

Учреждение образования  
«Белорусский государственный педагогический университет  
имени Максима Танка»

Факультет физико-математический  
Кафедра информатики и методики преподавания информатики

СОГЛАСОВАНО

И.о. заведующего кафедрой

С.В. Вабищевич  
16.11.2017 2017 г.



СОГЛАСОВАНО

Декан факультета

С.И. Василенц  
08.11.2017 2017 г.

(рег. № 14-3-183.03.06 2017г.)

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

«Программирование на языке JavaScript (дисциплина по выбору)»

Для специальности 1-02.05.01 Математика и информатика

Составители: Ю.А. Быкадоров, канд. физ.-мат. наук, доцент

Рассмотрено и утверждено

на заседании Совета БГПУ 16.11.2017 2017 г. протокол № 10

## ОГЛАВЛЕНИЕ

|                                                          |    |
|----------------------------------------------------------|----|
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА .....                              | 3  |
| ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ .....                               | 7  |
| Тема 1. Основные понятия языка JavaScript.....           | 7  |
| Тема 2. Алгоритмические структуры языка JavaScript ..... | 10 |
| Тема 3. События и CSS .....                              | 12 |
| Тема 4. Структура языка JavaScript.....                  | 13 |
| Тема 5. Примеры задания динамических страниц .....       | 15 |
| ПРАКТИЧЕСКИЙ РАЗДЕЛ .....                                | 16 |
| Тема 2. Алгоритмические структуры языка JavaScript ..... | 16 |
| Тема 3. События и CSS .....                              | 17 |
| Тема 4. Структура языка JavaScript.....                  | 18 |
| РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ.....                              | 24 |
| ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ.....                              | 26 |

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа дисциплины по выбору студента «Основы программирования на языке JavaScript» составлена для студентов физико-математического факультета в соответствии с требованиями образовательного стандарта высшего образования специальности 1-02 05 01 Математика и информатика.

Изучение дисциплины предусматривает знакомство с основами программирования на языке JavaScript с целью реализации интерактивных возможностей веб-программирования. Студенты познакомятся с технологиями создания интерактивных динамических веб-страниц разного типа. В результате изучения содержания данной дисциплины по выбору студенты смогут создавать свои интерактивные динамические веб-страницы различного назначения, а также виртуальные модели различных физических и информационных процессов. Возможности и опыт, приобретаемые студентами в процессе изучения данной учебной дисциплины, позволят им осуществлять междисциплинарные практико-ориентированные связи, поддерживать идеи профильного обучения в школе, развивать олимпиадное движение и исследовательскую деятельность по информатике, а также оптимизировать организационно-управленческий потенциал учебного процесса в учреждениях системы образования.

**Цель учебной дисциплины** – развитие алгоритмического мышления студентов, развитие у студентов целостного представления о возможностях использования языка программирования JavaScript в процессе разработки современных интерактивных динамических веб-страниц.

### **Основные задачи учебной дисциплины:**

- совершенствование навыков самостоятельности при разработке программного обеспечения;
- освоение возможностей языка программирования JavaScript;
- формирование понятий о семантике, синтаксисе, стандартах и структуре языка JavaScript;
- освоение программного обеспечения для отладки и тестирования кода на языке JavaScript;
- освоение приемов разработки интерактивных динамических веб-страниц средствами языка JavaScript;
- освоение студентами современных подходов и методов создания интерактивных динамических веб-страниц средствами языка JavaScript.

### ***Место учебной дисциплины в системе подготовки специалиста***

Изучение учебной дисциплины по выбору «**Основы программирования на языке JavaScript**» опирается на основные академические, социально-личностные и профессиональные компетенции, сформированные у студентов в процессе изучения ими таких учебных дисциплин как «Технологии программирования и методы алгоритмизации»,

«Вычислительные методы и компьютерное моделирование», «Практикум по решению задач по информатике». Благодаря ее изучению и приобретению умений создавать интерактивные динамические веб-страницы студенты смогут реализовать свои профессиональные потребности на современном уровне.

### **Профессиональные компетенции студентов**

Учебная дисциплина по выбору «**Основы программирования на языке JavaScript**» входит в компонент учреждения высшего образования. Изучение дисциплины по выбору студента «Основы программирования на языке JavaScript» должно обеспечить формирование у студентов академических, социально-личностных и профессиональных компетенций.

#### ***Требования к академическим компетенциям***

Студент должен:

- АК–3. Владеть исследовательскими навыками;
- АК–4. Уметь работать самостоятельно; – АК–5. Быть способным порождать новые идеи (обладать креативностью);
- АК–6. Владеть междисциплинарным подходом при решении проблемы;
- АК–7. Иметь навыки, связанные с использованием технических средств устройств, управлением информацией и работой с компьютером;
- АК–9. Уметь учиться, повышать свою квалификацию в течение всей жизни;
- АК–10. Уметь осуществлять учебно-исследовательскую деятельность. Требования к социально-личностным компетенциям Студент должен:
- СЛК–3. Обладать способностью к межличностным коммуникациям;
- СЛК–7. Быть способным к осуществлению самообразования и самосовершенствования профессиональной деятельности.

#### ***Требования к профессиональным компетенциям***

Студент должен быть способен:

##### **Обучающая деятельность**

- ПК–1. Управлять учебно-познавательной и учебно-исследовательской деятельностью обучающихся;
- ПК–2. Использовать оптимальные методы, формы и средства обучения;
- ПК–3. Организовывать и проводить учебные занятия различных видов;
- ПК–4. Организовывать самостоятельную работу учащихся.

##### **Воспитательная деятельность**

- ПК–8. Формировать базовые компоненты культуры личности воспитанника. Развивающая деятельность

– ПК–12. Развивать навыки самостоятельной работы обучающихся с учебной, справочной, научной литературой и др. источниками информации.

### **Ценностно-ориентационная деятельность**

– ПК–16. Оценивать учебные достижения учащихся, а также уровни их воспитанности и развития;

– ПК–17. Осуществлять профессиональной самообразование и самовоспитание с целью совершенствования профессиональной деятельности.

В результате изучения учебной дисциплины студент должен знать:

– понятия, связанные с семантикой, синтаксисом, стандартами и структурой языка JavaScript;

– программное обеспечение для отладки и тестирования программ на языке JavaScript;

– методы языка JavaScript, реализующие интерактивность и динамичность на веб-страницах; – приемы создания интерактивных динамических веб-страниц средствами языка JavaScript.

В результате изучения учебной дисциплины студент должен уметь:

– обрабатывать цифровую, текстовую и графическую информацию средствами языка JavaScript;

– разрабатывать интерактивные динамические веб-страницы средствами языка JavaScript.

В результате изучения дисциплины по выбору студент должен владеть:

– методами поиска, анализа и дидактической адаптации научной информации по решению задач создания интерактивных веб-страниц;

– практическими умениями применять полученные знания в нестандартных ситуациях науки и жизни;

– современными технологиями и средствами для решения профессиональных задач.

### **Структура и содержание учебной дисциплины**

Учебная дисциплина изучается на протяжении одного семестра, и ее содержание включает пять тем. Первая тема посвящена основным понятиям языка программирования JavaScript. Вторая тема связана с изучением основных алгоритмических структур языка. Третья тема посвящена событийному программированию и средствам CSS. Четвертая тема раскрывает структуру языка JavaScript. Пятая тема связана с освоением различных приемов создания динамических веб-страниц.

Данная учебная программа является основным документом, определяющим объем и содержание учебной дисциплины по выбору «Основы программирования на языке JavaScript» для специальности 1-02 05 01 Математика и информатика.

### **Методы обучения**

Обучение учебной дисциплине проходит в рамках организации лекционных и лабораторных занятий. При чтении лекций особое внимание следует уделять использованию мультимедийных технологий для представления графических иллюстраций и демонстрации приемов работы с программными средствами.

Организация лабораторных занятий предполагает использование личностно-ориентированных методов обучения, основанных на подготовке обширных методических рекомендаций, что способствует развитию индивидуально-творческих способностей студентов и приобретению ими умений самостоятельной работы.

Содержание самостоятельной работы студентов разрабатывается в соответствии с задачами изучения дисциплины. Формы самостоятельной работы могут обладать достаточным разнообразием: самостоятельная работа по программированию и отладке интерактивных веб-страниц во время основных аудиторных занятий (лекций и лабораторных занятий), самостоятельная работа в форме консультаций, в форме внеаудиторной работы. Для управления самостоятельной работой рекомендуется использовать электронные средства обучения, тестирующие программы. Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

#### ***Распределение общего количества часов по формам обучения и семестрам***

##### **Дневная форма получения высшего образования:**

Всего на учебную дисциплину – 50 часов.

8 семестр – 26 часов аудиторных (14 часов – лекции, 12 часов – лабораторные занятия), 24 часа – самостоятельная работа.

Форма контроля – зачет (8 семестр).

##### **Заочная форма получения образования:**

Всего на учебную дисциплину – 50 часов.

9 семестр – 8 часов аудиторных (4 часа – лекции, 4 часа – лабораторные занятия), 42 часа – самостоятельная работа.

Форма контроля – зачет (10 семестр).

# ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

## Тема 1. Основные понятия языка JavaScript

### Лекция 1. Области применения, особенности и возможности скриптовых языков программирования

Язык программирования JavaScript называют языком сценариев или языком скриптов ("скрипт" по-английски — сценарий).

Сценарий на интерактивных веб-страницах — это программа действий браузера в ответ на действие пользователя. Программу на языке JavaScript называют также JS-программой, сценарием или программным кодом, а команды программы инструкциями.

Программа представляет собой текст формата TXT, который большинство программистов создают при помощи текстовых редакторов. Мы будем использовать редактор Notepad++, который позволяет не только создавать сценарии, но и запускать их на исполнение.

JS-программы встраиваются в HTML-документ с помощью тегов `<script>` и `</script>`. Например,

```
<html>
<!-- saved from url=(0014)about:internet -->
<title>          </title>
<script>

</script>
</html>
```

Т.е. формально мы создаем текст на языке разметки гипертекстов HTML (HTML -документ), причем название JS-программы будем записывать между тегами `<title></title>`.

После записи программы, сохраняем ее. В окне сохранения после ввода имени файла следует в поле «Тип файла» найти и выбрать тип

**Hyper Text Markup Language file (\*.html; \*.htm; \*.shtml; \*.shtm; \*.xhtml; \*.hta;)**

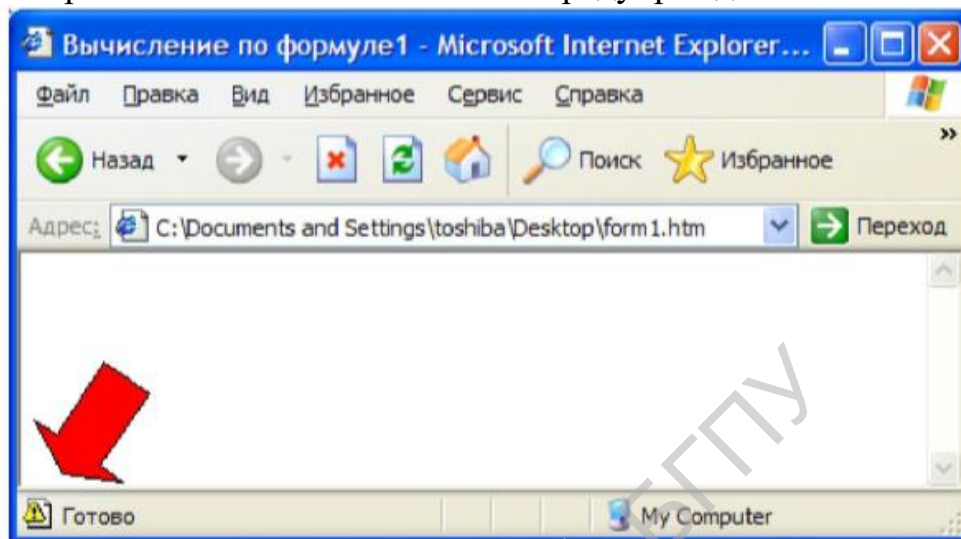
Потом запускаем файл, выбрав в меню редактора Notepad++ пункт «Запуск», а в нем команду запуска в браузере IE Запуск в IE. Файл запускается в браузере Internet Explorer

Отладка проводится так. После изменения текста программы сохраняем изменения (для этого есть кнопка «дискета» на панели инструментов редактора, пункт меню или клавишная комбинация).

Потом обновляем файл в браузере Internet Explorer с помощью соответствующей кнопки.

Именно так, без дополнительного запуска файла, проводится изменение и отладка сценариев JavaScript.

Если в сценарии есть синтаксическая ошибка, то в браузере IE8 в статусной строке слева появляется значок предупреждения.



Двойной щелчок по значку открывает диалоговое окно, в котором нужно щелкнуть по кнопке, которая открывает подробности. В раскрытом окне можно получить информацию о типе синтаксической ошибки и номер строки программы.

## Лекция 2. Семантика, синтаксис, стандарты языка JavaScript

Синтаксис языка **JavaScript** во многом напоминает синтаксис языков C, C++ и Java. В **JavaScript**:

- все идентификаторы регистрозависимы,
- в названиях переменных можно использовать буквы, подчёркивание,
- символ доллара, арабские цифры,
- названия переменных не могут начинаться с цифры,
- для оформления однострочных комментариев используются //,
- многострочные и внутристрочные комментарии начинаются с /\* и заканчиваются \*/.
- команды разделяются знаком «;»;
- типы данных не объявляются.

В информатике в качестве данных используются числа, строки (цепочки Символов), тексты (наборы строк) и др. Исходные данные и результаты в языке **JavaScript** представляются в форме констант и переменных.

Числа (числовые константы) на языке сценариев **JavaScript** записываются по обычным правилам. Причем в дробных числах целая и дробная части разделяются точкой.

Числа в языке **JavaScript** называются данными типа `Number` или данными числового типа.

Строки символов (строковые константы) в языке **JavaScript** записываются в кавычках, двойных «"» или одинарных «'».

Строки в языке **JavaScript** называются данными типа `string` или данными строкового типа.

Логические константы — это служебные слова `True` и `false`. Логические константы называют данными типа `Boolean` или данными логического типа.

Переменная величина (переменная) — это объект языка, который имеет имя, значение и тип.

Имя используется для записи выражений, значение заменяет имя при вычислении выражений, а тип определяется типом значения.

Имя переменной величины в языке **JavaScript** имеет вид слова (без пробелов). Кроме букв в именах могут использоваться символ подчеркивания «\_», символ «\$» (доллар) и цифры.

Имя переменной не может начинаться с цифры.

Строчные и прописные символы считаются различными.

В языке **JavaScript** в качестве имен величин нельзя использовать ключевые (служебные) слова языка.

Значением переменной величины может быть числовая или строковая константа, константа другого типа.

В языке **JavaScript** переменные получают свои значения при исполнении команд программы.

В языке **JavaScript** имеется большой набор операторов (специальных символов или наборов Символов). Например, арифметические операторы `+`, `-`, `*`, `/` служат для составления арифметических выражений.

Заметим, что в языке **JavaScript** нет оператора возведения в степень, как в языке `Pascal`.

Символ `"=` называется оператором присваивания (как в языке `C`).

С помощью оператора присваивания задаются и изменяются значения переменных величин.

Принцип построения команды присваивания ясен из примера:

```
x1=3.78;
```

Команда читается так: "величине `x1` присвоить значение 3.78". Переменным величинам могут присваиваться значения любого типа — числового, строкового и др.

**JavaScript** — это динамически типизированный язык. Это означает, что тип переменной может меняться в процессе выполнения программы.

В программе на языке **JavaScript** команды записываются одна за другой и отделяются друг от друга точкой с запятой «;».

В одной строке программы могут быть записаны несколько команд.

В конце строки точку с запятой можно не ставить, но это считается ПЛОХИМ ТОНОМ.

### Лекция 3. Алгоритмические структуры языка JavaScript

Массивы в программах используют для обработки больших объемов данных с помощью алгоритмических конструкций повторения.

Алгоритмическая конструкция повторения For (цикл for) в языке **JavaScript** (как и в C++) имеет вид

```
for (      ;      ;      ) {  
  
}
```

### Тема 2. Алгоритмические структуры языка JavaScript

В языке **JavaScript** имеется большой набор операторов (специальных символов или наборов Символов). Например, арифметические операторы +, -, \*, / служат для составления арифметических выражений.

Заметим, что в языке **JavaScript** нет оператора возведения в степень, как в языке Pascal.

Символ "=" называется оператором присваивания (как в языке C).

С помощью оператора присваивания задаются и изменяются значения переменных величин.

Принцип построения команды присваивания ясен из примера:

```
x1=3.78;
```

Команда читается так: "величине x1 присвоить значение 3.78". Переменным величинам могут присваиваться значения любого типа — числового, строкового и др.

**JavaScript** — это динамически типизированный язык. Это означает, что тип переменной может меняться в процессе выполнения программы.

В программе на языке **JavaScript** команды записываются одна за другой и отделяются друг от друга точкой с запятой «;».

В одной строке программы могут быть записаны несколько команд.

В конце строки точку с запятой можно не ставить, но это считается ПЛОХИМ ТОНОМ.

Массивы в программах используют для обработки больших объемов данных с помощью алгоритмических конструкций повторения.

Алгоритмическая конструкция повторения For (цикл for) в языке **JavaScript** (как и в C++) имеет вид

```
for (      ;      ;      ) {  
  
}
```

ЦИКЛ for используется для повторения блока команд конкретное число раз. Указание о том, сколько раз повторяется блок цикла, записывается

после служебного слова `for` в круглых скобках. При этом обязательно вводится переменная цикла и для нее записываются три выражения. Если нужно исполнить блок команд 21 раз, тогда первая строка цикла будет иметь вид:

```
for (i=1; i<=21; i++) {
```

Запись в круглых скобках означает:

- 1) введена переменная цикла `i` для нее задано начальное значение `i=1`;
- 2) ЦИКЛ будет повторяться до тех пор, пока `i<=21`, т.е 21 раз;
- 3) `i++` означает, что значение `i` увеличивается на единицу каждый раз, когда исполняется блок конструкции.

Команда `i++` — это сокращенная запись команды присваивания `i=i+1`.

Замечание. В цикле "для" МОЖНО использовать команду организации счетчика вида `(i=i- 1)`. В этом случае в круглых скобках нужно записать выражения `(i=21; i>=13 i--)`, И счетчик будет давать число оставшихся повторений.

Конструкция ветвления имеет две формы: полную и неполную. Полная конструкция ветвления имеет вид.

```
if ( ) {  
  
}  
else {  
  
}
```

При выполнении условия, которое записывается в круглых скобках после `If`, исполняются команды, которые записаны между фигурными скобками после `if`. Если условие не выполняется, то исполняются команды, которые записаны между фигурными скобками после `else`. Неполная конструкция имеет вид.

```
if ( ) {  
  
}
```

Если условие не выполняется, то команды в фигурных скобках пропускаются и все. Объект языка javascript — это набор переменных величин и функций, объединенных одним именем.

Существуют стандартные объекты языка с именами `Math`, `document`, `window` и др.

Имя свойства объекта строится по шаблону

Имя\_объекта.Имя\_свойства

Имя метода объекта строится по аналогичному шаблону

Имя\_объекта. Имя\_метода( )

Стандартный объект Math (математика) является объектом, встроенным в ядро языка JavaScript.

Свойства стандартного объекта Math (математика) имеют неизменяемые значения:

Math . PI — значение постоянной;

Math . E — значение числа 6.

На языке JavaScript, как и в других языках программирования, можно строить вспомогательные программы, которые потом можно использовать в основной программе для сокращения записей. В языке JavaScript вспомогательные программы носят название — функции пользователя. Функции пользователя создаются как первые разделы основной программы. Конструкция функции пользователя имеет вид

```
function ( ) {  
  
}
```

После служебного слова записывается ИМЯ функции, а В круглых скобках — ее аргументы. Код подпрограммы записывается между фигурными скобками.

### Тема 3. События и CSS

#### Лекция 4. События и CSS

Понятие "событие" является понятием Динамического HTML. События могут инициироваться пользователем ИЛИ процессом, происходящим в компьютере. События имеют определенные имена, в частности:

**onkeydown** — пользователь нажал любую клавишу клавиатуры, но не отпустил;

**onkeypress** — пользователь нажал И отпустил любую клавишу клавиатуры;

**onhelp** — пользователь нажал клавишу F1 клавиатуры;

**onclick** — пользователь выполнил щелчок левой клавишей мыши по объекту

**onmousedown** — пользователь нажал левую кнопку мыши;

**onmouseout** — пользователь вывел указатель мыши за границы объекта;

В HTML есть и другие события, которые используются не так часто, как перечисленные. Чтобы объект веб-страницы стал реагировать на определенное событие, в начальный тег этого объекта вносится атрибут вида

**имя\_события = "func()"**

Как только с объектом происходит событие с таким именем, выполняется функция, записанная в равенстве справа. Свойства любого объекта веб-страницы можно описать с помощью атрибутов и стилевых свойств. Стилиевые свойства записываются как значение универсального атрибута style (стиль).

Значение атрибута style включает одно или несколько стилевых правил, разделенных точками с запятой (список стилевых правил). В стилевом правиле слева записывается имя стилевого свойства, а после двоеточия — значение стилевого свойства. В списке и в стилевых правилах около знаков препинания можно добавлять пробелы.

Список стилевых правил атрибута style обычно называют каскадной таблицей стилей (CSS — Cascad Style Sheet). Описание атрибута style в тегах объектов называют встраиванием таблиц стилей.

Универсальность каскадных таблиц стилей состоит в том, что имена стилевых свойств определены однозначно и не зависят от вида тега, в котором они используются.

Дело в том, что имена стилевых свойств в таблицах CSS и имена свойств на языке JavaScript могут несколько различаться. Есть общее правило связи между именем стилевого свойства и именем свойства на языке JavaScript, которое состоит в следующем:

Если имя стилевого свойства CSS имеет дефисы, то имя свойства на языке JavaScript получают удалением дефиса и заменой на верхний регистр первой после дефиса буквы.

Не все браузеры в равной степени поддерживают все функции языка.

## **Тема 4. Структура языка JavaScript**

### **Лекция 5. Структура языка JavaScript**

Язык JavaScript объединяет в себе три составляющих части, которые образуют его структуру:

- ядро языка;
- средства по работе с объектной моделью Документа (DOM);
- средства по работе с объектной моделью браузера (BOM).

Именно в таком порядке язык и изучают.

Ядро языка — это средства языка JavaScript, связанные с обработкой чисел, строк, массивов, средства организации повторений, ветвлений и т.п.

BOM — это сокращение понятия "объектная модель Документа".

BOM — это сокращение понятия "объектная модель браузера".

При отображении веб-страницы в браузере автоматически создаются специальные объекты браузера.

Главным среди объектов браузера является объект «окно браузера», который имеет имя window. Все остальные объекты браузера являются дочерними для этого объекта.

Объект window и его дочерние объекты составляют иерархическую модель, которая называется объектной моделью браузера (BOM — Browser Object Model).

Дочерними для объекта window являются следующие объекты:

document — HTML-Документ в окне браузера;

screen — совокупность параметров монитора;

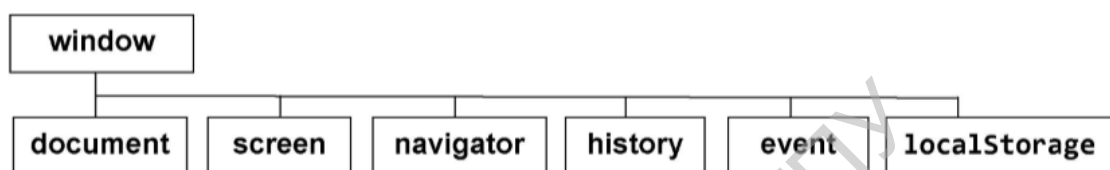
navigator — совокупность параметров браузера;

history — список адресов посещенных веб-страниц;

event — совокупность параметров текущего события;

location — содержит URL окна;

На рисунке представлены основные элементы объектной модели браузера.



Для объекта window Дочерней является также коллекция

Frames — коллекция (список) объектов-фреймов.

Дело в том, что окно браузера может Делиться на несколько областей (фреймов, Frames), В каждой ИЗ которых может отображаться своя веб-страница. Но последним стандартом HTML5 фреймы исключены из рекомендованных элементов DOM. Чтобы обратиться к Дочернему объекту, используют полное имя, которое строится с помощью разделяющей точки:

*имя\_объекта.имя\_дочернего\_объекта*

Объекты модели BOM могут иметь свойства и методы, как стандартные объекты ядра JavaScript. Чтобы обратиться к свойству объекта вом или к его методу, используют полное ИМЯ свойства ИЛИ метода, которое строится с помощью разделяющей точки:

*полное\_имя\_объекта.имя\_свойства*

*полное\_имя\_объекта.имя\_метода()*

Например, объект window имеет несколько важных для языка JavaScript методов:

alert () — вывод сообщения в Диалоговом окне;

prompt () — вывод запроса в Диалоговом окне;

confirm() — вывод Диалогового окна ДЛЯ подтверждения;

moveTo() — смещение окна браузера в определенную точку экрана (X,y);

resizeTo() — устанавливает размеры окна в соответствии с указанными в параметрах значениями пикселей (ширина, высота);

**Лекция 6. Объектная модель документа (BOM)**

HTML-теги каждого HTML-документа, отображаемого В браузере, составляют иерархическую структуру, которая называется объектной моделью документа (DOM — Document Object Model).

HTML-теги документа называют объектами модели BOM. Иерархия тегов понимается в том смысле, что если один тег входит в другой, то входящий является дочерним для второго. Изображение всей страницы задается объектом BOM, который определяется парой тегов `<html> </html>`. Дочерними для него являются объекты, задаваемые тегами `<head></head>`. Головным объектом модели BOM является объект `document` модели BOM.

## **Тема 5. Примеры задания динамических страниц**

### **Лекция 7. Приемы создания динамических страниц**

#### **Пример 1. Перетаскивание объектов**

Разместим в окне браузера две картинки, текстовый абзац и закрашенный зеленым цветом прямоугольник, который можно построить при помощи тега `<div></div>`. Реализуем возможность перетаскивания по окну отображаемых объектов при помощи указателя мыши.

При создании сайтов широко используется прием, когда нахождение указателя мыши над пунктом меню вызывает выделение этого пункта цветным прямоугольником, подсветкой или иным образом.

Чтобы реализовать этот эффект, можно, конечно, в тег каждого пункта меню вставить атрибут вида `onmouseover=""Func ( )`, где функция `Func ( )` описывает необходимые изменения свойств, используя Идентификатор объекта.

Более эффективный путь связан с возможностью определить объект DOM, который вызвал событие (в нашем случае событие `onmouseover` с помощью свойства объекта BOM `event`, которое носит имя `srcElement` или `target` (в зависимости от разновидности браузера)).

Свойство `event . srcElement (event, target)` определяет имя объекта DOM, который вызвал событие.

#### **Виды библиотек JavaScript . Подключение плагинов.**

##### **Технология jQuery**

Графическая технология HTML5 canvas («канвас» — в переводе «холст») включена в спецификацию HTML5 и поддерживается всеми современными браузерами. Из массово используемых версий браузеров ее не поддерживает только IE8.

Технология позволяет создавать прямоугольные области (холсты) на веб-страницах, а на холстах создавать рисунки, составляя программы для связанного с холстом компьютерного исполнителя.

Исполнитель холста очень похож на исполнители «Чертежник» И «Фломастер». Исполнитель холста является объектом языка JavaScript со своими свойствами и методами, поэтому построение графического изображения на холсте проводится командами языка JavaScript.

# ПРАКТИЧЕСКИЙ РАЗДЕЛ

## Тема 2. Алгоритмические структуры языка JavaScript

### Лабораторная работа 1.

#### Алгоритмические структуры языка JavaScript

Задание 1. Вычислить число символов в слове «Информатика» и вывести в окно браузера слово и число символов в нем. В программе задаем значение переменной

```
str="Информатика";
```

Вычисляем значение ее длины `n=str.length`;

Организуем вывод в окно браузера командой вывода с несколькими аргументами

```
document.writeln(str,"<br>","Длина = ",n);
```

Сохраняем и запускаем файл в браузере. Задание 2. Проверить действие оператора склейки.

В программе командой `var m1="Привет" + "всем!"`; сформируем величину, значение которой затем выведем в правое окно редактора командой

```
document.writeln(m1);
```

Сохраняем программу в файле и исполняем ее. Выводится значение "Приветвсем!". Чтобы слова не сливались, нужно в одной из констант добавить пробел, например, в первой:

```
var m1="Привет " + "всем!";
```

 Опять запускаем программу.

Для выделения фрагмента строки используется метод `substr()`.

Задание 3. С помощью метода `substr()` из букв слова "Информация" образовать слово "анимация" и вывести на экран оба слова.

Задание 4. Задано значение строковой величины. Сформировать другую строку, в которой первый и последний символы исходной строки поменялись местами.

Задание 5. В задании 4 задать исходное значение «краб».

Задание 6. Проверить работу методов преобразования чисел в строки и наоборот. В программе переменные `a` и `b` получают строковые значения. Третьей записана команда суммы чисел или склейки строк. Затем результат преобразуется в число и число 75 выводится на экран. Теперь удаляем кавычки в значении переменной `a`. Сохраняем и исполняем программу.

Задание 7. Задан набор чисел: 5, 8, -6, 3, 1, -7, 4, -5, -3, 2, -4. Вывести в окно браузера этот набор и число элементов в нем.

Задание 8. В программе задания 7 для заданного набора чисел добавить команды вычисления суммы `n` первых чисел набора для любого `n` от 2 до 10. Дополнительно вывести значение `n` и полученную сумму.

Задание 9. Заданы 7 слов: кувшинка, мак, лилия, одуванчик, роза, василек, фиалка. Составить программу поиска слова, которое начинается с буквы "р". Вывести элементы массива и найденное слово.

Задание 10. Создать программу, которая меняет порядок следования названий дней недели на обратный и выстраивает названия в алфавитном порядке. Вывести на экран названия дней недели в исходном порядке, в обратном и в алфавитном.

*Задания для самостоятельной работы.*

1. Массив задания 9 изменить так, чтобы кувшинка оказалась между лилией и одуванчиком, а фиалка — между маком и лилией. Вывести исходный массив и результат.

2. Массив задания 10 изменить так, чтобы в списке дней остались два вторника, среда, понедельник и воскресенье (именно в такой последовательности). Вывести исходный массив и результат.

### **Тема 3. События и CSS**

#### **Лабораторная работа 2.**

##### **События и CSS**

Задание 1. Создать изображение на зеленом фоне, в котором ниже текста "Пример использования формы с кнопками" желтого цвета расположена форма с тремя кнопками, имеющими названия "Первая", "Вторая", "Третья".

Задание 2. В условиях задания 1 добавить к каждой кнопке реакцию на щелчок указателем мыши (событие onclick).

Задание 3. Изменить задание 2 и записать реакции на события с использованием функций пользователя.

Задание 4. Вывести на экран значения свойства "Цвет фона" объекта body в задании 1. Для изменение значений свойств объектов модели DOM используют команду присваивания языка JavaScript вида:

*имя\_объекта.style.имя\_свойства\_на\_JS = "новое значение";*

Задание 5. Изменить программу задания 4 так, чтобы ответ на нажатие первой кнопки цвет фона страницы стал красным, в ответ на нажатие второй кнопки — белым, а третьей — зеленой.

Задание 6. Изменить текст программы для задания 1 так, чтобы в ответ на нажатие первой кнопки цвет текста над кнопками устанавливался красный, второй кнопки — черный, а фон текста белый, третьей кнопки — цвет текста желтый, а фон текста не был задан (т.е. прозрачный).

*Задания для самостоятельной работы.*

1. Измените программу решения так, чтобы надписи располагались посередине текстового поля.

2. Перепишите программу решения так, чтобы использовалась только одна функция пользователя, но с двумя аргументами.

3. Перепишите программу решения так, чтобы длина текстового поля каждый раз совпадала с длиной отображаемого текста.

4. Добавьте следующие действия: после нажатия на кнопку ее название меняется на "Нажата", а когда нажимается другая кнопка — исходное название возвращается.

5. Перепишите программу решения так, чтобы после нажатия первой кнопки в текстовом окне появлялись текущие координаты курсора на экране, второй — текущие координаты курсора в окне браузера, третьей — отображение координат выключалось.

6. Добавьте еще одно текстовое поле (над первым). Перепишите программу решения так, чтобы после нажатия первой кнопки текст, набранный с клавиатуры в верхнем поле, отображался в нижнем поле, а верхнее поле очищалось.

## **Тема 4. Структура языка JavaScript**

### **Лабораторная работа 3.**

#### **Объектная модель браузера (ВОМ)**

Задание 1. Создать изображение на зеленом фоне, в котором ниже текста "Пример использования формы с кнопками" желтого цвета расположена форма с тремя кнопками, имеющими названия "Первая", "Вторая", "Третья". организуя реакции кнопок, проверить действие методов `confirm()`, `resizeTo()`, `moveTo()` и `close()`.

Объект `document` имеет свойства:

`title` — текст названия страницы (между тегами `<title></title>`); `URL` — полный адрес текущего документа и метод:

`writeln()` — вывести данные в окно браузера.

Задание 2. В программе задания 1 проверить действие свойств объекта `document`.

Задание 3. В программе задания 1 проверить действие свойств объекта `screen`:

`width` — ширина экрана монитора в пикселах;

`height` — высота экрана монитора в пикселах;

`colorDepth` — глубина цветовой палитры (битов на пиксел).

Задание 4. В программе задания 1 проверить действие свойств объекта `navigator`:

`appName` — имя браузера;

`appVersion` — версия браузера;

`userAgent` — полное название браузера.

Задание 5. Используя программу задания 1, построить программу, которая в ответ на опускание курсор на фон веб-страницы, нажатие клавиши клавиатуры, наведение курсора на вторую кнопку формы выводит диалоговое окно с именем события.

Задание 6. Построить программу, в которой в ответ на щелчок курсором по внутренности окна браузера имя события появляется в заголовке страницы (на вкладке).

Задание 7. Используя программу задания 6, проверить свойства clientX и clientY объекта event.

*Задания для самостоятельной работы.*

1. По аналогии с заданием 7 проверьте свойства screenX и screenY объекта event, привязывая их вывод в заголовок страницы к событию onmousedown.

2. Проверьте свойства keyCode и button объекта event, привязывая их к событиям onkeydown и onmousedown, соответственно, при помощи разных функций с выводом кода в заголовок страницы в виде: "Код клавиши = XX" или "Код кнопки = X"

#### **Лабораторная работа 4.**

##### **Объектная модель документа (DOM)**

Задание 1. Создать изображение на зеленом фоне, в котором ниже текста "Пример использования формы с кнопками" желтого цвета расположена форма с пустым текстовым полем длиной 32 символа и тремя кнопками, имеющими названия "Первая", "Вторая", "Третья".

В ответ на нажатие первой кнопки в текстовом поле должна появиться надпись "Вы нажали на первую кнопку", при этом цвет фона текстового поля должен меняться на цвет кнопки. Аналогично и по остальным кнопкам.

*Задания для самостоятельной работы.*

1. Измените программу решения так, чтобы надписи располагались посередине текстового поля.

2. Перепишите программу решения так, чтобы использовалась только одна функция пользователя, но с двумя аргументами.

3. Перепишите программу решения так, чтобы длина текстового поля каждый раз совпадала с длиной отображаемого текста.

4. Добавьте следующие действия: после нажатия на кнопку ее название меняется на "Нажата", а когда нажимается другая кнопка — исходное название возвращается.

5. Перепишите программу решения так, чтобы после нажатия первой кнопки в текстовом окне появлялись текущие координаты курсора на экране, второй — текущие координаты курсора в окне браузера, третьей — отображение координат выключалось.

6. Добавьте еще одно текстовое поле (над первым). Перепишите программу решения так, чтобы после нажатия первой кнопки текст, набранный с клавиатуры в верхнем поле, отображался в нижнем поле, а верхнее поле очищалось.

Задание 2. Используя программу задания 1, создать программу со следующими функциями. В ответ на нажатие первой кнопки вместо текста "Пример использования формы с кнопками" в абзаце перед формой должна

появиться надпись "Вы нажали на первую кнопку" синего цвета, при этом цвет фона абзаца должен меняться на цвет кнопки. Аналогично и по остальным кнопкам.

*Задания для самостоятельной работы.*

1. Измените программу решения так, чтобы появляющиеся надписи имели разную степень насыщенности символов шрифта.

2. Измените программу решения так, чтобы текст абзаца не менялся, но разбивался второй кнопкой на две строки, третьей — на три, а первой — собирался опять в одну строку.

3. На экране два текстовых абзаца. Щелчок по любому из них должен обменять тексты абзацев.

Задание 3. Создать на экране с синим цветом фона изображение картинки из файла `pic1.jpg` с двумя кнопками "Первая" и "Вторая" под ним. Нажатие на вторую кнопку должно менять изображение на картинку из файла `pic2.jpg`, нажатие на первую — возвращает первую картинку (файлы с картинками должны быть в той же папке, что и исполняемый файл).

Задание 4. В условиях задания 2 удалить с экрана все кнопки и менять изображение щелчками по картинке.

Задание 5. В условиях задания 3 щелчок по картинке должен увеличивать размеры картинки вдвое и затем уменьшать их до исходных.

*Задания для самостоятельной работы.*

1. Измените программу решения задания 1 так, чтобы под картинками появлялись подписи "Картинка 1", "Картинка 2" (как текст абзаца).

2. Измените любую программу решения задания 2 так, чтобы под картинками появлялись подписи "Картинка 1", "Картинка 2" (как текст абзаца).

3. Измените любую программу решения задания 2 так, чтобы щелчками менялись последовательно 3 картинки.

4. Под одной картинкой две кнопки, всего источников 3 и они пронумерованы. Правая кнопка должна менять изображение в сторону увеличения номера картинки, левая — в сторону уменьшения. Кнопки не должны реагировать на щелчки, когда номер изменить уже нельзя.

5. Измените программу решения задания 3 так, чтобы размеры отображаемой картинки последовательно увеличивались в результате 5 щелчков, затем за один щелчок уменьшались до первоначальных.

6. Измените программу решения задания 3 так, чтобы размеры картинки последовательно увеличивались в результате 5 щелчков, затем щелчки переставали изменять размеры.

7. Измените программу решения задания 3 так, чтобы размеры картинки последовательно увеличивались в результате 5 щелчков, затем уменьшались до первоначальных размеров также за 5 щелчков.

8. Измените программу решения задания 3 так, чтобы под картинкой было две кнопки. Размеры картинки должны последовательно увеличиваться

в результате 5 щелчков по правой кнопке (затем щелчки по ней должны перестать изменять размеры). Щелчки по левой кнопке должны последовательно уменьшать картинку до исходных размеров в результате не более 5 щелчков. Исходные размеры щелчками уменьшить также нельзя.

9. Под картинкой четыре кнопки, всего источников 3. Две кнопки работают как в задании 4, еще две — как в задании 8.

### **Лабораторная работа 5.**

#### **Интерактивные эффекты**

Задание 1. Создать изображение на зеленом фоне, в котором ниже текста "Пример использования формы с кнопками" желтого цвета расположена форма с пустым текстовым полем длиной 32 символа и тремя кнопками, имеющими названия "Первая", "Вторая", "Третья". Чтобы увеличить количество объектов в окне, добавим ниже кнопок формы тега рисунка.

Проверить работу интерпретатора браузера по определению объекта, вызвавшего события `onmouseover` в окне. Имя тега, вызвавшего события показывать в заголовке страницы.

Задание 2. На базе программы задания 1 реализовать подсветку кнопки формы при наведении на нее указателя мыши. Пусть при наведении надпись на кнопке становится красной, а цвет фона кнопки меняется на желтый.

Задание 3. Разместить в окне браузера две картинки, текстовый абзац и закрашенный зеленым цветом прямоугольник, который можно построить при помощи тега `<div></div>`. Реализовать возможность перетаскивания по окну отображаемых объектов при помощи указателя мыши.

*Задания для самостоятельной работы.*

1). В программе решения задания 3 включить перетаскивание только для картинок, только для прямоугольника.

2). В задании 3 в начале перетаскивания объект мгновенно перемещается ("дергается") в фиксированную позицию относительно указателя мыши. Изменить текст программы так, чтобы объект при перетаскивании сохранял первоначальную позицию относительно курсора.

Задание 4. Разместить в задании 3 две картинки в одном разделе `<div></div>`. Реализовать возможность перетаскивания по окну отображаемых объектов при помощи указателя мыши, причем картинки перемещаются одновременно.

*Задание для самостоятельной работы.*

В задании 4 увеличьте вертикальный размер прямоугольника вдвое и задайте положение второй картинки в правом нижнем углу прямоугольника.

### **Лабораторная работа 6.**

#### **Автоматические эффекты**

Задание 1. В левом верхнем углу окна браузера заданы две кнопки Пуск и Стоп, а ниже и правее — голубой прямоугольник, внутри которого находится объект-картинка. Нажатие кнопки Пуск запускает горизонтальное

движение объекта вправо. Доходя до края прямоугольника, объект меняет движение на противоположное. Нажатие кнопки Стоп останавливает движение.

*Задания для самостоятельной работы.*

1. Измените текст программы решения задания 1 так, чтобы объект двигался по вертикали.

2. Измените текст программы решения задания 1 так, чтобы объект начинал двигаться вправо-вниз под углом  $45^\circ$ , а отражался от краев прямоугольника по схеме "угол падения равен углу отражения".

3. Измените текст программы решения задания 2 так, чтобы объект начинал двигаться по прямой  $x=dy$ .

Задание 2. В левом верхнем углу окна браузера зеленого цвета заданы две кнопки Пуск и Стоп, а ближе к центру — объект-картинка. Нажатие кнопки Пуск запускает движение объекта по замкнутой кривой. Нажатие кнопки Стоп останавливает движение.

Задание 3. В условиях задания 2 запустить на экране два объекта по замкнутой кривой одного и того же типа (эллипс), но с разными параметрами.

*Задания для самостоятельной работы.*

1. В условиях задания 2 в центр вращения подвижной картинки установите центр еще одной картинки (неподвижной).

2. В условиях задания 2 запустите картинку по кардиоиде:  $x = X_c + R_x(1 + \cos t)\cos t$ ;  $y = Y_c + R_x(1 + \cos t)\sin t$ .

3. В условиях задания 3 запустите одну картинку по кардиоиде, а вторую — по астроиде:  $x = X_c + R_x \cos 3t$ ;  $y = Y_c + R_y \sin 3t$ . Одноименные исходные данные для объектов должны совпадать, а  $R_x = R_y$ .

4. Сделайте в тексте программы задания 3 исходные данные у объектов равными, кроме  $dt_1$  и  $dt_2$ , которые должны различаться только знаками.

5. Измените текст программы задания 3 так, чтобы вращались три объекта с равными исходными данными, за исключением  $X_{c1}=200$ ,  $X_{c2}=400$ ,  $X_{c3}=600$ .

Задание 4. Создать изображение на зеленом фоне, в котором ниже текста "Пример использования формы с кнопками" желтого цвета расположена форма с пустым текстовым полем длиной 32 символа и тремя кнопками, имеющими названия "Первая", "Вторая", "Третья". По нажатию первой кнопки вывести в текстовое поле информацию о текущей дате и текущем времени, по нажатию второй кнопки — информацию о числе миллисекунд, а по нажатию третьей кнопки — информацию о времени и дате 11:23:34 7 октября 2013 года.

Задание 5. В условиях задания 4 по нажатию первой кнопки вывести в текстовое поле информацию о текущем часе, по нажатию второй кнопки — о текущем часе по Гринвичу, а по нажатию третьей кнопки — изменить текущий час на 23 и вывести информацию об измененном времени и дате.

Задание 6. В условиях задания 4 по нажатию первой кнопки вывести в текстовое поле информацию о текущем времени и дате, по нажатию второй кнопки текстовое поле должно очищаться, а по нажатию третьей кнопки — целое число, которое введено в текстовое поле, должно стать числом дней, которые добавляются к текущей дате, и новые данные о времени выводятся в текстовое поле.

Задание 7. В окне браузера три текстовых поля (одно под другим) и еще ниже две кнопки Вычислить и Очистить. В первые два поля с клавиатуры цифрами вводятся даты в формате "дд мм гггг". По нажатию кнопки Вычислить в первых двух полях даты преобразуются в стандартные форматы даты в браузере, а в третьем поле появляется число дней, разделяющих эти даты.

Задание 8. Вывести в отдельном прямоугольнике цифровые (текстовые) показания текущего времени в формате "часы : минуты". Разделительное двоеточие должно раз в полсекунды исчезать и появляться.

*Задания для самостоятельной работы.*

1. Изменить текст задания 4 так, чтобы по нажатию третьей кнопки — целое число, которое введено в текстовое поле, должно стать числом часов, которые добавляются к текущему времени, и новые данные о времени выводятся в текстовое поле.

2. Изменить текст задания 7 так, чтобы в первые два поля с клавиатуры цифрами вводились моменты времени в формате "чч:мм:сс". По нажатию кнопки Вычислить в первых двух полях время преобразовалось в стандартные форматы времени в браузере, а в третьем поле появляется число минут, разделяющих эти моменты времени.

3. Изменить текст задания 8 так, чтобы часы отображали еще и секунды.

4. Используя конструкции для перетаскивания, изменить текст задания 8 так, чтобы построенные цифровые часы можно было перетаскивать по окну браузера.

## РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ

1. Основы JavaScript.
2. Объектно-событийная модель JavaScript. Типы данных. Объекты. События.
3. Базовые алгоритмические конструкции языка JavaScript: If Else switch
4. Базовые алгоритмические конструкции языка JavaScript while, do...while, for, for...in.
5. Встроенные объекты языка JavaScript. Объект Math.
6. Массивы в языке JavaScript.
7. Объект Array. Работа с массивами. JavaScript.
8. Объект String. Работа со строками JavaScript.
9. Объект Date. Работа с датой и временем.
10. Создание и использование функций и объектов в языке JavaScript.
11. Формат JSON
12. Объектная модель браузера. Объекты Navigator, Screen.
13. Получение информации о пользователе.
14. Объектная модель документа (DOM). Доступ к объектам web-Страницы средствами JavaScript.
15. Объект Window.
16. Свойства и методы.
17. Управление окнами. JavaScript.
18. Объект Document.
19. Свойства и методы.
20. Создание динамических страниц.
21. Разработка интерактивных web-ресурсов.
22. Формы. Методы форм:
23. Get, Post
24. Обработка форм на web
25. Страницы средствами JavaScript
26. Элементы форм: Input.
27. Проверка вводимых данных.
28. Обработка форм на web-странице средствами JavaScript.
29. Элементы форм: Textarea, Select.
30. Управление объектами браузера средствами JavaScript.
31. Динамические эффекты.
32. Регулярные выражения
33. Использование регулярных выражений в языке JavaScript
34. Графические возможности HTML5
35. Элемент Canvas.
36. API рисования на холсте.
37. Основные фигуры. Работа с растровыми изображениями.

- 38. Масштабируемая векторная графика SVG.
- 39. Основные элементы.
- 40. Атрибуты
- 41. Стили. Интерактивные эффекты
- 42. Использование JavaScript для разработки образовательных ресурсов. Простые тесты.

РЕПОЗИТОРИЙ БГПУ

## ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

### ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа дисциплины по выбору студента «Основы программирования на языке JavaScript» составлена для студентов физико-математического факультета в соответствии с требованиями образовательного стандарта высшего образования специальности 1-02 05 01 Математика и информатика.

Изучение дисциплины предусматривает знакомство с основами программирования на языке JavaScript с целью реализации интерактивных возможностей веб-программирования. Студенты познакомятся с технологиями создания интерактивных динамических веб-страниц разного типа. В результате изучения содержания данной дисциплины по выбору студенты смогут создавать свои интерактивные динамические веб-страницы различного назначения, а также виртуальные модели различных физических и информационных процессов. Возможности и опыт, приобретаемые студентами в процессе изучения данной учебной дисциплины, позволят им осуществлять междисциплинарные практико-ориентированные связи, поддерживать идеи профильного обучения в школе, развивать олимпиадное движение и исследовательскую деятельность по информатике, а также оптимизировать организационно-управленческий потенциал учебного процесса в учреждениях системы образования.

**Цель учебной дисциплины** – развитие алгоритмического мышления студентов, развитие у студентов целостного представления о возможностях использования языка программирования JavaScript в процессе разработки современных интерактивных динамических веб-страниц.

#### **Основные задачи учебной дисциплины:**

- совершенствование навыков самостоятельности при разработке программного обеспечения;
- освоение возможностей языка программирования JavaScript;
- формирование понятий о семантике, синтаксисе, стандартах и структуре языка JavaScript;
- освоение программного обеспечения для отладки и тестирования кода на языке JavaScript;
- освоение приемов разработки интерактивных динамических веб-страниц средствами языка JavaScript;
- освоение студентами современных подходов и методов создания интерактивных динамических веб-страниц средствами языка JavaScript.

#### ***Место учебной дисциплины в системе подготовки специалиста***

Изучение учебной дисциплины по выбору «**Основы программирования на языке JavaScript**» опирается на основные академические, социально-личностные и профессиональные компетенции, сформированные у студентов в процессе изучения ими таких учебных

дисциплин как «Технологии программирования и методы алгоритмизации», «Вычислительные методы и компьютерное моделирование», «Практикум по решению задач по информатике». Благодаря ее изучению и приобретению умений создавать интерактивные динамические веб-страницы студенты смогут реализовать свои профессиональные потребности на современном уровне.

### **Профессиональные компетенции студентов**

Учебная дисциплина по выбору «**Основы программирования на языке JavaScript**» входит в компонент учреждения высшего образования. Изучение дисциплины по выбору студента «Основы программирования на языке JavaScript» должно обеспечить формирование у студентов академических, социально-личностных и профессиональных компетенций.

#### ***Требования к академическим компетенциям***

Студент должен:

- АК–3. Владеть исследовательскими навыками;
- АК–4. Уметь работать самостоятельно; – АК–5. Быть способным порождать новые идеи (обладать креативностью);
- АК–6. Владеть междисциплинарным подходом при решении проблемы;
- АК–7. Иметь навыки, связанные с использованием технических средств устройств, управлением информацией и работой с компьютером;
- АК–9. Уметь учиться, повышать свою квалификацию в течение всей жизни;
- АК–10. Уметь осуществлять учебно-исследовательскую деятельность. Требования к социально-личностным компетенциям Студент должен:

- СЛК–3. Обладать способностью к межличностным коммуникациям;
- СЛК–7. Быть способным к осуществлению самообразования и самосовершенствования профессиональной деятельности.

#### ***Требования к профессиональным компетенциям***

Студент должен быть способен:

##### **Обучающая деятельность**

- ПК–1. Управлять учебно-познавательной и учебно-исследовательской деятельностью обучающихся;
- ПК–2. Использовать оптимальные методы, формы и средства обучения;
- ПК–3. Организовывать и проводить учебные занятия различных видов;
- ПК–4. Организовывать самостоятельную работу учащихся.

##### **Воспитательная деятельность**

- ПК–8. Формировать базовые компоненты культуры личности воспитанника. Развивающая деятельность

– ПК–12. Развивать навыки самостоятельной работы обучающихся с учебной, справочной, научной литературой и др. источниками информации.

### **Ценностно-ориентационная деятельность**

– ПК–16. Оценивать учебные достижения учащихся, а также уровни их воспитанности и развития;

– ПК–17. Осуществлять профессиональной самообразование и самовоспитание с целью совершенствования профессиональной деятельности.

В результате изучения учебной дисциплины студент должен знать:

– понятия, связанные с семантикой, синтаксисом, стандартами и структурой языка JavaScript;

– программное обеспечение для отладки и тестирования программ на языке JavaScript;

– методы языка JavaScript, реализующие интерактивность и динамичность на веб-страницах; – приемы создания интерактивных динамических веб-страниц средствами языка JavaScript.

В результате изучения учебной дисциплины студент должен уметь:

– обрабатывать цифровую, текстовую и графическую информацию средствами языка JavaScript;

– разрабатывать интерактивные динамические веб-страницы средствами языка JavaScript.

В результате изучения дисциплины по выбору студент должен владеть:

– методами поиска, анализа и дидактической адаптации научной информации по решению задач создания интерактивных веб-страниц;

– практическими умениями применять полученные знания в нестандартных ситуациях науки и жизни;

– современными технологиями и средствами для решения профессиональных задач.

### **Структура и содержание учебной дисциплины**

Учебная дисциплина изучается на протяжении одного семестра, и ее содержание включает пять тем. Первая тема посвящена основным понятиям языка программирования JavaScript. Вторая тема связана с изучением основных алгоритмических структур языка. Третья тема посвящена событийному программированию и средствам CSS. Четвертая тема раскрывает структуру языка JavaScript. Пятая тема связана с освоением различных приемов создания динамических веб-страниц.

Данная учебная программа является основным документом, определяющим объем и содержание учебной дисциплины по выбору «Основы программирования на языке JavaScript» для специальности 1-02 05 01 Математика и информатика.

### **Методы обучения**

Обучение учебной дисциплине проходит в рамках организации лекционных и лабораторных занятий. При чтении лекций особое внимание следует уделять использованию мультимедийных технологий для представления графических иллюстраций и демонстрации приемов работы с программными средствами.

Организация лабораторных занятий предполагает использование личностно-ориентированных методов обучения, основанных на подготовке обширных методических рекомендаций, что способствует развитию индивидуально-творческих способностей студентов и приобретению ими умений самостоятельной работы.

Содержание самостоятельной работы студентов разрабатывается в соответствии с задачами изучения дисциплины. Формы самостоятельной работы могут обладать достаточным разнообразием: самостоятельная работа по программированию и отладке интерактивных веб-страниц во время основных аудиторных занятий (лекций и лабораторных занятий), самостоятельная работа в форме консультаций, в форме внеаудиторной работы. Для управления самостоятельной работой рекомендуется использовать электронные средства обучения, тестирующие программы. Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

#### ***Распределение общего количества часов по формам обучения и семестрам***

##### **Дневная форма получения высшего образования:**

Всего на учебную дисциплину – 50 часов.

8 семестр – 26 часов аудиторных (14 часов – лекции, 12 часов – лабораторные занятия), 24 часа – самостоятельная работа.

Форма контроля – зачет (8 семестр).

##### **Заочная форма получения образования:**

Всего на учебную дисциплину – 50 часов.

9 семестр – 8 часов аудиторных (4 часа – лекции, 4 часа – лабораторные занятия), 42 часа – самостоятельная работа.

Форма контроля – зачет (10 семестр).

## **СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА**

### **Тема 1. Основные понятия языка JavaScript**

Области применения, особенности и возможности скриптовых языков программирования. Инструменты для отладки и тестирования кода JavaScript. Встраивание программного кода JavaScript в веб-страницы. Семантика, синтаксис, стандарты языка JavaScript.

### **Тема 2. Алгоритмические структуры языка JavaScript**

Структура программы на языке JavaScript. Организация повторений и ветвлений. Объекты и функции.

### **Тема 3. События и CSS События в JavaScript.**

JavaScript и CSS. Особенности поддержки браузерами.

### **Тема 4. Структура языка JavaScript**

Составные части языка JavaScript. Ядро языка JavaScript. Объектная модель браузера (BOM). Доступ к универсальным объектам модели BOM. Объектная модель документа (DOM). Доступ к свойствам объектов документа.

### **Тема 5. Приемы создания динамических страниц**

Интерактивные эффекты. Работа с формами. Перетаскивание объектов. Автоматические эффекты. Автоматическое движение объектов. Работа с датой и временем.

**УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА**  
**для специальности 1-02 05 01 Математика и информатика**  
**для дневной формы получения образования**

| Номер раздела, темы | Название раздела, темы                                                                                                                                                                                | Количество аудиторных часов |                      |                      |      | Количество часов самостоятельной работы | Формы контроля знаний        |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|----------------------|----------------------|------|-----------------------------------------|------------------------------|
|                     |                                                                                                                                                                                                       | лекции                      | практические занятия | лабораторные занятия | Иное |                                         |                              |
| 1                   | 2                                                                                                                                                                                                     | 3                           | 4                    | 5                    | 6    | 7                                       | 8                            |
| 1.                  | <b>Основные понятия языка JavaScript</b>                                                                                                                                                              | 4                           |                      |                      |      | 4                                       |                              |
| 1.1.                | <b>Области применения, особенности и возможности скриптовых языков программирования.</b><br>1. Инструменты для отладки и тестирования кода JavaScript.<br>2. Встраивание программного кода JavaScript | 2                           |                      |                      |      | 2                                       | Устный опрос                 |
| 1.2.                | <b>Семантика, синтаксис, стандарты языка JavaScript.</b><br>1. Семантика языка JavaScript.<br>2. Синтаксис языка JavaScript.<br>3. Стандарты языка JavaScript.                                        | 2                           |                      |                      |      | 2                                       | Устный опрос                 |
| 2.                  | <b>Алгоритмические структуры языка JavaScript</b>                                                                                                                                                     | 2                           |                      | 2                    |      | 4                                       |                              |
| 2.1.                | <b>Алгоритмические структуры языка JavaScript</b><br>1. Структура программы на языке JavaScript.<br>2. Организация повторений и ветвлений.<br>3. Объекты и функции.                                   | 2                           |                      | 2                    |      | 4                                       | Проверка лабораторной работы |

| 1        | 2                                                                                                                                                              | 3         | 4 | 5         | 6 | 7         | 8                            |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|---|-----------|---|-----------|------------------------------|
| 3.       | <b>События и CSS</b>                                                                                                                                           | <b>2</b>  |   | <b>2</b>  |   | <b>4</b>  |                              |
| 3.1.     | <b>События и CSS</b><br>1. События в JavaScript.<br>2. JavaScript и CSS.<br>3. Особенности поддержки браузерами.                                               | 2         |   | 2         |   | 4         | Проверка лабораторной работы |
| 4.       | <b>Структура языка JavaScript</b>                                                                                                                              | <b>4</b>  |   | <b>4</b>  |   | <b>6</b>  |                              |
| 4.1.     | <b>Составные части языка JavaScript.</b><br>1. Ядро языка JavaScript.<br>2. Объектная модель браузера (BOM).<br>3. Доступ к универсальным объектам модели BOM. | 2         |   | 2         |   | 3         | Проверка лабораторной работы |
| 4.2.     | <b>Объектная модель документа (DOM).</b><br>1. Доступ к свойствам объектов документа.                                                                          | 2         |   | 2         |   | 3         | Проверка лабораторной работы |
| <b>5</b> | <b>Приемы создания динамических страниц</b>                                                                                                                    | <b>2</b>  |   | <b>4</b>  |   | <b>6</b>  |                              |
| 5.1.     | <b>Интерактивные эффекты</b><br>1. Работа с формами.<br>2. Перетаскивание объектов.                                                                            | 2         |   | 2         |   | 3         | Проверка лабораторной работы |
| 5.2.     | <b>Автоматические эффекты</b><br>1. Автоматическое движение объектов.<br>2. Работа с датой и временем.                                                         |           |   | 2         |   | 3         | Проверка лабораторной работы |
|          | <b>Итого: 50 ч.</b>                                                                                                                                            | <b>14</b> |   | <b>12</b> |   | <b>24</b> | <b>зачёт</b>                 |

**УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА**  
**для специальности 1-02 05 01 Математика и информатика**  
**для заочной формы получения образования**

| Номер раздела, темы | Название раздела, темы                                                                                                                                                                                | Количество аудиторных часов |                      |                      |      | Количество часов самостоятельной работы | Формы контроля знаний |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|----------------------|----------------------|------|-----------------------------------------|-----------------------|
|                     |                                                                                                                                                                                                       | лекции                      | практические занятия | лабораторные занятия | Иное |                                         |                       |
| 1                   | 2                                                                                                                                                                                                     | 3                           | 4                    | 5                    | 6    | 7                                       | 8                     |
| 1.                  | <b>Основные понятия языка JavaScript</b>                                                                                                                                                              | 2                           |                      |                      |      | 6                                       |                       |
| 1.1.                | <b>Области применения, особенности и возможности скриптовых языков программирования.</b><br>1. Инструменты для отладки и тестирования кода JavaScript.<br>2. Встраивание программного кода JavaScript | 2                           |                      |                      |      | 2                                       |                       |
| 1.2.                | <b>Семантика, синтаксис, стандарты языка JavaScript.</b><br>1. Семантика языка JavaScript.<br>2. Синтаксис языка JavaScript.<br>3. Стандарты языка JavaScript.                                        |                             |                      |                      |      | 4                                       |                       |
| 2.                  | <b>Алгоритмические структуры языка JavaScript</b>                                                                                                                                                     |                             |                      | 2                    |      | 6                                       |                       |
| 2.1                 | <b>Алгоритмические структуры языка JavaScript</b><br>1. Структура программы на языке JavaScript.<br>2. Организация повторений и ветвлений.<br>3. Объекты и функции.                                   |                             |                      | 2                    |      | 6                                       |                       |

| 1        | 2                                                                                                                                                              | 3        | 4 | 5        | 6 | 7         | 8            |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---|----------|---|-----------|--------------|
| 3.       | <b>События и CSS</b>                                                                                                                                           |          |   | <b>2</b> |   | <b>6</b>  |              |
| 3.1      | <b>События и CSS</b><br>1. События в JavaScript.<br>2. JavaScript и CSS.<br>3. Особенности поддержки браузерами.                                               |          |   | 2        |   | 6         |              |
| 4.       | <b>Структура языка JavaScript</b>                                                                                                                              | <b>2</b> |   |          |   | <b>12</b> |              |
| 4.1      | <b>Составные части языка JavaScript.</b><br>1. Ядро языка JavaScript.<br>2. Объектная модель браузера (BOM).<br>3. Доступ к универсальным объектам модели BOM. | 2        |   |          |   | 6         |              |
| 4.2      | <b>Объектная модель документа (DOM).</b><br>1. Доступ к свойствам объектов документа.                                                                          |          |   |          |   | 6         |              |
| <b>5</b> | <b>Приемы создания динамических страниц</b>                                                                                                                    |          |   |          |   | <b>12</b> |              |
| 5.1      | <b>Интерактивные эффекты</b><br>1. Работа с формами.<br>2. Перетаскивание объектов.                                                                            |          |   |          |   | 6         |              |
| 5.2      | <b>Автоматические эффекты</b><br>1. Автоматическое движение объектов.<br>2. Работа с датой и временем.                                                         |          |   |          |   | 6         |              |
|          | <b>Итого: 50 ч.</b>                                                                                                                                            | <b>4</b> |   | <b>4</b> |   | <b>42</b> | <b>зачёт</b> |

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### Основная литература:

1. Флэнаган, Д. JavaScript. Подробное руководство / Д.Флэнаган. – 6-е изд. – СПб.: Издательство «Символ-плюс», 2012. – 1080 с.
2. Флэнаган, Д. JavaScript. Карманный справочник / Д.Флэнаган. – 3-е изд. – М.: Издательство «Вильямс», 2015. – 320 с.
3. Роббинс, Д. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Д. Роббинс. – 4-е изд. – М.: Издательство «Эксмо», 2014. – 516 с.
4. Быкадоров, Ю.А. Информатика и ИКТ. 9 кл.: учеб. для общеобразовательных учреждений / Ю.А. Быкадоров. – М.: Дрофа, 2013. – 319 с.

### Дополнительная литература

5. Моррисон, М. Изучаем JavaScript / М.Моррисон. – СПб.: Издательство «Питер», 2012. – 272 с.
6. Беляев, С.А. Разработка игр на языке JavaScript: учебное пособие / – СПб.: Издательство «Лань», 2016. – 128 с.
7. Стефанов С. JavaScript. Шаблоны / С.Стефанов. – СПб.: Издательство «Символ-плюс», 2011. – 272 с.
8. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа <https://learn.javascript.ru/>. – Дата доступа: 30.06.1916.

### ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

| №      | Наименование разделов, тем                 | Всего | Лекции | Лабо<br>ратор<br>ные |
|--------|--------------------------------------------|-------|--------|----------------------|
| 1.     | Основные понятия языка JavaScript          | 4     | 4      |                      |
| 2.     | Алгоритмические структуры языка JavaScript | 4     | 2      | 2                    |
| 3.     | События и CSS                              | 4     | 2      | 2                    |
| 4.     | Структура языка JavaScript                 | 8     | 4      | 4                    |
| 5.     | Приемы создания динамических страниц       | 6     | 2      | 4                    |
| Итого: |                                            | 26    | 14     | 12                   |

## **Материалы на электронных носителях**

(сайт физико-математического факультета, локальная сеть физико-математического факультета, кафедральные компьютеры)

1. Тексты методических указаний к лабораторным работам.
2. Вопросы к зачету.
3. Задания для самостоятельной работы

## **Методические рекомендации по организации и выполнению самостоятельной работы студентов**

Содержание и формы самостоятельной работы студентов разрабатываются кафедрами в соответствии с целями и задачами подготовки специалиста. Для управления самостоятельной работой рекомендуется использовать электронные средства обучения, тестирующие программы. Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

### **Перечень используемых средств диагностики результатов учебной деятельности**

Основным средством диагностики усвоения знаний, умений и овладения необходимыми навыками по учебной дисциплине являются:

- фронтальный опрос на лекционных занятиях, направлен систематизацию знаний студентов, определение уровня готовности аудитории к восприятию нового материала, а также на формирование у преподавателя представление об усвоении студентами основополагающих понятий и фактов изучаемой учебной дисциплины;

- проверка заданий разнообразного типа (рецептивные, репродуктивные, продуктивные, творческие), выполняемых в рамках часов, отводимых на учебные занятия (лабораторные), представляет собой диагностику систематичности подготовки студентов к занятиям, уровень усвоения ими практико-ориентированного содержания программного материала учебной дисциплины;

- групповые и индивидуальные консультации студентов предназначены для диагностики уровня овладения определенными знаниями, умениями и навыками, как теоретического материала, так и практического; устранения типичных ошибок и пробелов в знаниях обучающихся;

- самостоятельные работы используются для определения индивидуальных особенностей, темпа продвижения студентов и усвоения ими необходимых знаний;

- компьютерное тестирование позволяет относительно быстро провести диагностику усвоения студентами учебного материала как по

отдельным темам и разделам учебной дисциплины, так и по учебной дисциплине в целом;

– зачет используется для осуществления итоговой диагностики усвоения учащимися содержания учебной дисциплины за учебный семестр и оценивается обычно в форме «зачтено» или «не зачтено» в соответствии с критериями оценки результатов учебной деятельности обучающихся в учреждениях высшего образования по десятибалльной шкале.

#### ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ

| Название учебной дисциплины, с которой требуется согласование | Название кафедры                                      | Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине                                                                                                       | Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола) |
|---------------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Методы изображения фигур и основания геометрии                | Кафедра математики и методики преподавания математики | При рассмотрении вопросов, связанных с решением практико-ориентированных задач, использовать согласованную терминологию в соответствии с действующими учебными пособиями для учреждений общего среднего образования | Протокол № 10 от 26.05.2016                                                                       |