

Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»
Дата и время: 2024-02-21 00:00:00
471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кемеровский государственный университет»
Новокузнецкий институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования
«Кемеровский государственный университет»
(Наименование филиала, где реализуется данная дисциплина)

Факультет информатики, математики и экономики
Кафедра информатики и общетехнических дисциплин

УТВЕРЖДАЮ

Декан ФИМЭ

А.В. Фомина

« 13 » февраля 2020 г.

Рабочая программа дисциплины
Б1.В.ДВ.08.02 Языки программирования

Направление подготовки
44.03.05 Педагогическое образование (с двумя профилями подготовки)

Направленность (профиль) подготовки
Информатика и Физика

Программа *академического бакалавриата*

Квалификация выпускника
бакалавр

Форма обучения
Очная

Год набора 2016

Новокузнецк 2020

Лист внесения изменений
в РПД Б1.В.ДВ.08.02 Языки программирования

Сведения об утверждении:

утверждена Ученым советом факультета информатики, математики и экономики
(протокол Ученого совета факультета № 9 от 14.02.2019)

для ОПОП 2016 год набора _____ на 2019 / 2020 учебный год
по направлению подготовки 44.03.05 Педагогическое образование
(код и название направления подготовки / специальности)

направленность (профиль) подготовки Информатика и Физика

Одобрена на заседании методической комиссии факультета
протокол методической комиссии факультета № 6 от 14.02.2019)

Одобрена на заседании обеспечивающей кафедры ИОТД
протокол № 5 от 19.01.2019г. Можаров М.С / _____
(Ф. И.О. зав. кафедрой) (Подпись)

Переутверждение на учебный год:

на 2020 / 2021 учебный год

утверждена Ученым советом факультета _____
(протокол Ученого совета факультета № 8 от 13.02.20 г.

Одобрена на заседании методической комиссии факультета _____
протокол методической комиссии факультета № 6 от 06.02.2020г.

Одобрена на заседании обеспечивающей кафедры _____
протокол № 5 от 19.12.2019 г. Можаров М.С / _____
(Ф. И.О. зав. кафедрой) (Подпись)

на 20 ____ / 20 ____ учебный год

утверждена Ученым советом факультета _____
(протокол Ученого совета факультета № ____ от ____ . ____ .201 ____ г.

Одобрена на заседании методической комиссии факультета _____
протокол методической комиссии факультета № ____ от ____ . ____ .20 ____ г.

Одобрена на заседании обеспечивающей кафедры _____
протокол № ____ от ____ . ____ .20 ____ г. _____ / _____
(Ф. И.О. зав. кафедрой) (Подпись)

на 20 ____ / 20 ____ учебный год

утверждена Ученым советом факультета _____
(протокол Ученого совета факультета № ____ от ____ . ____ .201 ____ г.

Одобрена на заседании методической комиссии факультета _____
протокол методической комиссии факультета № ____ от ____ . ____ .20 ____ г.

Одобрена на заседании обеспечивающей кафедры _____
протокол № ____ от ____ . ____ .20 ____ г. _____ / _____
(Ф. И.О. зав. кафедрой) (Подпись)

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной образовательной программы «Педагогическое образование» по программе бакалавриата профиля "Информатика и физика"	4
2. Место дисциплины в структуре ООП бакалавриата.....	5
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся.....	5
3.1. Объем дисциплины (модуля) по видам учебных занятий (в часах)	5
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий	6
4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)	6
4.2 Содержание дисциплины (модуля), структурированное по темам (разделам)	6
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)	8
6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)	9
6.1 Типовые контрольные задания или иные материалы	9
6.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций	22
7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)	25
а) основная учебная литература:	25
б) дополнительная учебная литература:	25
8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля).....	25
9. Методические указания для обучающихся по освоению дисциплины (модуля)	25
10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)	26

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной образовательной программы «Педагогическое образование» по программе бакалавриата профиля "Информатика и физика"

В результате освоения ООП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

<i>Коды компетенции</i>	Результаты освоения ООП <i>Содержание компетенций*</i>	Перечень планируемых результатов обучения по дисциплине
СПК-1	готов к применению знаний теоретической информатики, фундаментальной и прикладной математики для анализа и синтеза информационных систем и процессов, а также для решения прикладных задач получения, хранения, обработки и передачи информации	Знать: • общие проблемы и задачи теоретической информатики, основные принципы и этапы информационных процессов, наиболее широко используемые классы информационных моделей; • основные математические методы получения, хранения, обработки, передачи и использования информации; • состояние и перспективы развития информационных и инфокоммуникационных технологий, рынок программно-аппаратных средств; • основы архитектуры, устройства и функционирования вычислительных систем и компьютерных сетей; • регламенты обеспечения информационной безопасности, методы и средства защиты информации, типовые уязвимости, учитываемые при эксплуатации устанавливаемого программного обеспечения; Уметь: • применять математический аппарат анализа и синтеза информационных систем; • устанавливать, настраивать, обновлять системное и прикладное программное обеспечение на конечных устройствах пользователей и/или серверном оборудовании, осуществлять лицензионную регистрацию; • настраивать программное обеспечение в соответствии с регламентами обеспечения информационной безопасности, использовать программно-аппаратные и программные средства защиты информации; Владеть: • современными

		формализованными математическими, информационно-логическими и логико-семантическими моделями и методами представления, сбора и обработки информации
--	--	---

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина « Языки программирования» входит в вариативную часть профессионального цикла программы бакалавра с кодом (Б.З.В), дисциплина по выбору.

Для освоения данной дисциплины студенты используют знания, умения, навыки, сформированные в процессе изучения дисциплин «Операционные системы», «Информационные технологии», «Программное обеспечение».

Освоение данной дисциплины является необходимой основой для последующего изучения дисциплин вариативной части профессионального цикла, прохождения практики.

Дисциплина (модуль) изучается на 1,2_ курсе в 2,3 семестрах.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины (модуля) составляет 8 зачетных единиц (ЗЕТ), 288 академических часов.

3.1. Объём дисциплины (модуля) по видам учебных занятий (в часах)

Объём дисциплины	Всего часов
	для очной формы обучения
Общая трудоемкость дисциплины	288
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	
Аудиторная работа (всего):	108
в т. числе:	
Лекции	36
Семинары, практические занятия	
Практикумы	
Лабораторные работы	72
в т.ч. в активной и интерактивной формах	26
Внеаудиторная работа (всего):	
В том числе, индивидуальная работа обучающихся с преподавателем:	
Курсовое проектирование	
Групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем	
Творческая работа (эссе)	

Объём дисциплины	Всего часов
	для очной формы обучения
Самостоятельная работа обучающихся (всего)	108
Вид промежуточной аттестации обучающегося (зачет / экзамен)	экзамен 72

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

для очной формы обучения

№ п/п	Раздел дисциплины	Общая трудоёмкость (часах)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах)			Формы текущего контроля успеваемости
			аудиторные учебные занятия		самостоятел ьная работа обучающихс я	
		всего	лекции	семинары, практические занятия		
1.	Web- программирование	44	18	6	20	Опрос, защита лабораторной работы
2.	Ориентированный язык программирования JavaScript	78	8	30	40	Опрос, защита лабораторной работы
3.	Язык программирования PHP	94	10	36	48	Опрос, защита лабораторной работы

4.2 Содержание дисциплины (модуля), структурированное по темам (разделам)

№ п/п	Наименование раздела дисциплины	Содержание
1. Web-программирование		
<i>Содержание лекционного курса</i>		
1.1	Языки веб-программирования	

№ п/п	Наименование раздела дисциплины	Содержание
1.2	Язык разметки HTML	
1.3	Интерактивные формы HTML	
1.4	Каскадные таблицы стилей CSS	
1.5	Клиент-серверное взаимодействие	
1.6	Ориентированный язык программирования JavaScript	
1.7	Язык программирования PHP	
Темы лабораторных занятий		
1.1	Структура веб-документа	
1.2	Веб-формы	
2. Ориентированный язык программирования JavaScript		
<i>Содержание лекционного курса</i>		
2.1	Основные конструкции JavaScript	
2.2	Типы данных: Массивы, Объекты.	
2.3	Основные операторы. Операторы сравнения и логические сравнения.	
2.4	Работа с формами в JavaScript	
<i>Темы лабораторных занятий</i>		
2.1	Основные конструкции JavaScript. Свойства и методы объекта Window.	
2.2	Типы данных. Методы.	
2.3	Массивы.	
2.4	Объекты (null, Images).	
2.5	Операторы.	
2.6	Инструкции. Простые и составные инструкции.	
2.7	Цикл while с предусловием, Цикл while с постусловием Цикл for.	
2.8	Условная инструкция if.	
2.9	Элементы форм. Общее представление.	
2.10	Функции.	
2.11	Области видимости переменных.	
2.12	Элементы форм. Общее представление.	
2.13	Атрибуты элементов форм	
2.14	JavaScript и HTML	
2.15	Работа с формами в JavaScript	
2.16	Включение файлов со сценариями JavaScript	
2.17	Определение обработчика события	
2.18	Обработчики событий и методы элементов форм	
2.19	Определение возможностей клиентского JavaScript	
2.20	Объектная модель документа	
2.21	Реализация абстракций данных методами объектно-ориентированного программирования.	
2.22	Структурированные типы языка программирования высокого уровня.	
2.23	Создание фильтров и их сохранение	
3. Язык программирования PHP		
<i>Содержание лекционного курса</i>		
3.1	Основной синтаксис PHP	
3.2	Условные операторы PHP	
3.3	Функции и массивы в PHP	
3.4	Работа с формами в PHP	
<i>Темы лабораторных занятий</i>		
3.1	Основные конструкции языка PHP	

№ п/п	Наименование раздела дисциплины	Содержание
3.2	Взаимодействие PHP и HTML	
3.3	Элементы языка	
3.4	Управление потоком if	
3.5	Управление потоком for	
3.6	Управление потоком while	
3.7	Функции, определяемые пользователем	
3.8	Функции внутри функции	
3.9	Работа со строками	
3.10	Работа с файлами	
3.11	Массивы	
3.12	Основы клиент-серверных технологий	
3.13	Работа с формами	

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

1) Видеолекции и презентации с интернет-ресурсов
<http://www.intuit.ru/department/se/standcertsoft/>
http://window.edu.ru/window_catalog/files/r23699/2019.pdf

2) А. Ю. Крупский, Л. А. Феоктистова. Разработка и стандартизация программных средств: Учебное пособие / — М.: Издательско-торговая корпорация «Дашков и К°», 2009. — 100 с.

Темы индивидуальных домашних заданий

1. Интерактивная обработка данных.
2. Глобальные сети компьютеров и Интернет. Мэйнфреймы.
3. Пакетный режим обработки данных
4. Персональный компьютер. Локальные сети компьютеров.
5. Узлы и каналы Интернета. Поставщики услуг Интернета.
6. Управление Интернетом. Стандарты Интернета.
7. TCP/IP. Название протокола TCP/IP. Адреса в сети TCP/IP.
8. Классы сетей. Запись адреса IP.
9. Маска подсети. Порты TCP/IP.
10. Маршрутизаторы в сет . Доменные имена узлов.
11. Утилиты для работы с TCP/ IP. Утилита PING.
12. Трассировка пакетов утилитой TRACERT
13. Утилиты для работы с TCP/ IP. Утилита NETSTAT.
14. Утилиты для работы с TCP/ IP Утилита ROUTE
15. Преимущества серверов Web. Адреса страниц Web.
16. Универсальный идентификатор ресурсов URL.
17. Ссылки на страницах Web. Протокол HTTP.
18. Статические и динамические страницы.
19. Серверы Web и базы данных.
20. Пассивные и активные серверы Web .
21. Активность на стороне клиента.
22. Интеграция серверов Web и SQL Server
23. Структура документа HTML. Заголовок документа . Тело документа
24. Вставка специальных символов и символьных объектов. Списки .
25. Ссылки в документах HTML
26. Графика, анимация, видео и звук . Выбор формата
27. Таблицы. Фреймы.

- 28 Отладка узла Web на локальном компьютере. Выбор способа размещения узла Web.
- 29 Настройка протокола TCP/IP.
- 30 Удаленная настройка и администрирование сервера.
- 31 Работа с протоколом FTP. Удаленное администрирование.
- 32 Администрирование средствами Terminal Services.
- 33 Регистрация доменного имени. Регистрация в поисковых системах Интернета.
- 34 Особенности JavaScript. Переменные в JavaScript.
- 35 Операторы языка JavaScript. Сценарии JavaScript в страницах HTML. Применение сценариев для создания интерфейса пользователя.
- 36 Модальные и немодальные диалоговые панели
- 37 Работа с фреймами. Взаимодействие фреймов друг с другом.
- 38 Основы ASP. Приложения ASP и сеансы.
- 39 Отладка сценариев в страницах ASP .
- 40 Работа с файлами. Использование объекта Dictionary

Методические указания по самостоятельной работе размещены по адресу:
<https://skado.dissw.ru/table>

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

6.1 Типовые контрольные задания или иные материалы

Лабораторная работа Условная инструкция if.

Вопросы

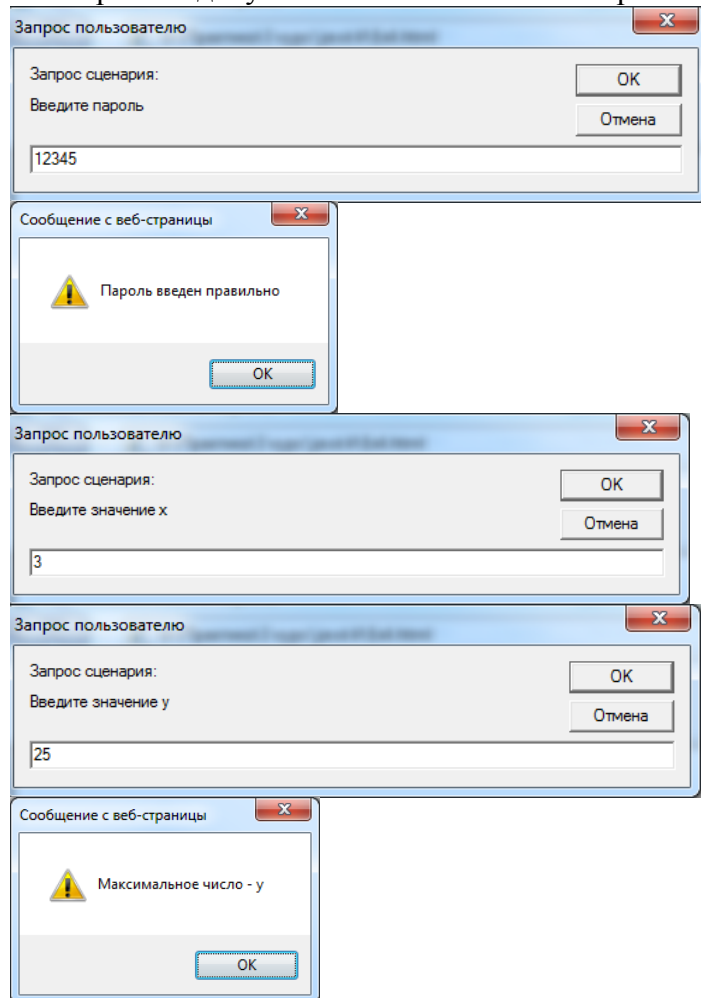
- Где в документе может располагаться тэг script по стандарту HTML?
- Какие конструкции для циклов есть в javascript?
- На что влияет описание DOCTYPE в начале документа?
- Что делает оператор = = ?
- Чувствителен ли JavaScript к регистру символов?

Задание 1:

- 1. Рассмотрите пример скрипта:

```
<html>
<head>
<title>if</title>
</head>
<body>
<script language="JavaScript" type="text/JavaScript">
var x, y;
x=parseInt(prompt("Введите значение x","")); // метод parseInt()
переводит строку в целое
y=parseInt(prompt("Введите значение y","")); // число
if(x<y)
{
alert("Максимальное число - y")
}
else {
alert("Максимальное число - x")
}
</script>
</body>
</html>
```

2. Допишите скрипт так, чтобы при введении пользователем одинаковых чисел, открывалось сообщение "Введенные числа равны!".
3. Напишите скрипт, в котором пользователя просят ввести правильный пароль. При вводе правильного пароля, в окне браузера появляется сообщение о том, что пароль верен. При вводе неправильного пароля – выпадает сообщение о неправильно введенном пароле. Для выполнения задания введите переменную password, в которую сохраните верное значение пароля.
4. Сохраните документ с именем Ex4.html в рабочей папке.



Вид скрипта

```
<html>
<head>
<title>Ex4</title>
</head>
<body>
<script language="JavaScript" type="text/JavaScript">
var x, y, pass;
pass=12345;
pass1=prompt("Введите пароль","");
if(pass==pass1){
alert("Пароль введен правильно");
}
else{
alert("Пароль введен неправильно!");
}
x=parseInt(prompt("Введите значение x",""));
```

```

y=parseInt(prompt("Введите значение y",""));
if(x<y)
{
alert("Максимальное число -y")
}
elseif(x=y) {
alert("Введенные числа равны!")
}
else{
alert("Максимальное число -x")
}
</script>
</body>
</html>

```

6.1.1. Экзамен / зачет

Тест

1. Выберите тег для создания нумерованного списка?

-
- <dl>
-
- <list>

2. Выберите верный способ вставки изображения?

- image.gif
-
-
- <image src="image.gif" alt="MyImage">

3. В HTML строчные элементы отображаются нормально без перехода на новую строку

- Истина
- Ложь

4. Выберите верный способ создания текстового поля для ввода информации

- <input type="textfield">
- <textinput type="text">
- <textfield>
- <input type="text">

5. Выберите верный способ создания многострочного текстового поля

- <input type="textbox">
- <textarea>
- <input type="textarea">

6. Выберите верный способ создания выпадающего списка

- <select>
- <list>
- <input type="dropdown">
- <input type="list">

7. Выберите верный способ установки фонового изображения страницы?

- `<body background="background.gif">`
- ``
- `<background img="background.gif">`

8. Выберите тег для создания нумерованного списка?

- `<dl>`
- `<list>`
- `<ol>`
- `<ul>`

9. Выберите теги для работы с таблицами?

- `<table><tr><td>`
- `<table><head><tfoot>`
- `<table><tr><tt>`
- `<thead><body><tr>`

10. Выберите верный способ создания чекбокса

- `<checkbox>`
- `<input type="checkbox">`
- `<input type="check">`
- `<check>`

11. Какие из этих вариантов задают массив из элементов "a", "b" ?

- `var a = new Array("a","b")`
- `var a = { "a", "b" }`
- `var a = ( "a", "b" )`
- `var a = [ "a", "b" ]`
- `var a = "a,b".split(',')`

12. Какие из этих тегов соответствуют стандарту HTML (strict) ?

- `<script>`
- `<script type="text/javascript">`
- `<script type="application/x-javascript">`
- `<script language="javascript">`
- `<script language="javascript" type="text/javascript">`

13. Можно ли инициировать DOM-событие из javascript? Например, смоделировать клик мышкой на элементе, чтобы javascript-код кликнул за пользователя.

- Да, можно
- Нет, нельзя
- В некоторых браузерах можно

14. Какой оператор из этих – выполняет не только математические операции?

- `*`
- `/`
- `+`
- `-`

- >>>

15. Сколько параметров можно передать функции ?

- Ровно столько, сколько указано в определении функции
- Сколько указано в определении функции или меньше
- Сколько указано в определении функции или больше
- Любое количество

16. Что из этого – не событие мыши ?

- onmousecroll
- onclick
- onmouseover
- onmousemove
- onmousewheel

17. В каком случае из перечисленных событие не попадет на обработку javascript?

- Если в момент его наступления обрабатывается другое событие
- Если страничка просматривается локально, т.е offline
- Только если javascript отключен

18. Какое событие не вызывается кликом мыши ?

- onfocus
- onclick
- onkeydown
- onmousedown

19. Что делает код:

- break me;
- Ломает интерпретатор javascript
- Выходит из текущего блока цикла или switch на метку “me”
- Выдает ошибку

В разных браузерах по-разному

20. Какие конструкции для циклов есть в javascript?

- Только две: for и while.
- Только одна: for
- Три: for, while и do...while.

21. Какие обработчики событий сработают при клике на div?

- div.onclick = function() { alert(1) }
- div.onclick = function() { alert(2) }
- Только первый
- Только второй
- Первый, затем второй

22. Верно ли следующее утверждение: Содержимое тэга script нужно заключать в комментарии <!-- .. -->, чтобы браузеры, которые не поддерживают javascript, работали корректно?

- Да

- Нет
23. Можно ли через javascript подключить внешний js-файл, отсутствующий на странице?
- Да, но только один раз
 - Да, но только до загрузки страницы
 - Да, сколько угодно файлов когда угодно
24. Где в документе может располагаться тэг script по стандарту HTML ?
- Только в HEAD
 - Только в BODY
 - В HEAD или в BODY
 - Где угодно, главное чтоб был
25. Какие из этих вызовов корректно бросят исключение ?
- throw “Ошибка”
 - throw new Error(“Ошибка”)
 - throw { message: “Ошибка” }
 - throw Error(“Ошибка”)
 - Ни один
26. Как правильно вывести приветствие через 5 секунд после запуска скрипта ?
- sleep(5); alert(“Привет!”);
 - sleep(5000); alert(“Привет!”);
 - setTimeout(‘alert(“Привет!”)’, 5000);
 - setTimeout(function() {alert(“Привет!”)}, 5000);
 - setTimeout(function() {alert(“Привет!”)}, 5);
27. Правда ли что a == b?
a = newArray(1,2,3)
b = newArray(1,2,3)
- Правда
 - Неправда
 - Зависит от множества случайных обстоятельств..
28. Верно ли следующее утверждение: Любое содержимое тэга script нужно обязательно заключать в специальную секцию <![CDATA[...]]>, чтобы документ соответствовал стандарту XHTML 1.0 ?
- Да
 - Нет
29. Что такое ECMAScript?
- Новый язык программирования
 - Переработанная реализация Javascript
 - Спецификация языка Javascript
30. Можно ли в скрипте перевести посетителя на другую страницу сайта?
- Да, куда угодно
 - Да, но только в рамках текущего сайта
 - Нет, нельзя

31. Что делает оператор ===?

- Сравнивает по ссылке, а не по значению
- Сравнивает без приведения типа
- Нет такого оператора

32. На что влияет описание DOCTYPE в начале документа?

- На внешний валидатор при проверке соответствия стандарту
- На отображение, работу CSS-свойств
- На наличие и отсутствие некоторых свойств в DOM
- На особенности работы XMLHttpRequest
- DOCTYPE – атавизм, он ни на что не влияет.

33. Чему равно i в конце кода ?

```
var i = 5  
{  
var i = 1  
}
```

- 1
- 3
- 5
- В коде ошибка

34. В каких браузерах будет работать этот код? element.style.setExpression(“width”, “100px”)

- Internet Explorer 5.0+
- Internet Explorer 6.0+
- Opera
- Firefox
- Safari

35. Какова длина a.length массива a:

```
var a = []  
a[1] = 5  
a[3] = 53  
delete a[3]
```

- 1
- 2
- 3
- 4
- Больше

36. Какая арифметическая операция приводит к ошибке в javascript?

- Деление на ноль
- Умножение числа на строку
- Корень из отрицательного числа
- Никакая из вышеперечисленных

37. Какова длина a.length массива a ?

```
var a = []  
a[1] = 5  
a[5] = 55
```

- 1
- 3
- 6
- Другое

38. Чему равно `Number.NEGATIVE_INFINITY`(“минус бесконечность”) + `Number.POSITIVE_INFINITY`(“плюс бесконечность”) ?

- Останется `Number.NEGATIVE_INFINITY`
- Получится `Number.NEGATIVE_INFINITY`
- Ноль
- Другое

39. Из родительского окна во фрейм `child` поставили переменную:

- `frames.child.users = ["Маша", "Паша", "Даша"]`
- `if (users instanceof Array)`
- `if (users instanceof parent.Array)`
- `if (users.constructor.name == "Array")`
- `if(typeof users == "array")`

40. Каких бинарных операторов НЕТ в javascript?

- *
- ^
- %
- #
- &
- >>
- >>>
- !

41. Является ли использование сокращённых PHP тегов хорошей практикой?

- Да
- Нет

42. Как писать многострочный комментарий?

- `##` это комментарий `##`
- `<*` это комментарий `*>`
- `/*` это комментарий `*/`

43. PHP код не может быть встроен в HTML. Это выражение...

- Ложь
- Правда

44. Какой из перечисленных типов данных не является обычным?

- Date
- String
- Integer
- Boolean

45. `$CustomVariable` и `$customvariable` это одна и та же переменная. Это выражение...

- Правда
 - Ложь
46. Каждое PHP выражение должно заканчиваться..
- Запятой (,)
 - Точкой с запятой (;)
 - Точкой (.)
 - Двоеточие (:)
47. Какая из этих переменных не является валидной?
- \$my_variable
 - \$myvariable
 - \$_myvariable
 - \$my-variable
48. Как определить константу?
- variable("FOO", "BAR");
 - constant("FOO", "BAR");
 - define("FOO", "BAR");
 - defineconstant("FOO", "BAR");
49. Выберите верный php тэг...
- <#php #>
 - ?..... ?
 - <%php %>
50. Выберите префикс, который необходим для создания переменных?
- @
 - Нет никаких специальных префиксов
 - \$
 - #
51. Является ли переменная \$221BakerStreet валидной?
- Да
 - Нет
52. Какой из следующих операторов используется для конкатенации строк?
- .
 - ==>
 - ->
 - &
53. Какой будет результат выполнения следующего кода:
- <?php
 - \$sport = 'soccer';
 - \$statement = 'He plays \$sport';
 - echo \$statement;
 - ?>
 - He plays \$sport

- He plays soccer

54. Название переменных в PHP чувствительны к регистру. Это выражение..

- Правда
- Ложь

55. У вас есть строка даты в формате уууу-мм-дд. Вы хотите получить год, месяц и число в отдельные переменные \$year, \$month, \$day. Какой функцией следует воспользоваться для решения данной задачи?

- scanf("2011-09-09", "%d-%d-%d", \$year, \$month, \$day);
- sscanf("2011-09-09", "%d-%d-%d", \$year, \$month, \$day);
- printf("2011-09-09", "%d-%d-%d", \$year, \$month, \$day);
- sprintf("2011-09-09", "%d-%d-%d", \$year, \$month, \$day);

56. Какой будет результат?

```
<?php
$array = array('a' => 'One', 'b' => 'Two', 'c' =>
array('d' => 'Three', 'e' => 'Four'));
function print_element($array)
{
extract($array);
return $c['e'];
}
print print_element ($array);
?>
```

- Three
- Four
- One
- Two

57. Какой будет результат?

```
<?php
$name = array("d" => "Mark", "a" => "David", "b" => "Peter", "c" => "Martha");
$nameArrayObject = new ArrayObject($name);
$nameArrayObject->ksort();
foreach ($nameArrayObject as $key => $val) {
    echo "$key = $val";
}
?>
```

- a = David
b = Peter
c = Martha
d = Mark
- a = Mark
b = David
c = Peter
d = Martha
- a = David
b = Mark

c = Martha

d = Peter

- Ошибка

58. Как будет выглядеть массив после выполнения следующего кода?

```
<?php
```

```
$array = array ('1', '2', '3');
```

```
foreach ($array as $key => $value)
```

```
{
```

```
    $value = 4;
```

```
}
```

```
print_r($array);
```

```
?>
```

- Array ([0] => 1 [1] => 2 [2] => 3)
- Array ([0] => 4 [1] => 8 [2] => 12)
- Array ([0] => 1 [4] => 2 [8] => 3)
- Ошибка

59. Какую функцию необходимо использовать для того чтобы увидеть на экране Hello World?

```
function myfunction()
```

```
{
```

```
// место для функции
```

```
print $string;
```

```
}
```

- myfunction("Hello World!");
- \$string = func_get_arg();
- list(\$string) = func_get_args();
- \$string = \$argv[1];
- \$string = get_function_args();

60. Какая из следующих функций преобразует все ключи массива к верхнему регистру?

- array_count_values()
- array_combine()
- array_change_key_case()
- array_chunk()

61. Какой будет результат?

```
<?php
```

```
function a(&$a = 19)
```

```
{
```

```
    $a .= 1;
```

```
}
```

```
$b = 6;
```

```
a($b);
```

```
echo $b++;
```

```
?>
```

- 61
- 7
- 6
- Ошибка

62. Какой будет результат?

```
<?php
$input = array(4, "4", "3", 4, 3, "3", 3, 3, 3, 3, 3, 5, 5, 5, 5, 7, 7, 7, 7);
echo count(array_unique($input));
?>
```

- 4
- 19
- 6
- Ошибка

63. Как будет выглядеть массив после выполнения следующего кода?

```
<?php
$queue = array(1, 2);
array_unshift($queue, 0, 4);
print_r($queue);
?>
```

- Array([0] => 4[1] => 2[2] => 1[3] => 0)
- Array([0] => 0[1] => 4[2] => 1[3] => 2)
- Array([0] => 0[1] => 1[2] => 2[3] => 4)
- Array([0] => 1[1] => 2)

64. У нас есть массив:

```
Array
(
    [0] => book1.pdf
    [1] => book11.pdf
    [2] => book12.pdf
    [3] => book2.pdf
)
```

Какую функцию нужно использовать для того чтобы отсортированный массив выглядел так:

```
Array
(
    [0] => book1.pdf
    [3] => book2.pdf
    [1] => book11.pdf
    [2] => book12.pdf
)
```

- sort()
- asort()
- ksort()
- natasort()

65. Какая ошибка в коде:

```
<?php
echo "Привет";
session_start();
?>
```

- Функции session_start() не существует, правильная функция – start_session().
- Нельзя выводить информацию до начала сессии.
- После начала сессии необходимо её использовать, а в коде она никак не используется.

- В этом коде нет ошибок.

66. Вы установили пакет Denwer и написали скрипт, который должен отправлять письма на электронный ящик mail.ru. Но письма почему-то не доходят. Почему?

- Что-то сломалось в настройках sendmail.
- Ошибка в коде.
- Denwer по умолчанию использует свой sendmail, который лишь эмулирует отправку, но в реальности ничего не отправляет.
- Mail.ru и другие серьезные почтовые серверы не принимают письма, отправленные с локального сервера.

67. Какая функция в PHP позволяет найти вхождение подстроки?

- str_repeat()
- strtok()
- strpos()
- substr()

68. Какая разница между равенством и эквивалентностью?

- Никакой разницы нет.
- Знак эквивалентности проверяет лишь значения операндов, а знак равенства значения и их типы.
- Знак эквивалентности работает лишь для строк, а знак равенства применим к любым типам.
- Знак равенства проверяет лишь значения операндов, а знак эквивалентности значения и их типы.

69. Как сделать редирект (например, на google.ru) на PHP?

- document.location = "http://google.ru";
- header("Location: http://google.ru");
- location.href = "http://google.ru";
- header("Redirect: http://google.ru");

70. Что вы увидите после запуска скрипта:

```
<?php
for ($i = 0; $i < 5; $i++) {
    if ($i % 2 == 0) continue;
    echo $i;
}
?>
```

- 13
- 024
- 013
- 24

71. Каков результат выполнения у данного скрипта:

```
<?php
$x = 5;
echo "Переменная $x = 5";
?>
```

- Переменная x = 5

- Переменная 5 = 5
- Переменная \$x = 5
- Ошибка, поскольку строки с переменными должны быть всегда в одинарных кавычках.

72. Есть строка: "my string". В чём заключается ошибка в коде?

```
<?php
```

```
if (strpos("mystring", "m") == false) echo "Символа m в строке mystring нет";
```

```
?>
```

- Функция strpos() вернула 0, который в PHP равен false. Чтобы не было ошибки надо вместо знака равенства (==) использовать знак эквивалентности (===).
- Нет фигурных скобок у блока операторов при срабатывании условия.
- Функция strpos() неправильно использована. Сначала должна идти искомая строка, а уже потом та строка, в которой происходит поиск.
- Здесь нет ошибок, так как strpos("mystring", "m") не равен false, и строки "Символа m в строке mystring нет" не появляется

73. Каков результат выполнения у данного скрипта:

```
<?php
```

```
$x = 5;
```

```
echo 'Переменная $x = 5';
```

```
?>
```

- Переменная x = 5
- Переменная 5 = 5
- Переменная \$x = 5
- Ошибка, поскольку строки с переменными должны быть всегда в двойных кавычках.

6.2.2 Наименование оценочного средства

Лабораторная работа

Структура лабораторной работы представлена следующим образом:

- устный опрос по теме данного лабораторного занятия;
- выполнение лабораторной работы (индивидуальное, в микрогруппах).

Таким образом, на лабораторной работе занятии студент гарантированно получает оценку за опрос и за выполнение лабораторной работы.

6.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций

Распределение баллов по темам и видам учебной работы за 1 семестр

№ п/п	Темы и виды учебной работы	min	max
1	Посещение лекционных занятий	1	9
2	Лабораторная работа «Структура веб-документа»	1	2
3	Лабораторная работа Веб-формы	1	2

4	Лабораторная работа Основные конструкции JavaScript. Свойства и методы объекта Window.	1	2
5	Лабораторная работа Типы данных. Методы.	1	2
6	Лабораторная работа Массивы.	1	2
7	Лабораторная работа Объекты (null, Images).	1	2
8	Лабораторная работа Операторы.	1	2
9	Лабораторная работа Инструкции. Простые и составные инструкции.	1	2
10	Лабораторная работа Цикл while с предусловием, Цикл while с постусловием Цикл for.	1	2
11	Лабораторная работа Условная инструкция if.	1	2
12	Лабораторная работа Элементы форм. Общее представление.	1	2
13	Лабораторная работа Функции.	1	2
14	Лабораторная работа Элементы форм. Общее представление.	1	2
15	Лабораторная работа Атрибуты элементов форм	1	2
16	Лабораторная работа JavaScript и HTML	1	2
17	Лабораторная работа Работа с формами в JavaScript	1	2
18	Лабораторная работа Включение файлов со сценариями JavaScript	1	2
19	Лабораторная работа Определение обработчика события	1	2
20	Лабораторная работа Обработчики событий и методы элементов форм	1	2
21	Лабораторная работа Определение возможностей клиентского JavaScript	1	2
	ИТОГО	10	51

Распределение баллов по темам и видам учебной работы за 2 семестр

№ п/п	Темы и виды учебной работы	min	max
1	Посещение лекционных занятий	1	9
2	Лабораторная работа Объектная модель документа	1	2
3	Лабораторная работа Реализация абстракций данных методами объектно-ориентированного программирования.	1	2
4	Лабораторная работа Структурированные типы языка программирования высокого уровня.	1	2
5	Лабораторная работа Создание фильтров и их сохранение	1	2
6	Лабораторная работа Основные конструкции языка PHP	1	2
7	Лабораторная работа Взаимодействие PHP и HTML	1	2
8	Лабораторная работа Элементы языка	1	2
9	Лабораторная работа Управление потоком if	1	2
10	Лабораторная работа Управление потоком for	1	2
11	Лабораторная работа Управление потоком while	1	2
12	Лабораторная работа Функции, определяемые пользователем	1	2
13	Лабораторная работа Функции внутри функции	1	3
14	Работа со строками	1	3
15	Лабораторная работа Работа с файлами	1	3
16	Лабораторная работа Массивы	1	3
17	Лабораторная работа Основы клиент-серверных технологий	1	3
18	Лабораторная работа Работа с формами	1	3
	ИТОГО	18	49

В конце изучения всех тем подводятся итоги работы студентов на лекционных и практических занятиях путем суммирования всех заработанных баллов.

Максимальное количество баллов, которое может заработать студент за время обучения, равно 100 баллов.

Это предполагает следующие виды заданий:

1) Посещение лекционных занятий оценивается по 1 баллу – максимальное количество баллов = 18;

2) За правильный ответ по теоретической части лабораторной испытуемый получает 1 балл.

Практическая часть лабораторной работы – 1 балл за каждую задачу, максимальное количество баллов на одной лабораторной работе - 2 или 3 балла. За все время обучения 82 балла.

Поскольку студент выполняет различные виды работ, получает за них не только максимальное, но и минимальное количество баллов, то получаемый результат (сумма) целиком зависит от его активности в течение семестра.

Получение оценки за экзамен по итогам текущей успеваемости –

Удовлетворительно – 50 – 67

Хорошо – 68 – 83

Отлично – 84 – 100

Экзамен включает теоретическую и практическую часть для этого студенту необходимо:

1) решить задачу на составление алгоритма;

2) пройти тест по теоретическому материалу дисциплины(раздел 6.2.1).

В случае сдачи экзамена студент может набрать 10 баллов за прохождение теста и 7 баллов за правильно решенную задачу, однако в связи с недобором баллов по итогам текущей успеваемости для получения повышенной оценки студент автоматически теряет на экзамене 7 баллов.

Балльно-рейтинговая оценка результатов учебной работы обучающихся по видам (БРС)

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации	Баллы
Текущая учебная работа в семестре (Посещение занятий по расписанию и выполнение заданий)	60	Лекционные занятия (конспект)	1 балл посещение лекционного занятия	9 - 16
		Лабораторные работы и практические(отчет о выполнении работы)	2 балла - посещение 1 практического или лабораторного занятия и выполнение работы на 51-65%	18 - 36
			3 балла - посещение 1 практического или лабораторного занятия и выполнение работы на 66-85%	
			4 балла – посещение 1 занятия и существенный вклад на занятии в работу всей группы, самостоятельность и выполнение работы на 86-100%	
		Контрольные работы	24 балла (пороговое значение) 46 баллов (максимальное значение)	24 - 46
Итого по текущей работе в семестре				51 – 100 (%)
Промежуточная аттестация	40 (100 %)	Теоретическая часть	6 баллов (пороговое значение) 20 баллов (максимальное значение)	6 - 20

(экзамен)	/баллов приведенной шкалы)	Практическая часть	6 баллов (пороговое значение) 20 баллов (максимальное значение)	6 – 20
Итого по промежуточной аттестации (экзамену)				51 – 100% (по приведенной шкале к 12 – 40 баллам)
Суммарная оценка по дисциплине/ аттестации 51 – 100 б. Сумма баллов текущей и промежуточной				

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

а) основная учебная литература:

1. Белоцерковская, И. Е. Алгоритмизация. Введение в язык программирования С++ [Электронный ресурс] : учебное пособие / И. Е. Белоцерковская, Н. В. Галина, Л. Ю. Катаева. - 2-е изд., испр. – Эл.текстовые данные. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 197 с. : ил. –Режим доступа:<http://biblioclub.ru/index.php?page=book&id=428935>
2. Кузин, А. В. Программирование на языке Си [Электронный ресурс]: учебное пособие / А.В. Кузин, Е.В. Чумакова - Электронные текстовые данные. – Москва : Форум : ИНФРА-М, 2015. - 144 с. – Режим доступа:<http://znanium.com/catalog.php?bookinfo=505194>

б) дополнительная учебная литература:

1. Канцедал С.А. Алгоритмизация и программирование [Электронный ресурс] : Учебное пособие / С.А. Канцедал. - М.: ИД ФОРУМ: НИЦ ИНФРАМ, 2014. - 352 с Режим доступа: <http://znanium.com/catalog.php?bookinfo=429576>
2. Т.И. Немцова Т.И. Программирование на языке высокого уровня. Программирование на языке Object Pascal [Электронный ресурс] : Учеб. пос. / Под ред. Л.Г. Гагариной - М.: ИДФОРУМ: НИЦ Инфра- М, 2013 - 496с Режим доступа: <http://znanium.com/catalog.php?bookinfo=397789>

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

1. Лидовский В.В. Основы теории информации и криптографии: Учебный курс // <http://www.intuit.ru/>
2. Дехтярь М.И. Введение в схемы, автоматы и алгоритмы: Учебный курс // <http://www.intuit.ru/>
3. Князьков В.С., Волченская Т.В. Введение в теорию автоматов: Учебный курс // <http://www.intuit.ru/>
4. Сперанский Д.В. Теория экспериментов с конечными автоматами: Учебный курс // <http://www.intuit.ru/>
5. Местецкий Л.М. Математические методы распознавания образов: Учебный курс // <http://www.intuit.ru/>

9. Методические указания для обучающихся по освоению дисциплины

(модуля)

Вид учебных занятий	Организация деятельности студента
Лекция	Лекции построены на основе использования активных форм обучения: - лекция-беседа (преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание студентов к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей студентов), - проблемная лекция (с помощью проблемной лекции обеспечивается достижение трех основных дидактических целей: усвоение студентами теоретических знаний; развитие теоретического мышления; формирование познавательного интереса к содержанию учебного предмета и профессиональной мотивации будущего специалиста), -- лекция с заранее запланированными ошибками (Эта форма проведения лекции необходима для развития у студентов умений оперативно анализировать профессиональные ситуации, выступать в роли экспертов, оппонентов, рецензентов, вычленять неверную или неточную информацию). На каждой лекции применяется сочетание этих форм обучения в зависимости от подготовленности студентов и вопросов, вынесенных на лекцию. Присутствие на лекции не должно сводиться лишь к автоматической записи изложения предмета преподавателем. Более того, современный насыщенный материал каждой темы не может (по времени) совпадать с записью в тетради из-за разной скорости процессов – мышления и автоматической записи. Каждый студент должен разработать для себя систему ускоренного фиксирования на бумаге материала лекции. Поэтому, лектором рекомендуется формализация записи посредством использования общепринятых логико-математических символов, сокращений, алгебраических (формулы) и геометрических (графики), системных (схемы, таблицы) фиксаций изучаемого материала. Овладение такой методикой, позволяет каждому студенту не только ускорить процесс изучения, но и повысить его качество, поскольку успешное владение указанными приемами требует переработки, осмысления и структуризации материала.
Лабораторная работа	Вузовская подготовка специалистов должна обеспечивать приобретение ими не только знаний, но и умений использовать полученные знания на практике. Это требование и положено в основу целей и методов проведения лабораторных работ по вышеуказанной учебной дисциплине. Лабораторные работы предлагаются в соответствии с рабочей программой в рамках каждой темы.
Подготовка к экзамену	Подготовка к экзамену предполагает изучение рекомендуемой литературы и других источников, конспектов лекций, повторение материалов практических занятий.

Методические указания размещены на сайте *НФИ КемГУ* <https://eios.nbikemsu.ru/>

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Материально-техническая база

Учебные занятия по дисциплине проводятся в учебных аудиториях НФИ КемГУ:

Языки программирования	303 Компьютерный класс. Учебная аудитория (мультимедийная) для проведения занятий: - лабораторного (практического) типа; - групповых и индивидуальных консультаций; - текущего контроля и промежуточной аттестации.	654027, Кемеровская область - Кузбасс, г. Новокузнецк, пр-кт Пионерский, д.13,
------------------------	--	--

	Специализированная (учебная) мебель: доска маркерно-меловая, столы компьютерные, стулья. Оборудование для презентации учебного материала: <i>стационарное</i> - ноутбук преподавателя, экран, проектор. Оборудование: компьютеры для обучающихся (11 шт.). Используемое программное обеспечение: MSWindows (MicrosoftImaginePremium 3 year по лицензионному договору № 1212/KMP от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), BloodshedDevC++ 4.9.9.2 (свободно распространяемое ПО), Java (бесплатная версия), MicrosoftSQLServer 2008 (MicrosoftImaginePremium 3 year по лицензионному договору № 1212/KMP от 12.12.2018 г. до 12.12.2021 г.), Яндекс.Браузер (отечественное свободно распространяемое ПО), OracleVMVirtualBox (бесплатная версия), FreePascal(свободно распространяемое ПО), Lazarus(свободно распространяемое ПО), Pascal ABC.NET(свободно распространяемое ПО). Интернет с обеспечением доступа в ЭИОС.	пом. 2
--	---	--------

Составитель (и): Можаров М.С. , профессор

(фамилия, инициалы и должность преподавателя (ей))



Составитель (и): Можаров М.С. , профессор

(фамилия, инициалы и должность преподавателя (ей))



Макет рабочей программы дисциплины (модуля) разработан в соответствии с приказом Минобрнауки России от 19.12.2013 № 1367, одобрен научно-методическим советом (протокол № 8 от 09.04.2014 г.) и утвержден приказом ректора от 23.04.2014 № 224/10..

Макет обновлён с поправками в части подписей на титульной странице, п.3 добавлена строка для указания часов, проводимых в активной и интерактивной формах обучения, добавлен п. 12.1 Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья (протокол НМС № 6 от 15.04.2015 г.), утвержден приказом ректора.