

**М. А. Петруша**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. В. Андреев**, д-р физ.-мат. наук, профессор

## **РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ НА ЯЗЫКЕ JAVASCRIPT**

Благодаря паттерну проектирования можно документировать решения, которые будут приняты на основе предыдущей практики и извлеченных уроков. При реализации модели проектирования используется несколько программных компонентов. Таким образом, модель ускоряет процесс, в котором задействовано несколько компонентов. Разработчики используют знакомый им язык для применения каждого решения.

Польза от использования паттерна проектирования очевидна. Шаблон проектирования позволяет ускорить процесс разработки. Он предоставляет проверенные парадигмы разработки, что позволяет сэкономить время и не изобретать шаблоны каждый раз, когда возникает проблема.

Согласно [1] паттерн "Команда" (Command) является поведенческим шаблоном проектирования, который инкапсулирует запрос как объект, позволяя параметризовать клиентов с разными запросами, очередями и отменой операций. Это позволяет поддерживать дополнительные операции без изменения клиентского кода.

Применение:

- для параметризации объектов выполняемым действием;
- для того чтобы ставить операции в очередь, выполнять их по расписанию или передавать по сети;
- для добавления операции отмены.

«Итератор» – это поведенческий паттерн проектирования, который даёт возможность последовательно обходить элементы составных объектов, не раскрывая их внутреннего представления.

Применение:

- при необходимости создания сложной структуры данных, при условии, что надо скрыть от клиента детали её реализации (из-за сложности или вопросов безопасности);
- при необходимости иметь несколько вариантов обхода одной и той же структуры данных.

Согласно [2] одна из важнейших особенностей языка JavaScript – это объектная ориентированность. Программистам доступно огромное количество объектов, таких, как формы, фреймы, гиперссылки, и т. д. Объекты характеризуются свойствами и методами.

В основном JavaScript активно применяется не только во front-end, но и в back-end разработке. На JavaScript разрабатываются приложения, исполняющиеся не только на стороне клиента, но и на серверной стороне. Согласно [3] это стало возможно благодаря node.js. Node.js является платформой с открытым исходным кодом, основанная на движке V8, позволяющая писать серверную часть веб-приложения, а также программ командной строки. Вся интерактивность на сайте (например: всплывающее меню), обычно обеспечивается кодом, написанным на JavaScript.

JavaScript высокоуровневый язык программирования, позволяющий разработчику создавать интерфейсы, которые будут обрабатываться асинхронно.

Разработчики могут использовать различные фреймворки JavaScript для разработки и создания веб-приложений и мобильных приложений. Фреймворки JavaScript – это наборы библиотек кода JavaScript, которые предоставляют разработчикам предварительно написанный код для использования в рутинных функциях и задачах программирования – буквально каркас для создания веб-сайтов или веб-приложений.

Для того чтобы правильно применить Command паттерн и иметь возможность отмены предыдущих операций мы должны создать методы resetMemory(), changeMemory(), resetValue(), executeCommand() и undo() (рисунок 1 (а)).

```
import { MEM_VALUE } from './contains'

export class Calculator {
  constructor() {
    this.value = 0
    this.history = []
    this.memory = 0
  }

  resetMemory() {
    this.memory = 0
    MEM_VALUE.innerHTML = this.memory
  }

  changeMemory(value) {
    this.memory += value
    MEM_VALUE.innerHTML = this.memory
  }

  resetValue(value) {
    this.value = value
  }

  executeCommand(command) {
    this.history.push(this.value)
    this.value = command.execute(this.value)
  }

  undo() {
    this.resetValue(this.history.pop())
  }
}
```

а

```
import { CALCULATOR, SCOREBOARD } from './contains'

export class AddCommand {
  constructor(valueToAdd) {
    this.valueToAdd = valueToAdd
  }

  execute(currentValue) {
    return currentValue + this.valueToAdd
  }
}

export class SubtractCommand {
  constructor(valueToSubtract) {
    this.valueToSubtract = valueToSubtract
  }

  execute(currentValue) {
    return currentValue - this.valueToSubtract
  }
}

export class DivideCommand {
  constructor(valueToDivide) {
    this.valueToDivide = valueToDivide
  }

  execute(currentValue) {
    return currentValue / this.valueToDivide
  }
}

export class MultiplyCommand {
  constructor(valueToMultiply) {
    this.valueToMultiply = valueToMultiply
  }

  execute(currentValue) {
    return currentValue * this.valueToMultiply
  }
}
```

б

Рисунок 1 – (а) Реализация класса калькулятора на языке TypeScript.  
(б) Реализация классов для операций

Далее необходимо создать собственный класс для каждой математической операции (рисунок 1б). Каждый класс должен иметь конструктор который будет принимать значение введенное пользователем и метод executeCommand().

В результате получаем рабочее приложение калькулятора, в котором помимо математических операции есть возможность сохранять предыдущие операции и по нажатию на кнопку 'Back' вернуть сохраненное значение (рисунок 2).

Также в приложение можно добавить возможность менять тему. Как следует из [4] для этого требуется создать JavaScript функцию, которая будет менять css стили всего приложения, например, цвет символов и фона (рисунок 2).

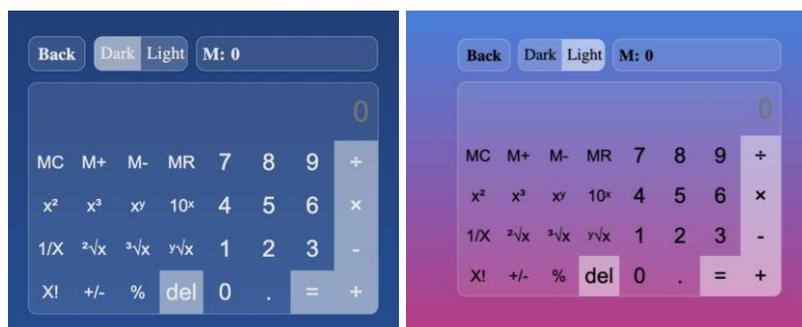


Рисунок 2 – Пример готового web-приложения с демонстрацией изменения темы

В заключение можно сказать, что паттерны проектирования представляют собой ценный инструмент разработки программного обеспечения, позволяя создавать гибкие, удобные для поддержки и расширяемые системы. Рассмотренные паттерны «Команда» и «Итератор» предоставляют эффективные решения для определенных типов задач, а их сочетание может дополнительно улучшить структуру кода.

## Литература

1. Современный учебник JavaScript [Электронный ресурс] / И. Кантор: электрон. версия газ. – 2023. – 22 марта. – URL: <https://learn.javascript.ru>. – Дата доступа: 22.03.2023.
2. You Don't Know JS [Электронный ресурс] / К. Simpson: электрон. версия газ. – 2023. – 22 марта. – URL: <https://github.com/azat-io/you-dont-know-js-ru>. – Дата доступа: 22.03.2023.
3. Resources for Developers [Электронный ресурс]: электрон. версия газ. – 2023. – 22 марта. – URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript>. – Дата доступа: 22.03.2023.
4. Flanagan, D. JavaScript The Definitive Guide / D. Flanagan, 2011. – O'Reilly Media. 6th edition. – 1096 с.

**В. И. Пилипчук, В. С. Чуль**

(БрГТУ, Брест)

Науч. рук. **Н. Н. Ворсин**, канд. физ.-мат. наук, доцент

## МОДЕЛИРОВАНИЕ МАГНИТНОГО ПОЛЯ В ЛАБОРАТОРНОЙ УСТАНОВКЕ

В лабораторной работе «Измерение удельного заряда электрона» используется электронная лампа с цилиндрическими электродами (в нашем случае 2Ц21П), находящаяся под действием осевого магнитного поля. Это поле действует на электроны, летящие радиально от катода к аноду, и искривляет их траекторию. Если индукция поля превышает некую величину, названную критической, искривление траектории становится таким, что электроны не достигают анода, закручиваясь в межэлектродном пространстве. Это приводит к обнулению анодного тока. Если индукцию поля увеличивать постепенно, контролируя при этом анодный ток лампы, то можно определить величину критической индукции по резкому уменьшению анодного тока. Далее полученная величина критической индукции вместе с размерами электродов лампы подставляются в формулу, которая выражает удельный заряд электрона – отношение заряда к массе.

Описанная схема лабораторной работы очень популярна и, судя по публикациям, используется, наверно, во всех физических лабораториях ВУЗов, например [1]. На практике оказывается, что никакого резкого скачка анодного тока не наблюдается, а по плавно спадающему графику его зависимости от тока катушки, создающей магнитное поле, точно определить критическую величину индукции не удаётся.

Мы считаем, что имеются несколько причин отличия экспериментально и теоретически полученных зависимости анодного тока лампы от тока питания катушки, создающей магнитное поле. Одна из них – неоднородность магнитного поля внутри катушки в области расположения электродов лампы. Вследствие этого момент достижения полем критической величины оказывается разным для различных участков анода, что «размазывает» скачок анодного тока. Если иметь реальное распределение индукции поля вдоль оси катушки, то можно правильным расположением лампы и конструкции катушки ослабить этот эффект размазывания. Наиболее простым методом получения зависимости индукции поля катушки от осевой координаты является компьютерное моделирование этого поля.