# JS Patterns

## FOLDER STRUCTURE

```
FL16_HW10/ *                                          * - required
        ├── task
        ├── homework/ *
        │       ├── index.html *
        │       ├── index.js *
        │       ├── . eslintrc.json *
```

## TASK

You should create a tool to manage magazines.

1. Magazine
   a. A magazine should have possible 4 states: "ReadyForPushNotification", "ReadyForApprove", "ReadyForPublish", "PublishInProgress".
   b. Default state is "ReadyForPushNotification"
   c. a magazine will have staff, a list of articles for followers, and a list of followers.
2. Staff
   a. Should be at least one manager
   b. Only the manager can approve notifications
   c. Other employees can have different types. For instance, they can be responsible for "sport", "info", etc. Only they can add articles.
   d. Once the number of articles reached at least 5, the magazine's state becomes "ReadyForApprove" and a manager can approve the articles
   e. After approval, the magazine's state will become "ReadyForPublish", and each employee can publish the changes
   f. Once changes were published, the state becomes "PublishInProgress", and Followers will receive notifications from a magazine
   g. In 60 seconds the state will become again as default, and we will return to the start
3. Followers
   a. A follower should have a possibility to subscribe to the magazine and choose a topic (i.e. 'sport', 'info'), the topic might be the same as types of employee (see 2.c)
   b. A follower should have the possibility to unsubscribe from the changes

Frontend Lab

        c. For each object that represents a follower `onUpdate(data)` method should be present, where for `data` will be passed text of the notification.

4. For each magazine's state we will have different validations for method `approve` and `publish` (of course only if the employee has permissions to do it, otherwise in console should be `you do not have permissions to do it`)

        a. "ReadyForPushNotification"
           i. Publish: `Hello ${name}.  You can't publish. We are creating publications now.`
           ii. Approve: `Hello ${name}.  You can't approve. We don't have enough of publications.`

        b. "ReadyForApprove"
           i. Publish: `Hello ${name} You can't publish. We don't have a manager's approval.`
           ii. Approve: no validation, only `Hello ${name} You've approved the changes` in the console, and state must be changed

        c. "ReadyForPublish"
           i. Publish: no validation, only `Hello ${name} You've recently published publications.` in the console and state must be changed
           ii. Approve: `Hello ${name} Publications have been already approved by you.`

        d. "PublishInProgress"
           i. Publish: `Hello ${name}.  While we are publishing we can't do any actions.`
           ii. Approve: `Hello ${name}.  While we are publishing we can't do any actions`

**USAGE:**

```javascript
const magazine = new Magazine();
const manager = new MagazineEmployee('Andrii', 'manager', magazine);
const sport = new MagazineEmployee('Serhii', 'sport', magazine);
const politics = new MagazineEmployee('Volodymyr', 'politics', magazine);
const general = new MagazineEmployee('Olha', 'general', magazine);

const iryna = new Follower('Iryna');
const maksym = new Follower('Maksym');
const mariya = new Follower('Mariya');

iryna.subscribeTo(magazine, 'sport');
maksym.subscribeTo(magazine, 'politics');
mariya.subscribeTo(magazine, 'politics');
mariya.subscribeTo(magazine, 'general');
```

```
sport.addArticle('something about sport');
politics.addArticle('something about politics');
general.addArticle('some general information');
politics.addArticle('something about politics again');
sport.approve() //you do not have permissions to do it
manager.approve();//Hello Andrii. You can't approve. We don't have e
nough of publications
politics.publish(); //Hello Volodymyr. You can't publish. We are cre
ating publications now.
sport.addArticle('news about sport');
manager.approve(); //Hello Andrii. You've approved the changes
sport.publish(); //Hello Serhii. You've recently published publicati
ons.

/*
something about sport Iryna
news about sport Iryna
something about politics Maksym
something about politics Mariya
something about politics again Maksym
something about politics again Mariya
some general information Mariya
*/
manager.approve('news about sport'); //Hello Andrii. While we are pu
blishing we can't do any actions
```

**HINTS:**
— Use **State** **design pattern** to manage magazine's states
— Use **Observer** **(PubSub) pattern** for notification part
— Don't need to implement any UI changes, everything can be tested in the console

## RESTRICTIONS
- Adding **task/** folder is forbidden. Do not push it to repository. (Only **homework/** folder should be pushed)
- Do not use any external libraries

## BEFORE SUBMIT
- Code should be clean, without comments, readable, and tested

</> Frontend Lab

- Make sure your GitLab folder structure meets folder structure from this document (without **task** folder)
- Use linter :
    - In order to use npm package manager you should install nodejs (https://nodejs.org/ )
    - Install eslint to check your code (npm install -g eslint)
    - open a terminal (or cmd)
    - run eslint (i.e. eslint ./js/task1.js)

Code should be without 'errors'

## SUBMIT

- The **FL16_HW10** folder without **task** folder should be uploaded to GitLab repository "**FL-16**" into **master** branch.
-