УДК 004.432.2

УНИВЕРСАЛЬНЫЙ NULL OBJECT НА JAVASCRIPT

К.И. ЛАПКОВСКИЙ
(Представлено: Д.В. ПЯТКИН)

Статья посвящена использованию паттерна null object. Рассмотрим цели использования этого паттерна в приложениях. Познакомимся со средствами, необходимыми для реализации поставленной задачи. В результате будет реализован универсальный null object на javascript.

Введение. В объектно-ориентированном программировании null object – это объект с определенным нейтральным («null») поведением [1]. Есть несколько причин использования null object:

1. Защита от null pointer exception. Своя версия null pointer exception есть во многих современных языках программирования: Java, JavaScript, C#. Такая ошибка появляется при обращении к свойствам или методам переменной, значение которой равно null. null object реализует интерфейс объекта, тем самым подстраиваясь под полноценный инстанс класса.
2. Предоставление дефолтного (нейтрального) поведения. Часто под нейтральным поведением понимают его отсутствие, то есть метод который необходимо реализовать просто оставляют пустым.

Основная проблема null object – его реализация. Под каждый интерфейс необходимо написать свой null object. В результате количество классов в системе вырастает пропорционально используемых абстракций.

Универсальный null object разработан для решения первой задачи. Основное преимущество универсального null object – способность подстраиваться под любой интерфейс.

Основной раздел. Для реализации универсального null object необходимо познакомиться с классом, который входит в стандартное окружения браузера – Proxy. Объект Proxy «оборачивается» вокруг другого объекта и может перехватывать (и, при желании, самостоятельно обрабатывать) разные действия с ним, например чтение/запись свойств и другие [2]. Работа с проксируемым объектом построена на концепции ловушек. Если необходимо контролировать все обращения к полям проксируемого объекта, необходимо реализовать get-ловушку, также есть ловушки для вызова методов, удаления/записи свойств и т.д.

Реализация перехвата получения значения свойства объекта

```
const proxy = new Proxy({}, {
  get(target, prop, receiver) {
    return receiver
  }
});
```

Листинг 1. – Реализация перехвата получения значения свойства объекта

Ловушка получения значения свойства объекта принимает три параметра: target – проксируемый объекта, в нашем случае это пустой объект ({}), prop – название свойства, значение которого необходимо получить, receiver – это либо сам proxy объект, либо его наследник. [MDN] Как видно из листинга 1, любое обращение к любому свойству объекта возвращает proxy объект.

Для проверки работоспособности попробуем обратиться к вложенным полям proxy объекта.

Листинг 2 – проверка реализации перехвата получения значения свойства объекта

```
proxy.a.b.c
```

Ошибка TypeError: Cannot read property, аналог null pointer exception в javascript, не выбросилась, значит универсальный null object реализован.

Заключение. В статье рассмотрен пример реализации универсального null object на javascript. Такой инструмент может заменить множество уже существующих null object, что упростит структуру программы уменьшив количество повторяющегося кода.

ЛИТЕРАТУРА

1. Wikipedia [Электронный ресурс] Null Object (шаблон проектирования). Режим доступа: https://ru.wikipedia.org/wiki/Null_object_(шаблон_проектирования). Дата доступа: 26.09.2019.
2. The Modern JavaScript tutorial [Электронный ресурс] Proxy и Reflect. Режим доступа: https://learn.javascript.ru/proxy. Дата доступа: 26.09.2019.
3. MDN web docks [Электронный ресурс] hander.get. Режим доступа: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Proxy/handler/get. Дата доступа: 26.09.2019