

Excel, PowerPoint, Corel Draw, Photoshop). Системы автоматизированного проектирования общего назначения (AutoCad, P-CAD и др.), узкоспециализированные САДы и моделирующие программы, математические продукты (MathCAD, MatLAB), системы программирования (Delphi, Builder C++ и др.), разработка баз данных.

Необходимость использования указанных программных продуктов обучаемыми и преподавателями обуславливает формирование определенных профессиональных компетенций у теперешних студентов – будущих преподавателей-программистов, включающих знания и умения в области анализа предметной области и разработки требований к создаваемым программным средствам и системам; эскизного проектирования программных средств и систем; технического проектирования программных средств; кодирования программных средств; тестирования, верификации и аттестации программных средств; измерения, оценки качества программных средств; эксплуатации программных средств, компьютерных систем и сетей; теоретических и экспериментальных исследований, связанных с разработкой, совершенствованием или оценкой программных средств и др.

Без знаний, умений и навыков, полученных при изучении дисциплины «Прикладное программное обеспечение» представить современного преподавателя компьютерных дисциплин (педагога-программиста) не возможно.

УДК 004.42

Пашковский В. И.

## **JAVASCRIPT НА NODE. JS ИЛИ PHP**

*Белорусский национальный технический университет,*

*г. Минск, Республика Беларусь*

*Научный руководитель: ст. преподаватель Астапчик Н. И.*

Языки программирования PHP и JavaScript раньше были союзниками и каждый из них занимал свою сферу. Каждый из них был необходим для нормальной работы сайтов. Сферы назначения языков были явно разделены, JavaScript работал на стороне пользователя в браузере, в то время как PHP выполнял все серверные задачи. По такому принципу сейчас работает множество сайтов и систем управления, таких как WordPress, Drupal и Facebook.

Но сейчас ситуация немного изменилась. Для создания приложений следующего поколения больше нет необходимости использовать PHP. Все что нужно – это Node.js и поддержка JavaScript в браузере. Некоторые разработчики уже прониклись идеей JavaScript везде.

JavaScript – это инструмент взаимодействия с HTML и CSS. HTML задает разметку сайта, CSS отвечает за внешний вид, а JavaScript все это оживляет. С помощью кода на JavaScript программист определяет, как страница отреагирует на действия пользователя также определяет динамику и поведение элементов.

Сейчас JavaScript – единственный язык программирования для браузеров. Он работает под Windows, macOS, Linux и на мобильных платформах, то есть везде.

JavaScript достаточно прост, но содержит все фундаментальные вещи: алгоритмы, объектно-ориентированную модель, структуры данных, замыкания, Prototype-наследование.

Начинать с JavaScript хорошо потому, что синтаксически он похож на C. Изучив JavaScript, получишь базовое представление обо всех «сиобразных» языках: C++, C#, Java, PHP. Они задают тренд в своих областях и весьма популярны, поэтому для новичка важно познакомиться с синтаксисом C.

Классический JavaScript – это язык программирования для интернета, он бессилен за пределами браузера. С помощью JavaScript нельзя запустить программу на компьютере или записать файл в нужную папку.

Из-за правил безопасности браузеры ограничивают мощь JavaScript за пределами страниц которыми он управляет. Управлять вкладками можно при определенных условиях или же вовсе нельзя. Например, JavaScript может закрыть только ту вкладку, которую создал сам.

Node.js, вывел JavaScript за пределы браузеров, дав программистам возможность писать код для десктопных приложений. А также запускать код на JavaScript на своём компьютере и других машинах в качестве отдельных приложений.

Необычно и то, что платформа имеет встроенные библиотеки для обработки запросов и ответов, тем самым не нужно использовать сторонний веб-сервер и какие-либо другие зависимости такие как язык PHP.

PHP специально предназначен для программирования на стороне сервера, что означает, что его библиотека специализирована для задач, которые разработчик будет делать снова и снова в ходе программирования сайта. PHP также может переплетаться с HTML кодом что дает ему огромное преимущество, таким образом позволяя смешивать верстку с программированием. PHP доступен для большинства операционных систем, включая Unix и Windows, а также является отличным языком программирования на стороне сервера для профессионального программирования. PHP является одним из наиболее широко используемых языков для веб-разработки. PHP - это встраиваемый в HTML скриптовый язык, который может быть использован для того чтобы своевременно сформировать динамические страницы. С помощью PHP можно очень быстро и легко расширить веб-приложение. PHP код может быть непосредственно размещен в HTML коде, без использования внешнего файла.

Не смотря на то что PHP и Node.js могут справляться с приложениями любой сложности, они созданы на основе разных концепций и архитектур. Если разработчик будет выбирать между двумя платформами, он должен иметь в виду их основные преимущества и недостатки.

### **Достоинства Node.js:**

1) Один язык на фронтенде и бекенде. Используя Node.js на сервере разработчик получает все преимущества скриптового языка на обеих платформах. Иметь один язык на фронтенде и бекенде очень хорошо для обеспечения поддержки приложения и координации между членами большой команды.

2) Гибкость. Node.js не имеет строгих правил или жестких зависимостей, что оставляет простор для креативности при разработке приложений. Разработчики сами выбирают архитектуру, зависимости.

### **Недостатки Node.js:**

1) Малая эффективность в операциях интенсивно использующих центральный процессор. Событийно-ориентированная архитектура Node.js имеет некоторые ограничения а именно низкую эффективность при большой нагрузке на процессор. Хотя и Node хорошо справляется с конкурентной обработкой множества запросов, он все же плохо справляется с генерированием графики, обработкой изображений.

2) Незрелость платформы. Вместе со стабильными стандартными библиотеками, доступно множество библиотек от сторонних

разработчиков, поэтому приходится зависеть от не очень качественно протестированных модулей (библиотек).

### **Достоинства PHP:**

1. Большая кодовая база (много библиотек). PHP имеет огромную кодовую базу для всевозможных решений от систем управления контентом до мощных фреймворков таких как Laravel, Symfony. Например с помощью WordPress можно запустить свой блог за считанные минуты.

2. Переносимое решения. PHP достаточно независим от платформы, он может быть запущен почти на любом сервере и на любой платформе в отличие от JavaScript, которому нужен отдельный сервер настроенный под Node.js.

В тоже время, имеется широкий выбор хостингов которые с поддержкой PHP поэтому не нужно арендовать полноценный сервер с SSH доступом чтобы запустить проект, в отличии от Node.js. Из этого следует, что интеграция PHP проектов и их развертывание несколько проще для небольших компаний или для отдельных лиц который могут запускать и управлять своими приложениями без знаний консольных команд, системы Linux

3. Спроектирован для WEB.В отличии от Node.js , PHP был разработан специально для Web. Именно поэтому он содержит всю необходимую функциональность для работы с HTML, серверами, базами данных. С всеобъемлющим языком PHP, в большинстве случаев можно обойтись минимальным количеством JS кода на фронтенде.

### **Недостатки PHP:**

1) Устаревшая клиент-серверная модель. PHP следует классической клиент-серверной модели. Это делает PHP несколько медленнее в сравнении с Node.js приложениями, которые иницируются при запуске, и поэтому Node больше подходит для написания приложений реального времени.

2) PHP используется только для разработки серверной части приложений. Это означает что разработчику нужно знать Javascript, если он хочет работать на стороне клиента или стать fullstack-разработчиком.

Итак, при выборе между JavaScript на Node.js и PHP, разработчик должен обязательно определить, какой тип приложения он собирается написать, каковы требования хостинга клиента, и какова скорость разработки ему нужна. Если клиент ищет мультиплатформ-

менную поддержку хостинга разработчик должен рассмотреть возможность использования РНР. С другой стороны, Node.js-лучший вариант для разработки приложений реального времени, таких как чаты и SPA (одностраничные приложения) так как его клиент-серверная модель не является устаревшей.

УДК 378.6

Пачишева В. А., Бруй А. И.

**ПРИМЕНЕНИЕ КЕЙС-ТЕХНОЛОГИЙ  
ПРИ ИЗУЧЕНИИ ДИСЦИПЛИНЫ «ТЕОРИЯ РЕЗАНИЯ  
И РЕЖУЩИЕ ИНСТРУМЕНТЫ»**

*Белорусский национальный технический университет,  
г. Минск, Республика Беларусь*

*Научный руководитель: ст. преподаватель Игнаткович И. В.*

Одной из новых форм эффективных технологий обучения является проблемно-ситуативное обучение с использованием кейсов. Внедрение учебных кейсов в практику образования в настоящее время является весьма актуальной задачей. Кейс-технология (метод) обучения – это обучение действием, направленное на самостоятельную индивидуальную и групповую деятельность студентов. Кейс-метод широко используется в обучении за рубежом. Впервые он был применён в учебном процессе на факультете права Гарвардского университета.

Кейс представляет собой описание конкретной реальной ситуации, подготовленное по определенному формату и предназначенное для обучения обучающихся анализу разных видов информации, ее обобщению, навыкам формулирования проблемы и выработки возможных вариантов ее решения в соответствии с установленными критериями. Зачастую кейс-технологии применяются на лекциях у одной группы так как она содержит небольшую численность студентов [1].

Сущность кейс-технологии состоит в том, что учебный материал подается студентам виде проблем (кейсов), а знания приобретаются в результате активной и творческой работы: самостоятельного осуществления целеполагания, сбора необходимой информации, ее анализа с разных точек зрения, выдвижения гипотезы, выводов, заключения, самоконтроля процесса получения знаний и его результатов.