

ממשקים – המפקח. דוגמה לממשק State Machine

התבנית State מאפשרת לאובייקט לשנות את התנהגותו בהתאם לשינוי המצב הפנימי שלו. התבנית מהווה פתרון לבעיה איך לעשות התנהגות של אובייקט תלוי במצבו.

בתבנית זו, אנו יוצרים אובייקטים המייצגים מצבים שונים ואובייקט הקשר שהתנהגותו משתנה כאשר משתנה את מצבו של אובייקטים שלו.

לדוגמה, ההתנהגות של מכונת קפה תלויה בסכום שמכניסים בו וסוג השתייה שבוחרים בו. ההתנהגות של טלוויזיה תלויה בכפתור השלט שלוחצים עליו.

יישום: אנו נבנה ממשק הנקרא State שמגדיר פעולה ומחלקות מצבים המממשות את הממשק. Context היא מחלקה המפעילה את המצבים.

פתרון שגוי: חיפוש המצבים בעזרת משפט switch-case. הקוד שנוצר אינו מודולארי, כדי להוסיף מצב חדש אנו חייבים לשנות את הקוד,

פתרון המבוסס על גישה מוכוונת עצמים מאפשר להוסיף מצבים חדשים ע"י הוספת מחלקה חדשה המתארת את המצב החדש.

דוגמה: התנהגות של טלוויזיה משתנה בהתאם למצב השלט. (א) יוצרים ממשק;

```
interface State {
    public void doAction();
}
```

(ב) יוצרים מספר מחלקות המממשות את הממשק:

```
class TVStartState implements State {
    @Override
    public void doAction() {
        System.out.println("TV is turned ON\n");
    }
}

class TVStopState implements State {
    @Override
    public void doAction() {
        System.out.println("TV is turned OFF\n");
    }
}

class TVChannel_1_State implements State {
    @Override
    public void doAction() {
        System.out.println("TV state: channel 1\n");
    }
}

class TVChannel_2_State implements State {
    @Override
    public void doAction() {
        System.out.println("TV state: channel 2\n");
    }
}
```

```
class TVContext {
    private State tvState;

    public void setState(State state) {
        this.tvState=state;
    }
    public State getState() {
        return this.tvState;
    }
    public void doAction() {
        this.tvState.doAction();
    }
}
```

ד) יוצרים מחלקת Menu כדי לנהל את המצבים:

```
class Menu{
    private final State[] states = {new TVStartState(), new TVStopState(),
                                    new TVChannel_1_State(), new TVChannel_2_State()};
    Scanner sc = new Scanner(System.in);

    public State getState(){
        System.out.println("menu: 0 - start, 1 - stop, 2 - channel 1, 3 - channel 2,
                           any other number to exit \nget state:");
        int menu = sc.nextInt();
        if (menu > 3 || menu < 0){
            return null;
        }
        return states[menu];
    }
}
```

ו) שימוש ב-Context : רואים איך ההתנהגות שלו משתנה כאשר ה-State משתנה.

```
public static void main(String[] args) {
    TVContext context = new TVContext();
    Menu menu = new Menu();
    State s = new TVStartState();
    do{
        context.setState(s);
        context.doAction();
        s = menu.getState();
    }while (s != null);
}
```