

## Особенности обучения программированию на языке JavaScript

Абдурахманов Орхан Абдурахманович,  
Начальник отдела информатизации  
Филиал Дагестанского государственного педагогического университета в г.Дербенте, г.Дербент,  
e-mail: [sckat@rambler.ru](mailto:sckat@rambler.ru),  
url: <http://Abdurakhmanov-Orkhan.scipeople.ru>

Абдурахманов Давид Абдурахманович,  
Зам. директора по информатизации  
Профессиональное училище №4 г.Дербент  
e-mail: [davidboss@mail.ru](mailto:davidboss@mail.ru)  
url: <http://davidboss.narod.ru>

*С развитием современных информационных технологий вопрос обучения программированию в школьных и университетских курсах информатики отошел на второй план. Этот процесс естественен, поскольку на сегодняшний день работа на компьютере по обработке текстовой, числовой, графической или мультимедийной информации сводится к применению готовых программных пакетов. Отпала необходимость в написании собственных прикладных программ для решения повседневных задач. Но при этом базовые навыки алгоритмизации и программирования все еще входят в стандарты среднего и высшего образования. Поэтому актуален вопрос выбора наиболее эффективных и общедоступных средств обучения программированию в рамках базовых и специальных курсов.*

*В данной статье рассматриваются основы языка JavaScript. Анализируются преимущества и недостатки применения этого языка при обучении программированию.*

Современный учитель как и его коллеги, преподававшие программирование в 60-х - 90-х годах XX столетия, сталкивается с проблемой выбора средства обучения программированию. Стандартным выбором среды программирования можно считать ставшие уже классическими языки: Basic, Pascal, C/C++, Fortran, Cobol, Ada, Simula и т. д. Если часть из этих языков живут и продолжают развиваться, то другая часть фактически стала достоянием истории. Наиболее простым и распространенным в школьных курсах информатики является язык Basic. Успех этого языка был обеспечен его простотой, ясностью и нежесткостью требований к программному коду. Популярность языка Basic была закреплена появлением его объектно-ориентированной версии Microsoft Visual Basic и модификацией VBA, встроенной как язык сценариев автоматизации в известный офисный пакет. Язык Pascal, разработанный профессором Николаусом Виртом, де факто считается стандартом обучения программированию в университетской среде. Современная версия Pascal реализована в визуальной объектно-ориентированной среде Delphi — одной из мощнейших систем быстрой разработки масштабируемых программных комплексов. Языки C и C++ - являются мировым стандартом в промышленном и профессиональном программировании, относятся к обязательным языкам при обучении в технических учебных заведениях. Объектно - ориентированный язык C++ так же имеет реализации в визуальных средах программирования таких как: Microsoft Visual C++ и Borland C++ Builder.

Ставшими популярными и неуклонно развивающиеся с 1990 г. Интернет - технологии внесли свои коррективы в мир языков программирования. Появились такие языки как: Java, Perl, PHP, Python, JavaScript и др.

В данной статье затрагиваются особенности обучения программированию с помощью языка JavaScript. Чтобы почувствовать эти особенности рассмотрим решения конкретных задач.

Изложение основ языка JavaScript начнем с традиционной задачи, с которой начинается знакомство с новым языком. Это задача вывода приветственного сообщения «Здравствуй, Мир!» (или "Hello World!" в англоязычном написании). Вывод сообщений в JavaScript может осуществляться разными способами, наиболее простым из них является использование метода `alert()` объекта `window`. Воспользовавшись любым текстовым редактором, сохраним в файле `hello.htm` следующий код:

```
<script> window.alert("Здравствуй Мир!"); </script>
```

Загрузка файла hello.htm в веб-браузере позволит просмотреть результат работы первой программы — диалоговое окно с приветствующим сообщением.

Знакомство с JavaScript продолжим, попробовав свои силы в решении задачи, к примеру, из учебника алгебры 8 класса: «найти корни квадратного уравнения  $ax^2+bx+c=0$ ». Но вначале немного теории.

Программа на языке JavaScript по сути есть набор алгоритмических конструкций (if-then; for; while; switch-case; try-catch) управляющих рядом встроенных и пользовательских объектов. Под встроенными понимаются объекты веб-браузера и объекты ядра JavaScript. Базовыми объектами веб-браузера являются объекты: window, document, navigator, history, location, event, screen. Наряду с объектами браузера программисту доступны предопределенные объекты ядра JavaScript: Number, String, Array, Boolean, Math, Date, Function, Object. Каждый из этих объектов неявно используется при вводе в программу соответственно: чисел, строк, массивов, логических величин, математических функций, дат, функций и объектов пользователя. Подробнее остановимся на объектах. Программное решение поставленной задачи предполагает описание некоторой последовательности команд обрабатывающих исходные данные. Идея объектно-ориентированного программирования предлагает объединить данные-свойства и методы их обработки в рамках единого объекта. Объекты кроме свойств и методов могут содержать описание реакций объекта на определенные события — нажатие кнопки, движение курсора, изменение текста и т. д.

Теперь вернемся к нашей задаче — вычислению корней квадратного уравнения. Переформулируем задачу в виде: исходные данные, вычисления, результат. Исходными данными служат значения коэффициентов уравнения:  $a$ ,  $b$ ,  $c$ . Вычисление производится по формулам:  $D=b*b-4*a*c$ ,  $x1=(-b-sqrt(D))/(2*a)$ ,  $x2=(-b+sqrt(D))/(2*a)$ . Результатом является сообщение о найденных корнях  $x1$ ,  $x2$  или сообщение об отсутствии действительных корней. Зададим описание объекта kvad\_uravn в свойствах  $a$ ,  $b$ ,  $c$ ,  $x1$ ,  $x2$  которого будем хранить исходные данные и результаты вычисления, а в методе solve() реализуем вычисление корней:

```
<b>Решение квадратного уравнения<br>
```

```
ax^2+bx+c=0</b>
```

```
<div style='border: solid blue 2pt; width:400'>
```

```
  a= <input id=a type=text><br>
```

```
  b= <input id=b type=text><br>
```

```
  c= <input id=c type=text><br>
```

```
  <input id=calc type=button  
value=Решить ><br>
```

```
  <span id=result ></span>
```

```
</div>
```

```
<script>
```

```
function kvad_uravn() {
```

```
  uravn = new Object();
```

```
  uravn.init = function (){
```

```
    this.a = parseFloat(a.value);
```

```
    this.b = parseFloat(b.value);
```

```
    this.c = parseFloat(c.value);
```

```
  }
```

```
  uravn.solve = function (){
```

```
    this.init();
```

```
    D=this.b*this.b-4*this.a*this.c;
```

```
    if(D>=0) {
```

```
      this.x1 = (-this.b-Math.sqrt(D))/(2*this.a);
```

```
      this.x2 = (-this.b+Math.sqrt(D))/(2*this.a);
```

```
      result.innerHTML = 'Найдены корни: x1='+this.x1+' x2='+this.x2;
```

```
    } else {
```

```
      result.innerHTML = 'D<0 - нет действительных корней'
```

```
    }
```

```
  }
```

```
  return uravn;
```

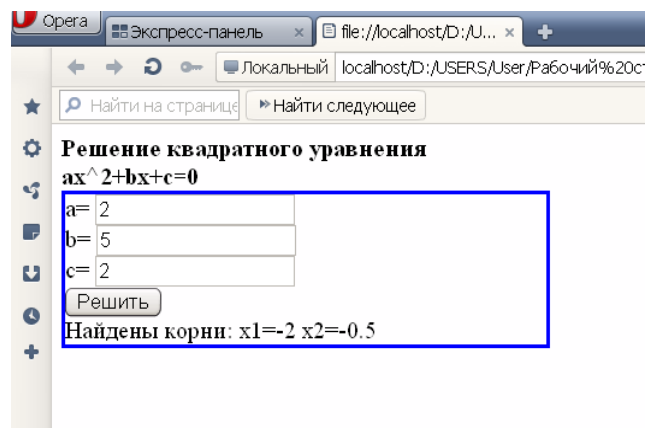


Рис. 1

```

    }
    uravn1 = new kvad_uravn();
    calc.onclick = function() { uravn1.solve(); }
</script>

```

Сохранив приведенный код в файле *uravnenie.htm* и загрузив его в браузере получим результат показанный на рис.1.

Разберем ряд особенностей полученного нами решения. Первым делом отметим, что интерфейс программы создается средствами языка HTML, так как JavaScript программы исполняются внутри html-страницы. Открывающий `<script>` и закрывающий `</script>` тэги, внутри которых располагается JavaScript код, так же являются командами языка HTML. В ряде тэгов мы указали атрибут `id`. Это было сделано с целью в дальнейшем получить доступ в нашей программе к одноименным объектам. Так первые три тэга `<input>` получили названия `a`, `b`, `c`, после чего в методе объекта `uravn.init()` они используются для получения значений вводимых пользователем коэффициентов уравнения. Это происходит обращением к свойству `value`, которые мы сразу преобразуем из текста в действительные числа, посредством встроенной функции `parseFloat()`. Отметим отдельно универсальную объектную ссылку `this`, указывающую на текущий объект. Так команды `this.a`, `this.b`, `this.c` обозначают обращение к свойствам нового объекта `uravn`. Основные расчетные формулы вынесены в отдельный метод `uravn.solve()`. В этом методе вычисляется дискриминант  $D$ , проверяется его неотрицательность, вычисляются корни и выводится результат. При выводе результата используется свойство `innerHTML` объекта `result`, это свойство есть у всех контейнерных элементов HTML страницы. Последние две команды нашей программы создают экземпляр-переменную `uravn1` описанного нами объекта `kvad_uravn` и указывают обработчик события нажатия кнопки `calc`, помеченной надписью «Решить».

Приведенный вариант решения достаточно простой задачи может показаться излишне громоздким. Действительно, решение задачи можно было бы сократить на пару — другую строчек, если отказаться от создания нового пользовательского объекта `kvad_uravn` (заинтересованным лицам в качестве упражнения можем предложить переписать эту программу в структурном виде). Но, однако, преимущества описания нового объекта выйдут на первый план, если мы захотим решать сразу не одно, а несколько квадратных уравнений. При структурном подходе нам придется создавать отдельные переменные `a1`, `b1`, `c1`, `D1`, `x11`, `x12`; `a2`, `b2`, `c2`, `D2`, `x21`, `x22`; ... для каждого из уравнений, или их массивы. Писать вызовы функций для каждого из уравнений. В то время, как в объектном подходе достаточно будет задать несколько экземпляров прототипа объекта (в языках Java, C++ именуется классом) `kvad_uravn`: `u1`, `u2`, `u3`: `new kvad_uravn()`; . При этом автоматически будет создан весь набор свойств и методов для каждого из экземпляров `u1`, `u2`, `u3`. А это означает, экономию строк кода, которые пришлось бы писать в структурном варианте программы, и плюс четкая, разделенная логика объектной программы. Ведь легче пользоваться уже отлаженным объектом, чем «тонуть в рядах» однообразных переменных. В конце лишь, отметим что как в структурном так и в объектном подходах указанного дублирования переменных недостаточно для решения нескольких уравнений. Еще необходимо изменить ввод исходных данных — коэффициентов уравнений ( что так же предлагаем в качестве упражнения).

В заключении отметим, что язык JavaScript не лишен недостатков, впрочем как и другие языки. Так имеются сложности с выполнением одной и той же программы в разных версиях браузеров; отсутствие в ряде браузеров прямой работы с файлами; сложность выявления логических ошибок — особенно вызванными опечатками (язык JavaScript регистрозависимый в нем переменные `x` и `X` считаются двумя различными переменными, и интерпретатор браузера ничего не сообщит о случайной опечатке). Но однако язык JavaScript — динамически развивающийся, он объединяет и позволяет управлять различными технологиями HTML, CSS, ActiveX, Flash, Ajax, XML, т. е. находится на острие развития современных интернет-технологий. Написанная JavaScript программа сразу же может быть опубликована в сети Интернет и работать в режиме онлайн. Полученные знания и навыки программирования на JavaScript учащимися, учителями и преподавателями будут также полезны и в их дальнейшей профессиональной деятельности, если они столкнутся с использованием и разработкой интернет-приложений.

Ваши предложения, вопросы и замечания можете присылать на указанные адреса авторов ([skat@rambler.ru](mailto:skat@rambler.ru), [davidboss@mail.ru](mailto:davidboss@mail.ru)).

Список литературы.

1. Дунаев В. Самоучитель JavaScript. – Питер: изд. ПИТЕР, 2003
2. Рева О.Н. JavaScript. Просто как дважды два. – М.: Эксмо, 2007
3. Мартынов Н.Н. Алгоритмизация и основы объектно-ориентированного программирования на JavaScript. – М.: БИНОМ, 2010