



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ И РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)**

«СОГЛАСОВАНО»  
Руководитель ОП

 Пак Т.В.

«УТВЕРЖДАЮ»  
Директор департамента Математического  
и компьютерного моделирования  
Сущенко А.А.  
«25» марта 2022г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

Интернет-технологии

Направление подготовки **02.03.01 Математика и компьютерные науки**

(Сквозные цифровые технологии)

**Форма подготовки очная**

курс 2 семестр 3  
лекции 00 час.  
практические занятия 00 час.  
лабораторные работы 32 час.  
в том числе с использованием МАО лек. час./ пр. час./ лаб. 18 час  
всего часов аудиторной нагрузки 32 час.  
самостоятельная работа 40 час.  
в том числе на подготовку к экзамену 36 час.  
контрольные работы (количество) 1  
курсовая работа / курсовой проект не предусмотрены  
зачет не предусмотрен  
экзамен 3 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 02.03.01 Математика и компьютерные науки, утвержденного приказом Министерства образования и науки Российской Федерации от 23 августа 2017 № 807 (с изменениями и дополнениями)

Рабочая программа обсуждена на заседании департамента математического и компьютерного моделирования, протокол № 6 от «05» марта 2022 г.

Директор департамента математического и компьютерного моделирования Сущенко А. А.  
Составитель: доцент Пак Т. В.

Владивосток  
2022

**Оборотная сторона титульного листа РПД**

**I. Рабочая программа пересмотрена на заседании департамента:**

Протокол от « \_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Директор департамента \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**II. Рабочая программа пересмотрена на заседании департамента:**

Протокол от « \_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Директор департамента \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**III. Рабочая программа пересмотрена на заседании департамента:**

Протокол от « \_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Директор департамента \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**IV. Рабочая программа пересмотрена на заседании департамента:**

Протокол от « \_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Директор департамента \_\_\_\_\_  
(подпись) (И.О. Фамилия)

## I. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ:

**Цель** дисциплины «Интернет-технологии» является изучение методологических и концептуальных теоретических сведений о интернет технологиях формирование у студентов умения и навыков работы с WEB-страницами и эффективного комбинирования элементов мультимедиа, а также подготовка специалистов, умеющих применять современные методики разработки и сопровождения WEB-сайтов, используемых в дальнейшей профессиональной деятельности. Задачей дисциплины является изучение основ аппаратных средств WEB-программирования, основных инструментальных средств, используемых для создания WEB-страниц, знакомство с возможностями создания базовых элементов WEB-страниц (текст, графические изображения, звук, анимация), с возможностями применения информационных технологий в сети Интернет.

### **Задачи:**

- ознакомить с основами web-дизайна и программирования;
- ознакомить с основами проектирования сайтов и технологии проектирования;
- сформировать основные навыки программирования сайтов различными программными средствами.

Для успешного изучения дисциплины необходимо сформировать следующие компетенции:

-способность использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач

В результате изучения данной дисциплины у обучающихся формируются следующие общекультурные/ общепрофессиональные/ профессиональные компетенции (элементы компетенций).

Профессиональные компетенции выпускников и индикаторы их достижения:

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
производственно-технологический	<b>ПК-5</b> Способен к формированию технической отчетной документации и разработке технических документов	ПК-5.1 определяет основные стандарты, нормы и правила разработки технической документации программных продуктов и программных комплексов
		ПК-5.2 составляет техническую документацию на различных этапах жизненного цикла информационных систем, программных продуктов
		ПК-5.3 разрабатывает тестовые документы, включая план тестирования, разрабатывает стратегии тестирования и управление процессом тестирования

Для формирования вышеуказанных компетенций в рамках дисциплины «Интернет-технологии» применяются следующие методы активного/интерактивного обучения:

- презентации с использованием доски, книг, видео, слайдов, компьютеров и т.п., с последующим обсуждением материалов,
- обратная связь с формированием общего представления об уровне владения знаниями студентов, актуальными для занятия,
- разминка с вопросами, ориентированными на выстраивание логической цепочки из полученных знаний (конструирование нового знания),
- работа в малых группах (дает всем студентам возможность участвовать в работе, практиковать навыки сотрудничества, межличностного общения).

## II. ТРУДОЁМКОСТЬ ДИСЦИПЛИНЫ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ ПО ДИСЦИПЛИНЕ

Общая трудоемкость дисциплины составляет 3 зачётные единицы (108 академических часа).

(1 зачетная единица соответствует 36 академическим часам)

Видами учебных занятий и работы обучающегося по дисциплине могут являться:

Обозначение	Виды учебных занятий и работы обучающегося
Лек	Лекции
Лаб	Лабораторные работы
Пр	Практические занятия
ОК	Онлайн курс
СР	Самостоятельная работа обучающегося в период теоретического обучения
Контроль	Самостоятельная работа обучающегося и контактная работа обучающегося с преподавателем в период промежуточной аттестации

Структура дисциплины:

Форма обучения – очная.

№	Наименование раздела дисциплины	Семестр	Количество часов по видам учебных занятий и работы обучающегося					Формы промежуточной аттестации	
			Лек	Лаб	Пр	ОК	СР		Контроль
1	Работа с регулярными выражениями. Проверка корректности ввода текстовых данных	3		6			40	36	экзамен
2	Работа с таймером			6					
3	Работа с cookies			6					
4	Перемещение слоев			6					
5	Калькулятор			8					

Итого:		32		40	36	
--------	--	----	--	----	----	--

### III. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Теоретическая часть не предусмотрена учебным планом.

### IV. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА И САМОСТОЯТЕЛЬНОЙ РАБОТЫ

**Лабораторная работа № 1. Работа с регулярными выражениями. Проверка корректности ввода текстовых данных** Понятие регулярного выражения. Основные методы. Проверка корректности ввода данных.

**Лабораторная работа № 2. Работа с таймером** Понятие таймера. Создание скрипта, реализующего электронные часы. Создание скрипта, реализующего электронный будильник.

**Лабораторная работа № 3. Работа с cookies** Понятие cookies. Работа с cookies. Статистика посещения страницы.

**Лабораторная работа № 4. Перемещение слоев** Понятие слоя. Объект event. Обработка событий. Перемещение слоев.

**Лабораторная работа № 5. Калькулятор** Обработка содержимого форм. Объект Math. Реализация калькулятора.

### V. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ (И ОНЛАЙН КУРСА ПРИ НАЛИЧИИ)

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Интернет-технологии» включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- методические рекомендации по выполнению заданий для самостоятельной работы;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

#### План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	1–3 недели	Лабораторная работа	6 час.	Письменный отчет

		№1		
2	4-7 недели	Лабораторная работа №2	6 час.	Письменный отчет
3	8–11 недели	Лабораторная работа №3	6 час.	Письменный отчет
4	12–14 недели	Лабораторная работа №4	6 час.	Письменный отчет
5	15–18 недели	Лабораторная работа №5	7 час.	Письменный отчет

### **Рекомендации по самостоятельной работе студентов**

Самостоятельная работа включает подготовку к выполнению индивидуальных лабораторных заданий, подготовку к зачету, экзамену.

### **Рекомендации по составлению отчетов**

В процессе подготовки отчетов к лабораторным работам у студентов развиваются навыки составления письменной документации и систематизации имеющихся знаний, что, несомненно, пригодится им в дальнейшей учебной и профессиональной деятельности. При составлении отчетов настоятельно рекомендуется придерживаться следующей структуры:

- 1) Постановка задачи;
- 2) Спецификация используемых функций и типов данных;
- 3) Описание тестов, на которых программа проходила проверку.

### **Критерии оценивания самостоятельных работ**

Результаты лабораторной работы оцениваются по трехбалльной шкале:

1. Решение частично удовлетворяет условию задачи, проходит большую часть тестов, однако требует существенной доработки;
2. Решение полностью удовлетворяет условию задачи, проходит все тесты, однако имеет ряд недостатков, требующих некоторой доработки;

Решение полностью удовлетворяет условию задачи, проходит все тесты и не требует дальнейшей доработки.

## **VI. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА**

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
1	Лабораторная работа №1	ПК-5	Знает	Письменный отчет	зачет
			Умеет	Письменный отчет	зачет
			Владеет	Письменный отчет	зачет
2	Лабораторная работа №2	ПК-5	Знает	Письменный отчет	зачет
			Умеет	Письменный отчет	зачет
			Владеет	Письменный отчет	зачет

3	Лабораторная работа №3	ПК-5	Знает	Письменный отчет	зачет
			Умеет	Письменный отчет	зачет
			Владеет	Письменный отчет	зачет
4	Лабораторная работа №4	ПК-5	Знает	Письменный отчет	зачет
			Умеет	Письменный отчет	зачет
			Владеет	Письменный отчет	зачет
5	Лабораторная работа №5	ПК-5	Знает	Письменный отчет	зачет
			Умеет	Письменный отчет	зачет
			Владеет	Письменный отчет	зачет

## **VII. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **Основная литература**

#### **(электронные и печатные издания)**

1. Алексеев, А.П. Введение в WEB-дизайн: учеб. пособие для вузов / А.П. Алексеев. – М.: СОЛОН-ПРЕСС, 2008. – 184 с.
2. Веллинг, Л. Разработка Web-приложений с помощью PHP и MySQL. / Л. Веллинг, Л. Томсон; 3-е изд.: пер. с англ. – М.: Издательский дом «Вильямс», 2008. – 880 с.
3. Вьюхин, В.В. Базы данных: учеб. пособие для вузов. Ч. 1. Лабораторный практикум / В.В. Вьюхин, С.В. Супрун, Т.А. Кочнева. – Екатеринбург: Изд-во РГППУ, 2005. – 66 с.
4. Евсеев, Д.А. Web-дизайн в примерах и задачах: учеб. пособие / Д.А. Евсеев, В.Р. Трофимов; Под. ред. В.В. Трофимова. – М.: КНОРУС, 2010. – 272 с.
5. Котеров, Д.В. PHP 5 / Д.В. Котеров, А.Ф. Костарев. – 2-е изд., перераб. и доп. – СПб.: БХВ Петербург, 2008. – 1104 с.
6. Рейсинг, Джон. JavaScript. Профессиональные приемы программирования: учеб. пособие / Джон Рейсинг. – СПб.: Питер, 2008. – 351 с.
7. Фролов И.К. Разработка, дизайн, программирование и раскрутка Web-сайта: И.К. Фролов, В.А. Перелыгин, Е.Э. Самойлов. – М.: Триумф, 2009. – 304 с.
8. Чиртик, А.А. HTML: А.А. Чиртик. – Спб. : Питер, 2006. – 219 с.

### **Дополнительная литература**

1. Action Script 3.0 для Adobe Flash CS4 Professional: официальный учебный курс. – М.: Эксмо, 2009. – 400 с.
2. Аксак, В.А. Новейшая энциклопедия Интернет 2007 [Текст]:

всеобъемлющее руководство по эффективному использованию Сети / В.А. Аксак; ред. В.В. Александров. – М.: ЭКСМО, 2007. – 912 с.

3. Биндер, Кейт. Adobe Photoshop CS4 за 24 часа/ Кейт Биндер; Пер. с англ. – М.: ООО «И.Д. Вильямс», 2010. – 528 с.

4. Венедюхин, А.А. Создание сайтов. Новейшая энциклопедия/ А.А. Венедюхин, А.А. Воробьев. – М.: Эксмо, 2009. – 416 с.

5. Глушаков, С.В. CorelDRAW X4/ С.В. Глушаков, Е.В. Гончарова. – 3-е изд. доп. и перераб. – М.: АСТ, 2008. – 476 с.

6. Дунаев, В.В. Самоучитель JavaScript: учеб. пособие / В.В. Дунаев. – 3-е изд. – СПб.: Питер, 2008. – 400 с.

7. Котеров, Д.В. PHP 5 / Д.В. Котеров, А.Ф. Костарев. – 2-е изд., перераб. и доп. – СПб.: БХВ Петербург, 2008. – 1104 с.

8. Третьяк, Т.М. Практикум Web-дизайна: практикум / Т.М. Третьяк, М.В. Кубарева. – М.: СОЛОН-ПРЕСС, 2007. – 173 с.

## **VIII. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Для выполнения лабораторных работ следует ознакомиться с рекомендуемой литературой, не пренебрегать советами преподавателя и задавать ему, интересующие Вас, вопросы.

## **IX. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Для обеспечения освоения дисциплины необходим класс ПК с доступом в Интернет. Программное обеспечение: Веб-сервер Apache с поддержкой PHP, СУБД MySQL.

Образовательный процесс по дисциплине проводится в лекционных и компьютерных аудиториях.

Мультимедийная лекционная аудитория (мультимедийный проектор, настенный экран, документ-камера) о. Русский, кампус ДВФУ, корпус 20(D), ауд. D738, D654/D752, D412/D542, D818, D741, D945, D547, D548, D732

Компьютерные классы: (доска, 15 персональных компьютеров) о. Русский, кампус ДВФУ, корпус 20(D), D733, D733а, D734, D734а, D546, D546а, D549а (Кампус ДВФУ), оснащенные компьютерами класса Pentium и



мультимедийными (презентационными) системами, с подключением к общекорпоративной компьютерной сети ДВФУ и сети Интернет.

## X. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

### Описание показателей и критериев оценивания:

Оценка	Неудовлетворительно	Удовлетворительно	Хорошо	Отлично
Набранная сумма баллов (% выполненных заданий) (макс – 5)	Менее 3 (Менее 60%)	3-3,5 (61-74%)	3,6 -4,4 (75-84%)	4,5-5 (85-100%)
Оценка	Незачет	Зачет		
Набранная сумма баллов (% выполненных заданий) (макс – 5)	Менее 3 (Менее 60%)	3,1 – 5 (61-100%)		

### Критерии оценивания

В течение семестра студентам последовательно выдается набор из 5-х лабораторных работ, каждая из которых имеет вес 14%. Посещаемость занятий также учитывается и имеет вес 10%. Также проводится экзамен, вес которого составляет 30% от общего рейтинга (результат экзамена оценивается по трехбалльной шкале). По результатам всех контрольных мероприятий необходимо набрать рейтинг не менее 65% для получения оценки *удовлетворительно*, не менее 75% для получения оценки *хорошо*, и не менее 85% для получения оценки *отлично*.

### Лабораторные работы

#### 1. Работа с формами. Программирование на Java Script.

*Вариант 1.* Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число. - =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 4-х рядах (во всю ширину).  
Параметры калькулятора:

Ширина кальк.	50% окна браузера
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий
Шрифт в текстовом поле	18pt, bold, серый
Кнопки	Цифры, арифм. операции, SQRT, C, ON/OFF
Цвет панели/кнопок	Голубой/Linen

*Вариант 2.* Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметическое выражение (напр.:  $2+(3/4-7.5)*3$ ) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 4-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	50% окна браузера
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий
Шрифт в текстовом поле	18pt, bold, серый
Кнопки	Цифры, арифм. операции, SQRT, C, ON/OFF
Цвет панели/кнопок	Голубой/Linen

*Вариант 3.* Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину).

Параметры калькулятора:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, серый
Кнопки	Цифры, арифм. операции, C, AC, M+, M-, M ON/OFF
Цвет панели	Желтый/голубой

*Вариант 4.* Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. :  $2+(3/4-7.5)*3$ ) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 4-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, серый
Кнопки	Цифры, арифм. операции, C, AC, M+, M-, M, ON/OFF
Цвет панели/кнопок	Голубой/ ....

*Вариант 5.* Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину).

Параметры калькулятора:

Ширина кальк.	500px
Высота текст. поля/кнопок	30px/25px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, серый
кнопки	Цифры, арифм. операции, C, ON/OFF, F0, F2, F4, где Fn – округление (ост. n цифр после запятой)

*Вариант 6.* Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. :  $2+(3/4-7.5)*3$ ) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	500px
Высота текст. поля/кнопок	30px/25px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, серый
кнопки	Цифры, арифм. операции, C, ON/OFF, F0, F2, где Fn –

	округление (ост. п цифр после запятой)
--	--

*Вариант 7.* Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =, функция-аргумент- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	50%
Высота текст. поля/кнопок	30px/25px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, красный
кнопки	Цифры, арифм. операции, C, SIN, COS, TG, CTG, PI (конст.), ON/OFF

*Вариант 8.* Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. :  $SIN(2)+(3/4-7.5)*3$ ) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	50%
Высота текст. поля/кнопок	30px/25px
Шрифт на кнопках	18pt, bold, черный (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, синий
кнопки	Цифры, арифм. операции, C, SIN, COS, TG, CTG, PI (конст.), ON/OFF

*Вариант 9.* Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =, функция-аргумент- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Требования:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, черный
кнопки	Цифры, арифм. операции, C, SIN, COS, TG, CTG, PI (конст.), ON/OFF

*Вариант 10.* Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. :  $SIN(2)+(3/4-7.5)*3$ ) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Требования:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, черный (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, синий
кнопки	Цифры, арифм. операции, C, SQRT, ^(степень), LOG2, LOG10, LN, ON/OFF

*Вариант 11.* Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =, функция-аргумент- =). Вид: верхн.

строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Требования:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, черный
кнопки	Цифры, арифм. операции, C, SIN, COS, TG, CTG, PI (конст.), ON/OFF
Радиокнопки (в отд. строке)	F0, F2 (Fn – округление (ост. n цифр после запятой)), F – с обычной точностью

*Вариант 12.* Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. :  $\text{SIN}(2)+(3/4-7.5)*3$ ) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Требования:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, черный (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, синий
кнопки	Цифры, арифм. операции, C, SQRT, ^(степень), LOG2, LOG10, LN, ON/OFF
Радиокнопки (в отд. строке)	F0, F2 (Fn – округление (ост. n цифр после запятой)), F – с обычной точностью

## 2. Программирование на Java Script. Регулярные выражения.

### Проверка корректности ввода текстовых данных

Создать web-страницу, содержащую текстовые поля для ввода персональных данных с ограничениями согласно таблице:

№	Текстовое поле	Ограничения
1.	Фамилия	Буквы русского алфавита. Первая - заглавная
2.	Имя	
3.	Отчество	
4.	Число, месяц и год рождения	Шаблон: dd-mm-yyuu, например 18-10-1986. Ограничения на вводимые числа
5.	e-mail	Латинский шрифт, цифры. Первый символ – буква. Разделители: @ и .
6.	Логин	Первый символ – буква. Не менее 6 символов
7.	Пароль	Не может содержать одни цифры.

Разбивку страницы и выравнивание текстовых полей осуществить с помощью таблицы. Разместить на странице кнопку, по нажатию на которую будет происходить проверка введенных данных и вывод соответствующего сообщения об их корректности.

### Приложение

1. Некоторые спец. символы, используемые в регулярных выражениях:

<code>^</code>	начало строки
<code>\$</code>	конец строки
<code>.</code>	любой символ
<code>x   y</code>	один из символов <i>x</i> или <i>y</i>
<code>{n}</code>	<i>n</i> повторений предыдущего символа
<code>{n,}</code>	<i>n</i> или более повторений предыдущего символа
<code>{n,m}</code>	от <i>n</i> до <i>m</i> повторений предыдущего символа
<code>+</code>	предыдущий символ встречается 1 или более раз
<code>[xyz]</code>	один из перечисленных символов
<code>^xyz</code>	любой символ кроме перечисленных
<code>\d</code>	цифровой символ (эквивалентно <code>[0-9]</code> )
<code>\D</code>	нецифровой символ (эквивалентно <code>[^0-9]</code> )
<code>\w</code>	любой буквенно-цифровой символ или знак <code>_</code>
<code>\W</code>	( эквивалентно <code>[A-z0-9_]</code> ) любой не буквенно-цифровой символ или знак <code>_</code> ( эквивалентно <code>^[A-z0-9_]</code> )

2. Метод `exec(строка)` осуществляет поиск регулярного выражения в заданной строке. Возвращает подстроку, соответствующую первому совпадению.

3. Некоторые свойства объекта `RegExp` :

`input` – возвращает строку, в которой был осуществлен результативный поиск

`index` – возвращает позицию первого успешного поиска (если нет совпадения, то -1)

4. Примеры регулярных выражений:

`\d+ /` - одна или более цифр

`/[A-z]{3,}/` - последовательность из не менее трех латинских букв

`/.{4}$/` - любая строка из 4-х символов

`/^\D+$/` - строка из одного или более нецифровых символов

5. Пример странички, осуществляющей поиск регулярного выражения в задаваемой строке:

```
<html>
```

```
<script>
```

```
function fun(str,regex)
```

```
{
```

```
var re = eval(regex);
```

```
var a = re.exec(str);
```

```

alert(' Строка поиска: ' + RegExp.input + '\n Совпадение: ' + a + '\n Позиция
первого совпадения: ' +
RegExp.index);
}
</script>
<body>
<p align="center">
<font face="Verdana,Arial" size="2">
Поиск регулярного выражения в заданной строке
</p>
<form name="fname">
Строка: <br>
<input type="text" name="str" size="30">
<br>
<br>
Регулярное выражение:<br>
<input type="text" name="regexp" size="30">
<br>
<br>
<input type="button" value="Поиск" onclick="fun(document.fname.str.value,
document.fname.regexp.value);">
</form>
</font>
</body>
</html>

```

### 3. Программирование на Java Script. Использование таймера.

Создать web-страницу, реализующую электронные часы в формате ЧЧ : ММ : СС. Вывод текущего времени осуществить в текстовое поле <input>. Используя таблицы стилей CSS, задать размер текстового поля, фон, параметры шрифта (размер, цвет, ...). Включить в функцию электронных часов будильник. Время для сигнала будильника задавать в двух текстовых полях (часы, минуты). Сигнал будильника осуществить в виде периодической смены цвета фона текстового поля часов. Разбивку страницы и выравнивание текстовых полей осуществить с помощью таблицы.

#### *Приложение*

1. Некоторые методы объекта Date:

getHours(), getMinutes(), getSeconds()

– возвращает час суток (0-23), минуты(0-59), секунды(0-59)

2. Создание таймера осуществляется методом `setInterval()`, который выполняет код через указанный временной интервал. Синтаксис метода: `setInterval(выполняемый код, временной интервал в миллисекундах)`.

По окончании работы, используйте метод `clearInterval()` для прекращения работы таймера и удаления его из памяти.

3. Пример реализации простейших часов:

```
<html>
<script>
var tmr;
function timer(){
tmr=setInterval("clock()",1000);
}
function clock(){
var d = new Date();
document.formname.textfield.value           =
d.getHours()+':' +d.getMinutes()+':' +d.getSeconds();
}
function stop(){
clearInterval(tmr);
}
</script>
<body onload="timer();" onbeforeunload="stop();">
<form name="formname">
<input type="text" name="textfield" style="font-size: 14pt; font-weight: 700;
width: 100">
</form>
</body>
</html>
```

#### 4. Программирование на Java Script. Использование Cookies.

Создать скрипт, который осуществляет сбор и вывод статистики посещений страницы конкретным клиентом. Идентификация компьютера клиента и хранение данных по статистике посещений осуществить с помощью Cookies. Выводимая статистика должна включать следующую информацию о посещениях:

<i>Номера варианта</i>	<i>Статистика</i>
<i>1+3n</i>	Посещения за последнюю неделю по дням

$2+3n$	Посещения за последний месяц, распределенные по времени суток (посещения в ночное, дневное и вечернее время)
$3+3n$	Количество посещений за последний месяц в выходные и рабочие дни

Предусмотреть в скрипте проверку разрешено ли использование Cookies на компьютере клиента (свойство cookieEnabled объекта navigator).

#### Приложение

1. Установка cookies осуществляется через свойство cookie объекта document: `document.cookie = "<имя cookie>=<значение cookie>; expires=<время действия cookie>"`.

Время действия cookie задается в миллисекундах в GMT формате.

2. При создании нескольких cookie (например, `cookie1`, `cookie2`, ...) значение `document.cookie` представляет собой следующую строку:

" `cookie1=<значение cookie1>; cookie2=<значение cookie2>; ....`"

3. Считывание значения cookie можно осуществлять через свойство `length` и метод `substring( )` объекта `String`.

4. Для установки времени действия cookie требуется использовать методы объекта `Date`: `getTime()` – получение текущего времени (количество миллисекунд с 00:00 1.01.1970); `toGMTString()` – перевод времени в GMT-формат строкового типа.

5. Пример простейшего счетчика.

```
<html>
```

```
<script>
```

```
function f1(){
```

```
if(document.cookie){
```

```
var c = document.cookie;
```

```
var n = c.length;
```

```
counter = c.substring(8,n-1);
```

```
counter++;
```

```
var d = new Date();
```

```
d.setTime(d.getTime() + 10000000);
```

```
var exp = d.toGMTString();
```

```
document.cookie="visits="+counter+"; visits1="+counter+"; expires="+exp;
```

```
alert(counter);
```

```
}
```

```
else{
```

```
var d = new Date();
```

```
d.setTime(d.getTime() + 10000000);
```

```
var exp = d.toGMTString();
```



```
document.cookie="visits=1; expires="+exp;
alert(1);
}
}
</script>
<body onload="f1()">
</body>
</html>
```

## 5. Программирование на Java Script. Перемещение слоев.

Создать страницу, содержащую несколько графических элементов (например, елка и игрушки, либо продуктовая корзинка и продукты и др.). Все графические элементы необходимо спозиционировать через стили. Написать скрипт, позволяющий перемещать графические элементы с помощью мыши. При перемещении какого-либо слоя он должен располагаться поверх остальных слоев. Предусмотреть наличие *главного* слоя, при перемещении которого все слои расположенные на нем перемещаются вместе с ним (например, при перемещении елки, игрушки перемещаются вместе с елкой).

### Приложение

Пример скрипта реализующего перемещение слоев:

```
<html>
<head>
<title> Перемещение слоев</title>
</head>
<script language=javascript>
var draggedEl=null;
var shiftX, shiftY;
function moveMouse(){
if((event.button == 1) && (draggedEl != null)){
draggedEl.style.pixelTop = event.clientY - shiftY;
draggedEl.style.pixelLeft = event.clientX - shiftX;
}
}
function isDragEnabled(el){
if(el != null){
if(el.getAttribute("dragEnabled") != null) return el;
}
return null;
}
function pressMouseDown(){
```

```

var el = isDragEnabled(event.srcElement);
if(el != null){
draggedEl = el;
shiftX = event.offsetX;
shiftY = event.offsetY;
}
}
</script>
<body onmousedown = "pressMouseDown();"
onmousemove = "moveMouse();"
onmouseup = "draggedEl = null;" >
<h1 style="position: absolute; left: 100; top: 100;" dragEnabled>
Привет участникам конференции!
</h1>
<h1 style="position: absolute; left: 100; top: 300;" dragEnabled>
Hello!!!
</h1>
</body>
</html>

```

## **6. Программирование на PHP. Файловый ввод/вывод. Счетчик обращения к странице.**

Написать сценарий, реализующий счетчик обращения к странице. Требования: традиционный дизайн (либо путем включения графических элементов, либо путем использования таблицы стилей), наличие количества посещений за текущий день и общего количества посещений. Хранение всей информации осуществлять в текстовом файле. Считывание и запись данных в файл осуществить с помощью функций `fwrite`, `fread`, либо их аналогов.

### *Приложение*

#### 1. Проверка существования файла:

`file_exists($filename)` → TRUE/FALSE

#### 2. Открытие и закрытие файлов:

`fopen($filename, режим)` → файловый манипулятор (file handle), целое

`fclose(манипулятор)`

режимы:

r только чтение. Указатель текущей позиции в начало.

r+ чтение и запись.....

w запись. Указатель в начало. Содержимое уничтожается. w+ чтение и запись.....

a только запись. Указатель в конец Если файла нет – попытка создать.....

a+ чтение и запись.....

3. Запись в файл:

fwrite(манипулятор, строка [, длина в символах])

4. Чтение из файла:

fread(манипулятор, длина в байтах) читает из файла заданное количество байт

fgets(манипулятор, длина в байтах)

5. Перевод файла в массив:

file ( \$filename) → массив (элементы – построчно)

6. Пример – Счетчик обращения к странице

<?

```
$cfile = "test1.txt";
```

```
$afile = file($cfile); // Записываем содержимое файла в массив
```

```
$n_vis = $afile[0]; // Извлекаем первый элемент
```

```
++$n_vis;
```

```
$fc = fopen($cfile, "w"); // Открываем файл, текущая позиция в начале
```

```
fwrite($fc, $n_vis); // Записываем новое значение счетчика
```

```
fclose($fc);
```

```
print $n_vis;
```

```
?>
```

## 7. Файловый ввод/вывод. Работа с формами. Строковые функции.

Доработайте предлагаемый ниже сценарий, реализующий гостевую книгу. Чтение из файла осуществить с помощью функции fgets или fread. Сообщения должны быть отсортированы по времени отправки: более новые располагаются выше старых. При отображении сообщений нужно организовать автоматические разрывы, так чтобы длина строк не превосходила 50 символов. Сценарий должен реагировать на незаполненные текстовую область и поле.

```
<html>
```

```
<head><title>Гостевая книга</title></head>
```

```
<body>
```

```
Сообщение:<br>
```

```
<form action="guestbook.php" method="post">
```

```
<textarea name="txt" cols="40" rows="8">
```

```
</textarea>
```

```
<br><br>
```

```
Имя:<br>
```

```
<input type="text" name="login">
```

```
<br>
```

```

<input type="submit" name="a" value="Отослать">
<br><br>
</form>
<?
if(isset($a)){
$fm = fopen("guestbook.txt","a");
$str = "<i>".date("H:i:s, d F Y")."\n$login</i>\n$txt<br>\n";
fwrite($fm,$str);
fclose($fm);
}
$m = file("guestbook.txt");
$i=0;
while(isset($m[$i])){
print $m[$i]."<br>";
$i++;
}
?>
</body>
</html>

```

## 8. Программирование на PHP. Работа с таблицами базы данных. Регулярные выражения.

Переделайте гостевую книгу из лаб. работы 7. Хранение всех данных должно осуществляться в таблице базы данных. Добавить режим регистрации посетителей. Указываемые при регистрации данные: логин, пароль, фамилия, имя, e-мэйл, дата рождения. Сценарий должен посредством регулярных выражений осуществлять проверку корректности ввода персональной информации по каждой позиции (требования корректности – на усмотрение разработчика, исходя из здравого смысла).

### *Приложение*

#### 1. Работа с регулярными выражениями

Примеры шаблонов:

/ph+/ совпадение шаблона: phhhh, php4, phh

/ph{2,4}/ совпадение шаблона: phh, phhhhp4

Метасимволы:

\d – любая цифра. Пример: /(\d+)000\b/ - цифры, заканчив. 000.

\w – любой алф-цифровой символ

\W – любой символ не являющ., алф-цифр. Пример: /(\W+)/

\b – граница слова. Пример: /sa\b/ - строки, заканчив. на 'sa'.

\B – не граница. Пример: /sa\B/ - строки, не заканчив. на 'sa'.

Функции:

`preg_match` (шаблон, строка [ , массив совпадений ]) – совпадение шаблона в строке

`preg_grep`( шаблон, массив) → массив

- перебирает все элементы и возвращает те, где совпадает шаблон.

*Пример.* Поиск шаблона в строке.

```
<html>
```

```
<head><title>Поиск шаблона </title></head>
```

```
<body>
```

```
Строка:<br>
```

```
<form action="regex.php" method="post">
```

```
<input type="text" name="str" value=<?if(isset($a)) print $str;?>>
```

```
<br><br>
```

```
Регулярное выражение:<br><br>
```

```
<input type="text" name="re" value=<?if(isset($a)) print $re;?>>
```

```
<br><br>
```

```
<input type="submit" name="a" value="Отослать"><br><br>
```

```
</form>
```

```
<?
```

```
if(isset($a)){
```

```
if(preg_match($re,$str)) print "Совпало!";
```

```
else print "No";
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```

## 2. Функции PHP для работы с MySQL

### 2.1. Подключение к серверу MySQL

[идентификатор соединения = ]

```
@mysql_connect (хост, имя пользов, пароль) or die (“соединение не может быть уст-но”);
```

### 2.2. Выбор базы данных

```
[идентификатор базы = ] @mysql_select_db (имя бд) or die (“соединение не может быть уст-но”);
```

### 2.3. Завершение работы с MySQL

```
@mysql_close (идентификатор соединения);
```

### 2.4. Организация запросов к БД

```
[идентификатор результата = ] @mysql_query (запрос [, идентиф соедин]);
```

### 2.5. Получение набора данных после запроса `mysql_query`

mysql\_result ( идентификатор рез-та, номер записи [, поле] );

## 2.6. Количество записей, возвращаемых командой SELECT

mysql\_num\_rows (идентификатор рез-та);

*Пример.* Дана таблица products с полями: id, name, price. Требуется организовать выборку всех записей и напечатать их в виде таблицы.

```
$x=0 ;
```

```
$query = " SELECT * FROM products" ;
```

```
$result = mysql_query ($query) ;
```

```
while ( $x < mysql_num_rows ($result)) :
```

```
    $id = mysql_result ( $result, $x, 'id' ) ;
```

```
    $name = mysql_result ( $result, $x, 'name' ) ;
```

```
    $price = mysql_result ( $result, $x, 'price' ) ;
```

```
    print " <tr> \n" ;
```

```
    print " <td> $id </td> \n <td> $name </td> \n <td> $price </td> \n" ;
```

```
    print " </tr> \n" ;
```

```
    $x++ ;
```

```
endwhile;
```

## 2.7. Передача набора данных после запроса mysql\_query в массив

mysql\_fetch\_row (идентификатор рез-та) ;    либо

mysql\_fetch\_array (идентификатор рез-та) ;

*Пример.* То же, что и в примере 2.1, но с использованием функции mysql\_fetch\_row .

```
...
```

```
while ($row = mysql_fetch_row ($result)) :
```

```
    print " <tr> \n" ;
```

```
    print " <td>". $row[0]. " </td> \n <td>". $row[2]. " </td> \n <td>". $row[2]. " </td> \n" ;
```

```
    print " </tr> \n" ;
```

```
endwhile;
```

## 3. Некоторые команды SQL

### 3.1. Создание БД, таблиц

```
CREATE DATABASE [ IF NOT EXISTS ] database
```

```
CREATE TABLE [ IF NOT EXISTS ] table (col1 CHAR(20), col2 INTEGER, col3 CHAR(20))
```

### 3.2. Выбор записей из таблицы

```
SELECT * FROM table - выбор всех записей
```

```
SELECT * FROM table WHERE col3 = 'Майор'
```

```
SELECT * FROM table WHERE col1 = 'И%' - выбор записей с фамилией на 'И'
```

SELECT col1, col2 FROM table WHERE col3 = 'Майор'

SELECT \* FROM table Limit 10 - выбор первых 10 результирующих записей

### 3.3. Изменение записей

UPDATE table SET col3 = 'Капитан' WHERE col3 = 'Майор' - понижение звания всем майорам

### 3.4. Вставка записей

INSERT INTO table (col1, col2, col3) VALUES ( 'Пупкин К.К.', 1977, 'Сержант')

### 3.5. Уничтожение записей

DELETE FROM table LIMIT 5 - уничтожение первых 5 записей.

## **9. Программирование на PHP. Работа с таблицами базы данных. Cookie и сеансовые переменные.**

Создать гостевую книгу, включающую в себя:

1. Наличие регистрации и идентификации пользователя.
2. Возможность изменять личные установки (e-mail, адрес, пол ;) и др.)
3. Проверка корректности ввода e-mail, даты рождения, Ф.И.О. посредством использования регулярных выражений.
4. Счетчик посещения страницы, не увеличивающий число посещений идущих с одного ip-адреса, либо с одного компьютера ранее чем за 3 часа.

Требования:

1. Хранение всех данных осуществляется в таблицах базы данных.
2. Предусмотреть использование сеансовых переменных и cookie.

## **10. Программирование на PHP. Обработка текстовых данных. Работа с формами.**

**10.1.** Написать программу, которая подсчитывает процентное содержание слов различной длины в данном текстовом файле. Результат работы программы представить в виде таблицы и диаграммы.

*Примечание.*

Словом, является слитная последовательность букв. Знаки препинания частью слова не являются, даже, если они написаны с ним слитно.

**10.2.** Разработать текстовый редактор с возможностями: отображение содержимого некоторого текстового файла в текстовой области; редактирование и сохранение, поиск заданной символьной последовательности, поиск и замена.

### **Вопросы к экзамену**

1. Основы HTML, CSS;
2. Базовые структуры языка JavaScript;
3. Библиотека JQuery. Где и зачем применяется;

4. Технология Ajax. Где и зачем применяется;
5. Базовые структуры языка PHP;
6. Особенности строк в PHP;
7. Принцип работы сессий в PHP;
8. Принцип работы POST и GET запросов;
9. Основные принципы MVC моделей;
- 10.СУБД. Реляционные базы данных.