Введение Углублённая теория Методика В Алгоритм выделения блоков Типовые интерфейсные решения Блоки и элементы Миксы блоков и элементов Модификаторы блоков и элементов № Кейс 1, лёгкий уровень ☑ Кейс 2, лёгкий уровень ☑ Кейс 3, лёгкий уровень

☑ Кейс 4, средний уровень ☑ Кейс 5, средний уровень Типовые ошибки

Перечислим типовые ошибки, которые совершают разработчики, когда используют БЭМ.

## Ошибка 1

Определяется неверный уровень абстракции. Что это значит? Мы находим общие свойства у блоков, которые не очень похожи. Добавляем условия, чтобы скрыть лишние и показать нужные свойства у блока. Много переопределений стилей. Когда мы используем принципы KISS (Keep It Simple, Stupid) и DRY (Don't Repeat Yourself), возникает когнитивная ошибка, и мы видим общее там, где его нет. Плохая абстракция приводит к мудрению в коде. Это попытка объединить в один БЭМ-блок разные сущности. Не факт, что такая композиция приживётся. Лучше сделать несколько отдельных компонентов, чем объединять необъединимое и «подставить костыли». Пусть поначалу будет дублирование в коде, пусть мы создадим похожие блоки, как разные компоненты. Потом, если

🙀 Виталий Ризун 🔻

Смотреть все

появится необходимость, можно отрефакторить, увидев понятную абстракцию. К примеру, на странице есть такие блоки:

Новинки

Палатка 4-местная Палатка 2-местная Палатка 2-местная Палатка 1-местная 4700₽ 5500₽ 2600₽ 9900₽ В корзину В корзину В корзину В корзину Блог Все статьи Выбор первой палатки Минимальная экипировка Составляем идеальный маршрут Разбираемся, как выбрать самый Как быстрее всего попасть из точки А в Есть вещи, без которых лучше не уходить подходящий вариант и не переплатить. точку Б и не заблудиться по пути. далеко от дома. Читать статью Читать статью Читать статью Страница лендинга с повторяющимися блоками Есть несколько блоков, которые выглядят для нас одинаково, хотя на самом деле это не так. Интуитивно хочется реализовать их как один блок, но, если внимательно присмотреться, то схожими будут только шапки блоков, вот их можно выделить в один компонент.

<section class="catalog"> <header class="catalog\_headline headline"> <h2 class="headline title">Новая коллекция</h2>

<a class="headline\_\_link link">Смотреть все</a>

</header> 

</section> <section class="blog"> <header class="blog\_headline headline"> <h2 class="headline\_\_title">Блог</h2> <a class="headline\_\_link link">Все статьи</a> </header> </section> У блока присутствует внешняя геометрия (внешние отступы, позиционирование, ширина жёстко зафиксирована). <section class="section section--catalog">

А надо так:

</a>

</div>

</div>

<!-- Ошибка в имени класса -->

<div class="company-logo\_\_cat"></div>

выражения выполнен, а план содержания — нет.

<!-- Очень линейное БЭМ-дерево -->

<div class="company-logo\_\_text">

<a class="menu-item link">

/\* такое использование модификаторов — ошибка \*/ .section--catalog {

```
margin-bottom: 132px;
 .section--novelty {
   margin-bottom: 324px;
Геометрия, в том числе и внешние отступы, могут применяться только к элементам блока.
Ошибка 3
Неверная декомпозиция на блоки. Не видно БЭМ-дерева, почти везде просто блок, вложенный в другой блок,
и всё живёт просто благодаря тому, что имена разные. Но это не декомпозиция, это не БЭМ. Например menu-item
> menu-link > menu-text — это отдельные блоки?
 class="menu-item">
```

<span class="menu-item\_\_text">Кабинет ученика</span>

<span class="company-logo\_\_text\_\_first-letter">C</span>at

А надо так: <div class="company-logo">

<h2 class="sources\_\_title">Наши партнёры/Источники финансирования</h2>

```
<span class="company-logo__first-letter">C</span>at
   </div>
 </div>
Ошибка 5
Неверное подчинение. Очень линейное БЭМ-дерево, один блок и куча элементов, нет попыток разбить, сделать
декомпозицию. Например, пишут sources > sources_logos > sources_logo, а могли бы sources_logos
сделать отдельным блоком, но не стали. К БЭМ не стоит подходить формально. Когда синтаксис соблюдён, но при
```

<!-- Видим и выделяем более мелкие блоки и составляем из них композицию --> <section class="sources"> <h2 class="sources\_\_title">Наши партнёры/Источники финансирования</h2> 

<img class="logo\_\_img" src="img/logo-1.svg" width="50" height="20" alt="">

```
</section>
 - План выражения и план содержания
 План выражения и план содержания — это из семиотики, науки о знаках. У любого знака, будь это
 картинка, символ или другой законченный знак, есть план содержания (что это значит, как это нужно
 понимать, какие оттенки и нюансы) и план выражения (это прозвучало-написано-нарисовано, как именно,
 насколько оно составное и так далее). По-другому это называется обозначаемое и обозначающее.
```

А надо так: <a class="button button--primary" href="#">Смотреть каталог</a> Ошибка 7

Элементы без блока, которые болтаются сами по себе. Элемент обязательно должен находиться внутри блока,

## <div class="logo"></div> </div> А надо так:

Ошибка 9

Ошибка 11

40</a>

</footer>

<input class="search-form\_\_input">

<button class="search-form button</pre>

<form class="search-form search-form\_theme\_lite">

<input class="search-form\_\_input">

Найти

</button>

Найти

</button>

Например:

Ошибка 15

</section>

Ошибка 16

</form>

search-form\_\_button\_size\_s

search-form\_\_button\_size\_m">

<button class="search-form\_\_button search-form\_\_button\_size\_s">

<div class="page-header">

<div class="page-header\_\_logo">

<div class="logo"></div>

Ошибка 6

класс для блока или элемента!

<a class="button--primary" href="#">Смотреть каталог</a>

Ошибка 8 Название классов включают в себя внешний вид: button--red (кнопка красная), footer--dark (подвал тёмный) и подобные. Исключение — button--size-xl. Не должно быть блоков с именами uppercase, text-center — это

стили вне блоков. Нежелательно использовать классы вида vertical-wrap или push-md--right. Стили должны

всегда относиться к конкретному блоку. А что такое push-md--right? Это блок такой? Нет.

Используются классы из списка запрещённых имён: for , list\_\_item1 , list\_\_item2 и другие.

href="mailto:info@edisonshoes.com">info@edisonshoes.com</a>

<!-- для имени модификатора используется стиль CamelCase -->

```
Разнобой в синтаксисе или нотации. Выбираем что-то одно, к примеру, разделение блок__элемент - -модификатор.
 <footer class="page-footer container">
  <!-- для имени модификатора используется классический стиль -->
      <a class="contacts__link contacts__link_phone" href="tel:88009995522">8 (800) 999 55 22</a>
    <!-- для имени модификатора используется стиль Two Dashes -->
      <a class="contacts__link contacts__link--mail"</pre>
```

<a class="contacts\_\_link contacts\_\_link-mapPin" href="#">Санкт-Петербург, ул. Набережная, д.

```
<!-- Блок `search-form` имеет модификатор `theme` со значением `islands` -->
 <form class="search-form search-form_theme_islands">
   <input class="search-form__input">
   <!-- Элемент `button` имеет булевый модификатор `disabled` -->
   <button class="search-form__button search-form__button_disabled">Сбросить</button>
   <!-- Элемент `button` имеет модификатор `size` со значением `m` -->
   <button class="search-form__button search-form__button_size_m">Hайти</button>
 </form>
Ошибка 12
Ошибкой будет одновременно использовать два одинаковых модификатора с разными значениями.
 <form class="search-form"</pre>
               search-form_theme_islands
               search-form_theme_lite">
```

Сложные модификаторы, например, вида article\_\_item--1of4 стоит писать в виде «ключ\_значение».

Ошибка 13

```
<footer class="footer">
   <span class="footer--brand">Elenea &copy;</span>
   <span class="footer--description">Copyright 2021
 </footer>
А надо так:
 <footer class="footer">
   <span class="footer__brand">Elenea &copy;</span>
   <span class="footer__description">Copyright 2021</span>
 </footer>
```

```
<h2 class="section__title">Контакты</h2>
<a class="section__icon" href="mailto:company@mail.com">company@mail.com</a>
```

class="tabs\_\_navigation-item">Характеристики class="tabs\_\_navigation-item">Отзывы </div> </body>

Блок хочет управлять CSS-свойствами вложенных блоков. К примеру, при использовании светлой темы цвет

текста на панели с закладками должен быть #2e2c9c, а фоновый цвет вкладок #eeeeee.

```
Устанавливаем цвет шрифта для закладок:
 .theme--light .tabs {
   color: #2e2c9c;
Задаём фоновый цвет самим вкладкам:
 .theme--light .tabs__navigation-item {
   background-color: #eeeeee;
Используется межблочный каскад, что категорически не рекомендуется делать. Это делает код разных блоков
```

Ознакомились со статьёй? Сохранить прогресс

Как можно применить параметры темы для внутренних блоков и при этом не использовать межблочный каскад,

```
Блог
Практикум
                               Курсы
                              HTML и CSS. Профессиональная вёрстка сайтов
Тренажёры
                                                                                              С чего начать
                              HTML и CSS. Адаптивная вёрстка и автоматизация
                                                                                              Шпаргалки для разработчиков
Подписка
Для команд и компаний
                                                                                              Отчеты о курсах
                               JavaScript. Профессиональная разработка веб-интерфейсов
Учебник по РНР
                               JavaScript. Архитектура клиентских приложений
```

Кейс 6, средний уровень ☑ Кейс 7, сложный уровень ☑ Кейс 8, сложный уровень 🖸 Кейс 9, сложный уровень

Ошибка 2 К примеру, для того чтобы попасть в *PerfectPixel* добавляют странные модификаторы для блоков. </section> <section class="section section--novelty"> </section>

<a class="menu-link"> <span class="menu-text">Кабинет ученика</span> </a> 

Ну конечно же нет, но выглядит, как будто да. Всё разбито на никак не связанные между собой блоки. Там, где

должны быть элементы блока — блок\_\_элемент, написано просто два отдельных несвязанных блока.

Ошибка 4 Попытка использовать элементы элементов, к примеру, так: <div class="company-logo"> <div class="company-logo\_\_cat"></div> <div class="company-logo text">

этом композиционно решено неверно, зависимости указаны неправильно, много связей и другое. План

<section class="sources">

```
<img class="sources__img" src="img/logo-1.svg" width="50" height="20" alt="">
   </section>
А надо так:
```

Термин ввёл Фердинанд де Соссюр.

Модификатор сам по себе, не относится ни к блоку ни к элементу. А у модификатора обязательно должен быть

но элементы могут находиться «друг в друге» в разметке в рамках одного блока. <!-- Ошибка --> <div class="logo\_\_container">

</div> </div>

```
Ошибка 10
```

```
</form>
Добавление дополнительных имён классов-блоков, а не модификаторов. Чтобы добавить визуальное отличие для
базового блока делают его композицией из двух блоков, например: class="section support", class="section
advantages".
А надо так: class="section section--support", class="section section--advantages".
Но классы также могут примиксоваться и как блоки, всё зависит от ситуации.
Ошибка 14
Использование модификаторов вместо элементов. Модификатор — это всегда только внешнее изменение для
блока/элемента.
```

В этом примере мы не можем использовать одновременно модификаторы search-form\_theme\_islands и search-

form\_theme\_lite, search-form\_\_button\_size\_s и search-form\_\_button\_size\_m. Нужно выбрать что-то одно.

в одном месте, в конкретном блоке. Стили сменятся во всех блоках, где их используют. <!-- Вот такой микс — сложно. Блоки не должны хотеть «вплетаться друг в друга». Исходя из такой разметки не считывается один момент — как их использовать по отдельности и где тут внутри чья зона ответственности? --> <section class="contacts section">

Формально это не ошибка, но будут проблемы, когда компонуют блоки так: list-item > contacts\_\_link >

section\_\_icon . Так-то можно «взболтать, но не смешивать», однако при этом не получится поменять стили только

```
<body class="page</pre>
  theme
  theme--light">
  <div class="tabs">
```

связанным.

**К** Модификаторы блоков и элементов

расскажем в навыке «Компонентная вёрстка по БЭМ».

Профессии Фронтенд-разработчик React-разработчик Фулстек-разработчик Бэкенд-разработчик Услуги Работа наставником Для учителей

Стать автором

React. Разработка сложных клиентских приложений РНР. Профессиональная веб-разработка PHP и Yii. Архитектура сложных веб-сервисов Node.js. Разработка серверов приложений и API Анимация для фронтендеров Вёрстка email-рассылок

Vue.js для опытных разработчиков

Алгоритмы и структуры данных

Шаблонизаторы HTML

Анатомия CSS-каскада

Регулярные выражения для фронтендеров

Остальное Написать нам Мероприятия Форум

Информация

О центре карьеры

Об Академии

Кейс 1, лёгкий уровень 🕻