

Московский Авиационный Институт
(Национальный Исследовательский Университет)

**Кафедра вычислительной математики и
программирования**

Реферат на тему «ООП на языке JavaScript»

По курсу «Фундаментальная информатика»

II семестр

Выполнил студент

1-го курса, 105-ой группы

Махмудов О. С.

(подпись)

Научный руководитель

Доцент кафедры 806

Сластушенский Ю. В.

(подпись)

Работа защищена

«__»_____ 2019

Оценка_____

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ИСТОРИЯ	3
ОСОБЕННОСТИ WEB-ПРОГРАММИРОВАНИЯ	4
Серверные приложения	5
Связь с использованием AJAX	6
Клиентские (одностраничные) приложения	6
ЯЗЫК JAVASCRIPT	6
Области применения.....	6
Возможности	7
Связь с Java	8
Библиотеки.....	8
Синтаксис в сравнении с языком C/C++	9
Отладка.....	13
Безопасность	13
ЗАКЛЮЧЕНИЕ	14
СПИСОК ЛИТЕРАТУРЫ.....	14

ВВЕДЕНИЕ

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

ИСТОРИЯ

В 1992 году компания Nombas (впоследствии приобретённая Openwave) начала разработку встраиваемого скриптового языка Smm (Си-минус-минус), который, по замыслу разработчиков, должен был стать достаточно мощным, чтобы заменить макросы, сохраняя при этом схожесть с Си, чтобы разработчикам не составляло труда изучить его. Главным отличием от Си была работа с памятью. В новом языке всё управление памятью осуществлялось автоматически: не было необходимости создавать буфера, объявлять переменные, осуществлять преобразование типов. В остальном языки сильно походили друг на друга: в частности, Smm поддерживал стандартные функции и операторы Си. Smm был переименован в ScriptEase, поскольку исходное название звучало слишком негативно, а упоминание в нём Си «отпугивало» людей. На основе этого языка был создан проприетарный продукт CEnvі. В конце ноября 1995 года Nombas разработала версию CEnvі, внедряемую в веб-страницы. Страницы, которые можно было изменять с помощью скриптового языка, получили название Espresso Pages — они демонстрировали использование скриптового языка для создания игры, проверки пользовательского ввода в формы и создания анимации. Espresso Pages позиционировались как демоверсия, призванная помочь представить, что случится, если в браузер будет внедрён язык Smm.

Работали они только в 16-битовом Netscape Navigator под управлением Windows.

Перед Бренданом Эйхом, нанятым в компанию Netscape 4 апреля 1995 года, была поставлена задача внедрить язык программирования Scheme или что-то похожее в браузер Netscape. Помимо Брендана Эйха, в разработке участвовали сооснователь Netscape Communications Марк Андрессен и сооснователь Sun Microsystems Билл Джой: чтобы успеть закончить работы над языком к релизу браузера, компании заключили соглашение о сотрудничестве в разработке. Они ставили перед собой цель обеспечить «язык для склеивания» составляющих частей веб-ресурса: изображений, плагинов, Java-апплетов, который был бы удобен для веб-дизайнеров и программистов, не обладающих высокой квалификацией.

Первоначально язык назывался Mocha, затем он был переименован в LiveScript и предназначался как для программирования на стороне клиента, так и для программирования на стороне сервера (там он должен был называться LiveWire). На синтаксис оказали влияние языки Си и Java, и, поскольку Java в то время было модным словом, 4 декабря 1995 года LiveScript переименовали в JavaScript, получив соответствующую лицензию у Sun. Анонс JavaScript со стороны представителей Netscape и Sun состоялся накануне выпуска второй бета-версии Netscape Navigator. В нём декларируется, что 28 лидирующих ИТ-компаний выразили намерение использовать в своих будущих продуктах JavaScript как объектный скриптовый язык с открытым стандартом.

ОСОБЕННОСТИ WEB-ПРОГРАММИРОВАНИЯ

Давайте введем два определения frontend и backend.

Frontend — все, что браузер может читать, выводить на экран и / или запускать. То есть это HTML, CSS и JavaScript.

HTML (HyperText Markup Language) говорит браузеру, каково содержание страницы, например, «заголовок», «параграф», «список», «элемент списка».

CSS (Cascading Style Sheets) говорит браузеру, как отображать элементы, например, «после первого параграфа отступ в 20 пикселей» или «весь текст в элементе body должен быть темно-серым и написан шрифтом Verdana».

JavaScript говорит браузеру, как реагировать на некоторые взаимодействия, используя легкий язык программирования. Большинство сайтов на самом деле не используют много JavaScript, но если вы нажмете на что-то и содержимое страницы поменяется без белого мигания экрана, значит, где-то использовался JavaScript.

Backend — все, что работает на сервере, то есть «не в браузере» или «на компьютере, подсоединенном к сети (обычно к Интернету), который отвечает на сообщения от других компьютеров».

Для backend'а вы можете использовать любые инструменты, доступные на вашем сервере (который, по сути, является просто компьютером, настроенным для ответов на сообщения). Это означает, что вы можете использовать любой универсальный язык программирования: Ruby, PHP, Python, Java, JavaScript / Node, bash. Это также означает, что вы можете использовать системы управления базами данных, такие как MySQL, PostgreSQL, MongoDB, Cassandra, Redis, Memcached.

Сегодня существует несколько основных архитектур, определяющих, как будут взаимодействовать ваши бэкенд и фронтенд.

Серверные приложения

В этом случае HTTP-запросы отправляются напрямую на сервер приложения, а сервер отвечает HTML-страницей.

Между получением запроса и ответом сервер обычно ищет по запросу информацию в базе данных и встраивает ее в шаблон (ERB, Blade, EJS, Handlebars).

Когда страница загружена в браузере, HTML определяет, что будет показано, CSS — как это будет выглядеть, а JS — всякие особые взаимодействия.

Связь с использованием AJAX

Другой тип архитектуры использует для связи AJAX (Asynchronous JavaScript and XML). Это означает, что JavaScript, загруженный в браузер, отправляет HTTP-запрос (XHR, XML HTTP Request) изнутри страницы и (так сложилось исторически) получает XML-ответ. Сейчас для ответов также можно использовать формат JSON.

Это значит, что у вашего сервера должна быть конечная точка, которая отвечает на запросы JSON- или XML-кодом. Два примера протоколов, используемых для этого — REST и SOAP.

Клиентские (одностраничные) приложения

AJAX позволяет вам загружать данные без обновления страницы. Больше всего это используется в таких фреймворках, как Angular и Ember. После сборки такие приложения отправляются в браузер, и любой последующий рендеринг выполняется на стороне клиента (в браузере).

Такой фронтенд общается с бэкендом через HTTP, используя JSON- или XML-ответы.

ЯЗЫК JAVASCRIPT

Области применения

- Разработка веб-приложений.
- «Активное участие» в AJAX. Эта технология позволила значительно ускорить работу приложений, осуществляя обмен данными с сервером в «фоновом» режиме
- Операционные системы. Windows, Linux и Mac имеют своих браузерных конкурентов, львиная доля кода которых написана на JavaScript
- Мобильные приложения

- Сфера обучения. Любая программистская специальность в университете включает в себя изучение JavaScript в том или ином объеме. Это обусловлено тем, что язык изначально разрабатывался для не очень сильных программистов. Уроки JavaScript логически вплетаются в базовый курс HTML, поэтому освоение проходит достаточно просто.

Возможности

С помощью языка JavaScript можно динамически управлять отображением и содержимым HTML-документов. Можно записывать в отображаемый на экран документ произвольный HTML-код в процессе синтаксического анализа загруженного браузером документа. С помощью объекта Document можно генерировать документы "с нуля" в зависимости от предыдущих действий пользователя или каких-либо иных факторов.

JavaScript позволяет контролировать работу браузера. Например, объект Window поддерживает методы, позволяющие выводить на экран всплывающие диалоговые окна, создавать, открывать и закрывать новые окна браузера, задавать режимы прокрутки и размеры окон и т.д.

JavaScript позволяет взаимодействовать с содержимым документов. Объект Document и содержащиеся в нем объекты позволяют программам читать части HTML-документа и иногда взаимодействовать с ними. Сам текст прочитать невозможно, но можно, например, получить список гипертекстовых ссылок, имеющихся в данном документе. На текущий момент широкие возможности взаимодействия с содержимым документов обеспечивает объект Form и объекты, которые он может содержать: Button, Checkbox, Hidden, Password, Radio, Reset, Select, Submit, Text и Textarea.

JavaScript обеспечивает взаимодействие с пользователем. Важной особенностью этого языка является реализованная в нем возможность определять обработчики событий — произвольные порции кода, которые выполняются при наступлении конкретных событий (обычно действий пользователя). JavaScript позволяет использовать в качестве обработчиков событий любые новые предварительно заданные функции. Например, можно написать программу, которая выведет в строке состояния специальное сообщение, если пользователь установит указатель "мыши" на гипертекстовую ссылку, или выведет на экран

диалоговое окно с запросом на подтверждение выполнения некоторого действия, или осуществит проверку введенного пользователем значения и в случае ошибки ввода выдаст соответствующую диагностику и заставит ввести правильное значение.

JavaScript дает возможность выполнять произвольные математические вычисления. Кроме того, этот язык имеет развитые средства работы со значениями даты и времени. JavaScript был создан в качестве альтернативы CGI-программам и языку сценариев Perl, а также в качестве дополнения в ряде случаев альтернативы языку Java.

Связь с Java

Общим заблуждением является то, что JavaScript аналогичен или тесно связан с Java, это не так. Оба языка имеют C-подобный синтаксис, являются объектно-ориентированными и как правило широко используются в клиентских веб-приложениях. Из важных различий можно отметить:

- Java реализует ООП подход, основанный на классах, JavaScript — на прототипах;
- Java имеет статическую типизацию, JavaScript — динамическую типизацию;
- Java загружается из скомпилированного байт-кода; JavaScript интерпретируется напрямую из файла (но часто с незаметной JIT-компиляцией).

Библиотеки

Для обеспечения высокого уровня абстракции и достижения приемлемой степени кросс-браузерности при разработке веб-приложений используются библиотеки JavaScript. Они представляют собой набор многократно используемых объектов и функций. Среди известных JavaScript библиотек можно отметить React.js, Vue.js, Ember.js, Adobe Spry, AngularJS, Dojo, Extjs, jQuery, Mootools, Prototype, Qooxdoo (англ.), Underscore и Node.js.

Синтаксис в сравнении с языком C/C++

JavaScript – это язык С-подобный, что делает его синтаксис очень похожим на синтаксис языка С, однако у него имеются различия, о них сейчас пойдет речь:

- **JavaScript – не жестко типизированный язык, это означает, что переменным не надо указывать тип, интерпретатор сам разберется. В языке Си для каждой переменной обязательно должен быть указан ее тип.**

Объявление переменной в JS:

var x = 1 // переменная типа integer
var x = 1.5 // переменная типа double

Объявление переменной в С:

int x = 1
double x = 1.5

- **В JavaScript можно создать два вида функций именованные и анонимные.**

Именованная функция (стандартная, часто применяемая):

Объявление функции в JS:

function sum(x,y) { return x+y }
--

Видна везде (даже до объявления функции) и вызывается так:

x=sum(3,5);

Функцию можно присвоить переменной:

s=sum; x=s(3,5); // сработает как и sum(3,5)

Анонимная функция:

Анонимная функция вызывается также как и именованная (т.е. `x=sum(2,3)`), ее так-же можно присваивать другой переменной (`s=sum`). Отличие заключается в следующем:

1. анонимная функция работает только после объявления
2. вытекает из первого, анонимную функцию можно определять в условных операторах (например, в зависимости от условия, функцию `sum` мы можем объявить с разными количеством аргументов):

```
if (str=="dva") {  
  var sum = function(x,y) { return x+y; }  
}  
if (str=="tri") {  
  var sum = function(x,y,z) { return x+y+z; }  
}
```

С именованной так не выйдет, интерпретатор JavaScript сначала ищет именованные функции, далее заносит их в память, а потом только выполняет скрипт, т.е. именованные функции объявляются сразу, не важно где они находятся.

В языке C нет возможности присвоить переменной функцию, а также отсутствует такое понятие, как анонимная функция.

- **В JavaScript в отличие от C нельзя явно указать функции как в нее передавать параметры, по ссылке или по значению.**

Важно помнить, что если в качестве параметров в функцию (именованную или анонимную) мы передаем простые типы вроде: целых чисел, вещественных, логических, строк, то они всегда передаются по значению (входящие параметры не изменяются, функция работает с их копиями). В случае если в функцию передаем массивы или объекты, они передаются по ссылке (входящие параметры изменяются, функция работает с ними напрямую).

- **В JavaScript в отличие от C можно осуществлять передачу функции в качестве аргумента другой функции.**

Функцию (именованную и анонимную) можно передавать по ссылке в качестве аргумента другой функции, например:

```
function sum(x,y) { return x+y;}  
  function factor(f,i) { // factor-функция множитель, умножает  
    return i*f(2,1);    // результат функции f  
  }  
factor(sum,3); // в качестве функции f используем функцию sum
```

Результат функции factor будет равен 9, т.к. если умножить значение переменной i=3 на результат работы функция sum с аргументами 2,1 будет 9.

- **В JavaScript в отличие от C++ нет понятия классов (в C есть структуры, там нет классов и этот язык не поддерживает ООП), объекты тут создаются не как экземпляры некого класса, а с нуля, с последующим указанием свойств и методов или с помощью функции-конструктора.**

Все дело в том, что JavaScript язык не чисто объектно-ориентированного программирования, он принадлежит к некому диалекту ООП, одним из его способов, который называется прототипно-ориентированное программирование.

рассмотрим как создавать объекты:

```
var myOb = {  
  name: "Vladik", //свойство name  
  family: "Ivanov", //свойство family  
  print: function(str) { //метод print(str)  
    if (str=="name")  
      alert (this.name);  
    if (str=="family")  
      alert (this.family);  
  }  
};
```

В JavaScript при определении свойства в отличие от C++ никак нельзя ограничить к нему доступ.

Обращаться к свойствам и методам следует так:

```
alert(myOb.name); // печать свойства name  
myOb.print("family"); //вызов метода print(str)
```

А что если нам надо создать множество объектов похожих на myOb с различными name и family? Тут возникает необходимость использовать конструктор.

Конструктор - это некое подобие класса, он задает по каким правилам должен строиться объект (конструктор в JavaScript - это не часть класса как в C++, тут вообще нет классов, это обычная функция).

Пример:

```
function Person (name_p, family_p) {  
  this.name = name_p; // свойство name  
  this.family = family_p; // свойство family  
  this.print = function(str) { //метод print(str)  
    if (str == "name")  
      alert(this.name);  
    if (str == "family")  
      alert(this.family);  
  };  
}
```

Создаем объекты с помощью конструктора person

```
var p1=new Person("Vladik","Ivanov");  
var p2=new Person("Dima","Petrov");
```

Вызываем методы данных объектов

```
p1.print("name"); // выведет Vladik  
p2.print("family"); //выведет Petrov
```

Здесь были рассмотрены различия в синтаксисе языков JavaScript и C/C++. На самом деле их намного больше, но мы рассмотрели самые

основные из них. Однако стоит отметить, что несмотря на различия JavaScript и C/C++, в них полностью совпадает синтаксис: условных операторов, циклов и массивов, что указывает на то, что программист на C/C++ может с легкостью освоить язык JavaScript.

Отладка

В JavaScript доступ к отладчикам становится особенно полезным при разработке крупных нетривиальных программ из-за различий в реализациях разных браузеров (в частности, в отношении объектной модели документа). Полезно иметь доступ к отладчику для каждого из браузеров, в которых будет работать веб-приложение.

По состоянию на ноябрь 2009 года, Internet Explorer, Opera, Firefox, Safari, и Google Chrome имеют отладчики сценариев.

Internet Explorer имеет три отладчика: Microsoft Visual Studio — самый полный из них, за ним следует Microsoft Script Editor (англ.) (компонент Microsoft Office), и, наконец, свободный Microsoft Script Debugger, гораздо более простой, чем два других. Бесплатный Microsoft Visual Web Developer Express предоставляет ограниченную версию с отладочной функцией JavaScript в Microsoft Visual Studio. В восьмой версии в IE вместе с инструментами для разработчиков появился встроенный отладчик.

В Opera также имеется собственный отладчик — Opera Dragonfly.

Разрабатываемые веб-приложения в Firefox можно отлаживать при помощи встроенных инструментов Firefox Developer Tools.

В Safari входит отладчик JavaScript WebKit Web Inspector. Этот же отладчик доступен и в других браузерах, использующих WebKit: Google Chrome, Arora, Rekonq, Midori и др.

Безопасность

JavaScript позволяет потенциальным авторам вредоносного кода запускать его на любом компьютере сети: для этого достаточно открыть на нём веб-страницу. Это обуславливает наличие двух принципиальных ограничений:

- JavaScript-программы выполняются в песочнице, в которой они могут выполнять только ограниченный круг действий, а не задачи

программирования общего назначения (например, создание файлов, работа с сокетами),

- для JavaScript-кода применяется политика общего происхождения, в соответствии с которой скрипт, встроенный в страницу, не может получить доступ к ряду свойств объектов другой страницы (в частности, к большинству свойств объекта `document`) при отличии в протоколе, хосте и номере порта этих страниц.

Помимо этого, разработчики браузеров вносят дополнительные ограничения в ответ на имеющиеся место злоупотребления. Так появился, в частности, запрет на открытие окна, размер одной стороны которого меньше ста пикселей.

ЗАКЛЮЧЕНИЕ

Таким образом можно сделать вывод, что язык программирования JavaScript очень востребован в наши дни. Созданный в конце XX века он и по сей день остается одним из самых популярных языков для web – программирования. JavaScript – это лёгкий в изучении и применении скриптовый язык, который используется практически на каждом сайте или веб-сервисе в интернете. На его основе создаются и продвигаются библиотеки, расширяющие функционал и упрощающие разработку. Стандартных же возможностей JavaScript хватает для реализации большинства задач, не только в web-программировании, но и с прикладными решениями.

СПИСОК ЛИТЕРАТУРЫ

- <https://ru.wikipedia.org/wiki/JavaScript>
- <http://s-engineer.ru/struktura-i-sintaksis-javascript-vyzhimka/>
- <http://fb.ru/article/336429/chto-takoe-javascript-istoriya-poyavleniya-i-osnovnyie-osobennosti>
- <https://tproger.ru/translations/frontend-backend-interaction/>