::offerMoneyBack);  
            article.getExpressWarranty().on(today).claim(this::offerRepair);  
            break;  
        case VISIBLY_DAMAGED:  
            break;  
        case SENSOR_FAILED:  
            article.getMoneyBackGuarantee().on(today).claim(this::offerMoneyBack);  
            article.getExtendedWarranty().on(today).claim(this::offerSensorRepair);  
            break;  
    }  
  
    System.out.println("-----");  
}  
  
public void run() {  
    LocalDate sellingDate = LocalDate.now().minus(amountToSubtract 40, ChronoUnit.DAYS);  
    Warranty moneyBack = new TimeLimitedWarranty(sellingDate, Duration.ofDays(60));  
    Warranty warranty = new TimeLimitedWarranty(sellingDate, Duration.ofDays(365));  
}
```

Computed jump

```
graph LR; A{ } --> B[ ]; B --> C[ ]; C --> D[ ]; D --> E[ ]; E --> F[ ]; F --> G[ ]; G --> H[ ]
```

Demo

3: Find

6: TODO

Terminal

0: Messages

Event Log

Pluralsight

src > com > codinghelmet > moreoojava > E DeviceStatus

Ant Build Maven

1: Project

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

2: Favorites

3: Find

4: TODO

5: Terminal

6: Messages

7: Event Log

Part.java x

Warranty.java x

LifetimeWarranty.java x

VoidWarranty.java x

Article.java x

Demo.java x

TimeLimitedWarranty.java x

DeviceStatus.java x

public enum DeviceStatus {

ALL\_FINE,

NOT\_OPERATIONAL,

VISIBLY\_DAMAGED,

SENSOR\_FAILED,

NOT\_OPERATIONAL\_DAMAGED,

NOT\_OPERATIONAL\_SENSOR\_FAILED,

DAMAGED\_SENSOR\_FAILED,

NOT\_OPERATIONAL\_DAMAGED\_SENSOR\_FAILED;

private final int id;

private DeviceStatus(int id) {

this.id = id;

}

2<sup>4</sup>

2<sup>3</sup>

2<sup>2</sup>

2<sup>1</sup>

2<sup>0</sup>



Pluralsight

src > com > codinghelmet > moreoojava > E DeviceStatus

Main

1: Project

Part.java x Warranty.java x LifetimeWarranty.java x VoidWarranty.java x Article.java x Demo.java x TimeLimitedWarranty.java x DeviceStatus.java x

Ant Build Maven

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

```
public enum DeviceStatus {  
    ALL_FINE,  
    NOT_OPERATIONAL(id: 1),  
    VISIBLY_DAMAGED,  
    SENSOR_FAILED,  
    NOT_OPERATIONAL_DAMAGED,  
    NOT_OPERATIONAL_SENSOR_FAILED,  
    DAMAGED_SENSOR_FAILED,  
    NOT_OPERATIONAL_DAMAGED_SENSOR_FAILED;  
  
    private final int id;  
  
    private DeviceStatus(int id) {  
        this.id = id;  
    }  
}
```

2<sup>4</sup>

2<sup>3</sup>

2<sup>2</sup>

2<sup>1</sup>

2<sup>0</sup>

1

DeviceStatus

3: Find

6: TODO

Terminal

0: Messages

Event Log



Pluralsight

src > com > codinghelmet > moreoojava > E DeviceStatus

Main

Ant Build Maven

1: Project

Part.java × Warranty.java × LifetimeWarranty.java × VoidWarranty.java × Article.java × Demo.java × TimeLimitedWarranty.java × DeviceStatus.java ×

2  
3 public enum DeviceStatus {  
4     ALL\_FINE,  
5     NOT\_OPERATIONAL ( id: 1 ),  
6     VISIBLY\_DAMAGED ( id: 2 ),  
7     SENSOR\_FAILED,  
8     NOT\_OPERATIONAL DAMAGED,  
9     NOT\_OPERATIONAL SENSOR FAILED,  
10    DAMAGED SENSOR FAILED,  
11    NOT\_OPERATIONAL DAMAGED SENSOR FAILED;  
12  
13    private final int id;  
14  
15    private DeviceStatus(int id) {  
16       this.id = id;  
17    }  
18 }  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29

DeviceStatus

2: Favorites

3: Find

6: TODO

Terminal

0: Messages

Event Log

2<sup>4</sup> 2<sup>3</sup> 2<sup>2</sup> 2<sup>1</sup> 2<sup>0</sup>

1 0

Pluralsight

src > com > codinghelmet > moreoojava > E DeviceStatus

Main

Ant Build Maven

1: Project

Part.java x Warranty.java x LifetimeWarranty.java x VoidWarranty.java x Article.java x Demo.java x TimeLimitedWarranty.java x DeviceStatus.java x

2  
3 public enum DeviceStatus {  
4     ALL\_FINE,  
5     NOT\_OPERATIONAL( id: 1 ),  
6     VISIBLY\_DAMAGED( id: 2 ),  
7     SENSOR\_FAILED( id: 4 ),  
8     NOT\_OPERATIONAL\_DAMAGED,  
9     NOT\_OPERATIONAL\_SENSOR\_FAILED,  
10    DAMAGED\_SENSOR\_FAILED,  
11    NOT\_OPERATIONAL\_DAMAGED\_SENSOR\_FAILED;  
12  
13    private final int id;  
14  
15    private DeviceStatus(int id) {  
16       this.id = id;  
17    }  
18 }  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29

DeviceStatus

2: Favorites

7: Structure

3: Find

6: TODO

Terminal

0: Messages

Event Log

private final int id;

```
private DeviceStatus(int id) {  
    this.id = id;  
}
```



Pluralsight

src > com > codinghelmet > moreoojava > E DeviceStatus

Part.java × Warranty.java × LifetimeWarranty.java × VoidWarranty.java × Article.java × Demo.java × TimeLimitedWarranty.java × DeviceStatus.java ×

1: Project

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

2: Favorites

3: Structure

Ant Build

Maven

DeviceStatus

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

public enum DeviceStatus {

ALL\_FINE ( id: 0 ),

NOT\_OPERATIONAL ( id: 1 ),

VISIBLY\_DAMAGED ( id: 2 ),

SENSOR\_FAILED ( id: 4 ),

NOT\_OPERATIONAL\_DAMAGED,

NOT\_OPERATIONAL\_SENSOR\_FAILED,

DAMAGED\_SENSOR\_FAILED,

NOT\_OPERATIONAL\_DAMAGED\_SENSOR\_FAILED;

private final int id;

private DeviceStatus(int id) {

this.id = id;

}

}

0 0 0 0 0

2<sup>4</sup> 2<sup>3</sup> 2<sup>2</sup> 2<sup>1</sup> 2<sup>0</sup>

Find

TODO

Terminal

Messages

Event Log

Pluralsight

src > com > codinghelmet > moreoojava > E DeviceStatus

Part.java × Warranty.java × LifetimeWarranty.java × VoidWarranty.java × Article.java × Demo.java × TimeLimitedWarranty.java × DeviceStatus.java ×

Ant Build Maven

1: Project

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

```
public enum DeviceStatus {  
    ALL_FINE( id: 0 ),  
    NOT_OPERATIONAL( id: 1 ),  
    VISIBLY_DAMAGED( id: 2 ),  
    SENSOR_FAILED( id: 4 ),  
    NOT_OPERATIONAL_DAMAGED( id: NOT_OPERATIONAL.id | VISIBLY_DAMAGED.id ),  
    NOT_OPERATIONAL_SENSOR_FAILED,  
    DAMAGED_SENSOR_FAILED,  
    NOT_OPERATIONAL_DAMAGED_SENSOR_FAILED;  
  
    private final int id;  
  
    private DeviceStatus(int id) {  
        this.id = id;  
    }  
}
```

private final int id;

00011

2<sup>4</sup> 2<sup>3</sup> 2<sup>2</sup> 2<sup>1</sup> 2<sup>0</sup>

2: Favorites 3: Structure

DeviceStatus

3: Find 6: TODO Terminal 0: Messages Event Log

Pluralsight

src > com > codinghelmet > moreoojava > E DeviceStatus

Part.java × Warranty.java × LifetimeWarranty.java × VoidWarranty.java × Article.java × Demo.java × TimeLimitedWarranty.java × DeviceStatus.java ×

Ant Build Maven

1: Project

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

```
public enum DeviceStatus {  
    ALL_FINE( id: 0 ),  
    NOT_OPERATIONAL( id: 1 ),  
    VISIBLY_DAMAGED( id: 2 ),  
    SENSOR_FAILED( id: 4 ),  
    NOT_OPERATIONAL_DAMAGED( id: NOT_OPERATIONAL.id | VISIBLY_DAMAGED.id ),  
    NOT_OPERATIONAL_SENSOR_FAILED( id: NOT_OPERATIONAL.id | SENSOR_FAILED.id ),  
    DAMAGED_SENSOR_FAILED,  
    NOT_OPERATIONAL_DAMAGED_SENSOR_FAILED;  
  
    private final int id;  
  
    private DeviceStatus(int id) {  
        this.id = id;  
    }  
}
```

private final int id;

00101

2<sup>4</sup> 2<sup>3</sup> 2<sup>2</sup> 2<sup>1</sup> 2<sup>0</sup>

2: Favorites

2: Structure

DeviceStatus

3: Find

6: TODO

Terminal

0: Messages

Event Log

Pluralsight

src > com > codinghelmet > moreoojava > E DeviceStatus

Main

Ant Build Maven

1: Project

Part.java × Warranty.java × LifetimeWarranty.java × VoidWarranty.java × Article.java × Demo.java × TimeLimitedWarranty.java × DeviceStatus.java ×

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

```
public enum DeviceStatus {  
    ALL_FINE( id: 0 ),  
    NOT_OPERATIONAL( id: 1 ),  
    VISIBLY_DAMAGED( id: 2 ),  
    SENSOR_FAILED( id: 4 ),  
    NOT_OPERATIONAL_DAMAGED( id: NOT_OPERATIONAL.id | VISIBLY_DAMAGED.id ),  
    NOT_OPERATIONAL_SENSOR_FAILED( id: NOT_OPERATIONAL.id | SENSOR_FAILED.id ),  
    DAMAGED_SENSOR_FAILED( id: VISIBLY_DAMAGED.id | SENSOR_FAILED.id ),  
    NOT_OPERATIONAL_DAMAGED_SENSOR_FAILED;  
  
    private final int id;  
  
    private DeviceStatus(int id) {  
        this.id = id;  
    }  
}
```

private final int id;

00110

2<sup>4</sup> 2<sup>3</sup> 2<sup>2</sup> 2<sup>1</sup> 2<sup>0</sup>

2: Favorites 3: Find 6: TODO Terminal 0: Messages Event Log

DeviceStatus

Pluralsight

src > com > codinghelmet > moreoojava > E DeviceStatus

Main

Part.java × Warranty.java × LifetimeWarranty.java × VoidWarranty.java × Article.java × Demo.java × TimeLimitedWarranty.java × DeviceStatus.java ×

1: Project

2

3 public enum DeviceStatus {

4 ALL\_FINE( id: 0),

5 NOT\_OPERATIONAL( id: 1),

6 VISIBLY\_DAMAGED( id: 2),

7 SENSOR\_FAILED( id: 4),

8 NOT\_OPERATIONAL\_DAMAGED( id: NOT\_OPERATIONAL.id | VISIBLY\_DAMAGED.id),

9 NOT\_OPERATIONAL\_SENSOR\_FAILED( id: NOT\_OPERATIONAL.id | SENSOR\_FAILED.id),

10 DAMAGED\_SENSOR\_FAILED( id: VISIBLY\_DAMAGED.id | SENSOR\_FAILED.id),

11 NOT\_OPERATIONAL\_DAMAGED\_SENSOR\_FAILED( id: NOT\_OPERATIONAL.id | VISIBLY\_DAMAGED.id | SENSOR\_FAILED.id);

12

13 private final int id;

14

15 private DeviceStatus(int id) {

16 this.id = id;

17 }

18 }

19

20

21

22

23

24

25

26

27

28

29

2: Favorites

3: Find

4: TODO

5: Terminal

6: Messages

7: Structure

8: DeviceStatus

9: Ant Build

10: Maven

11: Event Log

Diagram illustrating the bit representation of the `id` field in the `DeviceStatus` enum.

The `id` field is a 5-bit integer, represented by the bits 0, 0, 1, 1, 1.

The bits are labeled with their corresponding powers of 2:  $2^4$ ,  $2^3$ ,  $2^2$ ,  $2^1$ , and  $2^0$ .

The diagram shows the `id` field in the `DeviceStatus` enum and its corresponding bit representation.



## Logic

## State

DeviceStatus status

Optional&lt;LocalDate&gt; sensorFailureDate)

```
14
15 @
16
17
18 if (status.equals(DeviceStatus.allFine())) {
19     this.claimMoneyBack(article, today);
20 } else if (status.equals(DeviceStatus.notOperational())) {
21     this.claimMoneyBack(article, today);
22     this.claimExpress(article, today);
23 } else if (status.equals(DeviceStatus.visiblyDamaged())) {
24 } else if (status.equals(DeviceStatus.sensorFailed())) {
25     this.claimMoneyBack(article, today);
26     this.claimExtended(article, today, sensorFailureDate);
27 } else if (status.equals(DeviceStatus.notOperational().add(DeviceStatus.visiblyDamaged()))) {
28     this.claimExpress(article, today);
29 } else if (status.equals(DeviceStatus.notOperational().add(DeviceStatus.sensorFailed()))) {
30     this.claimMoneyBack(article, today);
31     this.claimExpress(article, today);
32     this.claimExtended(article, today, sensorFailureDate);
33 } else if (status.equals(DeviceStatus.visiblyDamaged().add(DeviceStatus.sensorFailed()))) {
34     this.claimExtended(article, today, sensorFailureDate);
35 } else { // notOperational() + visiblyDamaged() + sensorFailed()
36     this.claimExpress(article, today);
37     this.claimExtended(article, today, sensorFailureDate);
38 }
39
40
41
```

DeviceStatus status	Optional<LocalDate> sensorFailureDate)
---------------------	--



```
if (status.equals(DeviceStatus.allFine())) {
    this.claimMoneyBack(article, today);
} else if (status.equals(DeviceStatus.notOperational())) {
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.visiblyDamaged())) {
} else if (status.equals(DeviceStatus.sensorFailed())) {
    this.claimMoneyBack(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.visiblyDamaged().add(
    this.claimExtended(article, today, sensorFailureDate);
} else { // notOperational() + visiblyDamaged() + sensorFai
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
}
```

## Poor change handling symptoms

- Implementation must change
- All duplicates must change
- Even on unrelated changes

DeviceStatus status	Optional<LocalDate> sensorFailureDate)
---------------------	--



```
if (status.equals(DeviceStatus.allFine())) {
    this.claimMoneyBack(article, today);
} else if (status.equals(DeviceStatus.notOperational())) {
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.visiblyDamaged())) {
} else if (status.equals(DeviceStatus.sensorFailed())) {
    this.claimMoneyBack(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.visiblyDamaged().add(
    this.claimExtended(article, today, sensorFailureDate);
} else { // notOperational() + visiblyDamaged() + sensorFai
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
}
```

## Visible signs of lacking objects

- Raw state representation
- Consumer implementation
- Hard-coded branching

DeviceStatus status	Optional<LocalDate> sensorFailureDate)
---------------------	--



```
if (status.equals(DeviceStatus.allFine())) {
    this.claimMoneyBack(article, today);
} else if (status.equals(DeviceStatus.notOperational())) {
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.visiblyDamaged())) {
} else if (status.equals(DeviceStatus.sensorFailed())) {
    this.claimMoneyBack(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.visiblyDamaged().add(
    this.claimExtended(article, today, sensorFailureDate);
} else { // notOperational() + visiblyDamaged() + sensorFai
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
}
```

## Expected code qualities

- Flexible behavior
- Support for changing rules

## An active object

```
if (status.equals(DeviceStatus.allFine())) {
    this.claimMoneyBack(article, today);
} else if (status.equals(DeviceStatus.notOperational())) {
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.visiblyDamaged())) {
} else if (status.equals(DeviceStatus.sensorFailed())) {
    this.claimMoneyBack(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.visiblyDamaged().add(
    this.claimExtended(article, today, sensorFailureDate);
} else { // notOperational() + visiblyDamaged() + sensorFai
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
}
```

## An active object

```
if (status.equals(DeviceStatus.allFine())) {  
    this.claimMoneyBack(article, today);  
} else if (status.equals(DeviceStatus.notOperational())) {  
    this.claimMoneyBack(article, today);  
    this.claimExpress(article, today);  
} else if (status.equals(DeviceStatus.visiblyDamaged())) {  
} else if (status.equals(DeviceStatus.sensorFailed())) {  
    this.claimMoneyBack(article, today);  
    this.claimExtended(article, today, sensorFailureDate);  
} else if (status.equals(DeviceStatus.notOperational().add(  
    this.claimExpress(article, today);  
} else if (status.equals(DeviceStatus.notOperational().add(  
    this.claimMoneyBack(article, today);  
    this.claimExpress(article, today);  
    this.claimExtended(article, today, sensorFailureDate);  
} else if (status.equals(DeviceStatus.visiblyDamaged().add(  
    this.claimExtended(article, today, sensorFailureDate);  
} else { // notOperational() + visiblyDamaged() + sensorFai  
    this.claimExpress(article, today);  
    this.claimExtended(article, today, sensorFailureDate);  
}
```



## An active object

```
if (status.equals(DeviceStatus.allFine())) {
    this.claimMoneyBack(article, today);
} else if (status.equals(DeviceStatus.notOperational())) {
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.visiblyDamaged())) {
} else if (status.equals(DeviceStatus.sensorFailed())) {
    this.claimMoneyBack(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.visiblyDamaged().add(
    this.claimExtended(article, today, sensorFailureDate);
} else { // notOperational() + visiblyDamaged() + sensorFai
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
}
```





## An active object

```
if (status.equals(DeviceStatus.allFine())) {
    this.claimMoneyBack(article, today);
} else if (status.equals(DeviceStatus.notOperational())) {
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.visiblyDamaged())) {
} else if (status.equals(DeviceStatus.sensorFailed())) {
    this.claimMoneyBack(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.visiblyDamaged().add(
    this.claimExtended(article, today, sensorFailureDate);
} else { // notOperational() + visiblyDamaged() + sensorFai
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
}
```



## An active object

```
if (status.equals(DeviceStatus.allFine())) {  
    this.claimMoneyBack(article, today);  
} else if (status.equals(DeviceStatus.notOperational())) {  
    this.claimMoneyBack(article, today);  
    this.claimExpress(article, today);  
} else if (status.equals(DeviceStatus.visiblyDamaged())) {  
} else if (status.equals(DeviceStatus.sensorFailed())) {  
    this.claimMoneyBack(article, today);  
    this.claimExtended(article, today, sensorFailureDate);  
} else if (status.equals(DeviceStatus.notOperational().add(  
    this.claimExpress(article, today);  
} else if (status.equals(DeviceStatus.notOperational().add(  
    this.claimMoneyBack(article, today);  
    this.claimExpress(article, today);  
    this.claimExtended(article, today, sensorFailureDate);  
} else if (status.equals(DeviceStatus.visiblyDamaged().add(  
    this.claimExtended(article, today, sensorFailureDate);  
} else { // notOperational() + visiblyDamaged() + sensorFai  
    this.claimExpress(article, today);  
    this.claimExtended(article, today, sensorFailureDate);  
}
```



## An active object

```
if (status.equals(DeviceStatus.allFine())) {  
    this.claimMoneyBack(article, today);  
} else if (status.equals(DeviceStatus.notOperational())) {  
    this.claimMoneyBack(article, today);  
    this.claimExpress(article, today);  
} else if (status.equals(DeviceStatus.visiblyDamaged())) {  
} else if (status.equals(DeviceStatus.sensorFailed())) {  
    this.claimMoneyBack(article, today);  
    this.claimExtended(article, today, sensorFailureDate);  
} else if (status.equals(DeviceStatus.notOperational().add(  
    this.claimExpress(article, today);  
} else if (status.equals(DeviceStatus.notOperational().add(  
    this.claimMoneyBack(article, today);  
    this.claimExpress(article, today);  
    this.claimExtended(article, today, sensorFailureDate);  
} else if (status.equals(DeviceStatus.visiblyDamaged().add(  
    this.claimExtended(article, today, sensorFailureDate);  
} else { // notOperational() + visiblyDamaged() + sensorFai  
    this.claimExpress(article, today);  
    this.claimExtended(article, today, sensorFailureDate);  
}
```



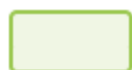
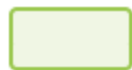
## An active object

```
if (status.equals(DeviceStatus.allFine())) {
    this.claimMoneyBack(article, today);
} else if (status.equals(DeviceStatus.notOperational())) {
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.visiblyDamaged())) {
} else if (status.equals(DeviceStatus.sensorFailed())) {
    this.claimMoneyBack(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.visiblyDamaged().add(
    this.claimExtended(article, today, sensorFailureDate);
} else { // notOperational() + visiblyDamaged() + sensorFai
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
}
```



## An active object

```
if (status.equals(DeviceStatus.allFine())) {
    this.claimMoneyBack(article, today);
} else if (status.equals(DeviceStatus.notOperational())) {
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.visiblyDamaged())) {
} else if (status.equals(DeviceStatus.sensorFailed())) {
    this.claimMoneyBack(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.visiblyDamaged().add(
    this.claimExtended(article, today, sensorFailureDate);
} else { // notOperational() + visiblyDamaged() + sensorFai
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
}
```



An active  
object

Chain  
of  
rules

```
if (status.equals(DeviceStatus.allFine())) {
    this.claimMoneyBack(article, today);
} else if (status.equals(DeviceStatus.notOperational())) {
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.visiblyDamaged())) {
} else if (status.equals(DeviceStatus.sensorFailed())) {
    this.claimMoneyBack(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimExpress(article, today);
} else if (status.equals(DeviceStatus.notOperational().add(
    this.claimMoneyBack(article, today);
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
} else if (status.equals(DeviceStatus.visiblyDamaged().add(
    this.claimExtended(article, today, sensorFailureDate);
} else { // notOperational() + visiblyDamaged() + sensorFai
    this.claimExpress(article, today);
    this.claimExtended(article, today, sensorFailureDate);
}
```

# Summary



## Dealing with non-binary decisions

- Traditional design relies on an enum
- Branching implemented with `switch`
- Alternatively, a chain of `if-else` blocks





# Summary



## Downsides of multiway branching

- What if flags had to be combined?
- That leads to combinatorial explosion
- What if more state should be added?
- That complicates branching conditions
- What if more flag values are added?
- All consuming code must be fixed

# Summary



## Modeling state with objects

- New state means to add a new class
- Existing classes remain



# Summary




## Migrating an enum to a class

- enum values become concrete instances
- Factory functions combine instances
- Objects could still be compared
- Consuming code could remain
- Then let the objects expose behavior
- Join the test pattern with the action

# Summary



 **Next module:**  
Turning Chained Branching  
into the Chain of Rule Objects

