Домашнее задание №1

Курс:
«Язык сценариев JavaScript и библиотека jQuery»

Модуль 11

Паттерны проектирования

Задание

Создать набор классов для обработки формы:

- **Rule** – описывает одно правило валидации одного элемента формы;
- **Logger** – логгирует результаты валидации элементов формы;
- **Validator** – валидирует элементы формы;
- **Processor** – обрабатывает элементы формы.

Таким образом, каждый этап валидации будет описан с помощью отдельного класса и у одного этапа может быть несколько реализаций (несколько классов) для разных ситуаций. Все эти этапы можно будет собирать воедино как конструктор и запускать с помощью **Processor**'а.

Например, класс **Validator** через конструктор принимает в качестве параметра объект класса **Logger**. При этом класс **Validator** не знает, как и куда логгер будет сообщать об ошибках, главное, чтобы у него был метод `log()` для оповещения об ошибках. Класс **Processor** через конструктор принимает объект валидатора и не знает, как именно валидатор делает проверку полей, главное, что у него есть метод `validate()`, с помощью которого можно запустить эту проверку.

Подробнее о классах.

Класс Rule необходим для описания правила валидации элемента формы. То есть один объект класса **Rule** – это одно правило для проверки одного элемента формы. Например, правило для ввода номера телефона будет проверять введеные данные с помощью регулярного выражения, а правило для ввода года рождения будет проверять входит ли введенное число в диапозон от 1900 до 2018.

Свойства:

- `name` – имя элемента формы (значение атрибута `name`);
- `errorText` – текст ошибки.

Метод:

- `isValid()` – проверяет корректны ли данные в элементе и возвращает логический результат.

Класс Logger необходим для описания логики информирования об ошибках.

Например:

- класс `ConsoleLogger` – вывод ошибок в консоль;
- класс `AlertLogger` – вывод ошибок с помощью `alert`;
- класс `DomLogger` – добавление ошибок в DOM и отображение их.

Метод:

- `log(errorText)` – информирует об ошибке (способ информирования зависит от конкретного логгера).

Домашнее задание №1

Класс Validator имеет доступ к форме, которую необходимо проверить, и набор правил, с помощью которых ее проверить.

Свойства:

- `logger` – объект любого логгера, у которого есть метод `log` для информирования об ошибках;
- `rules` – массив правил валидации;
- `form` – форма, которую необходимо проверять.

Метод:

- `validate(form)` – метод проходится по всем элементам формы и проверяет их в соответствии с правилами из массива `rules`; если хоть одна проверка вернула `false`, то весь метод возвращает `false`, иначе – `true`.

Класс Processor связывает валидатор и форму, а также знает какую функцию необходимо вызвать в случае успешной валидации. Например, если пользователь правильно ввел логин и пароль, то в функции может происходить запись пользователя в куки и перенаправление на другую страницу.

Свойства:

- `validator` – объект валидатора, который описывали выше;
- `success` – функция, которая будет вызвана при успешном прохождении валидации.

Метод:

- `attach(formSelector)` – метод принимает селектор, по которому можно найти форму в DOM, находит ее и привязывает валидатор на событие `onsubmit`; в случае успешного прохождения валидации, вызывает функцию `success`.

Как применить?

1. Описать форму со следующими элементами:
   - имя пользователя (от 5 до 15 символов, только буквы);
   - год рождения (от 1900 до текущего);
   - цвет глаз (одно из значений: brown, green, gray, blue);
   - цвет волос (одно из значений: black, brown, white, red, other);
   - рост (от 0 до 2.60);
   - вес (от 0 до 300).
2. Описать правила для каждого элемента формы.
3. Создать три логгера (`ConsoleLogger`, `AlertLogger`, `DomLogger`) и описать логику информирования об ошибке.
4. Описать класс **Validator** и создать его объект, передав необходимые параметры.
5. Описать класс **Processor** и создать его объект, передав необходимые параметры.
6. Одним из параметров при создании объекта типа **Processor** будет функция, которая должна быть вызвана при успешной валидации. Такая функция должна выводить все введеные данные в блоке под формой и очищать элементы формы.
7. Вызвать метод `attach` у объекта класса **Processor**.
8. Открыть в браузере страницу с формой и проверить, как происходит валидация.