



Федеральное агентство по рыболовству
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Калининградский государственный технический университет»
(ФГБОУ ВО «КГТУ»)

УТВЕРЖДАЮ
Начальник УРОПС

Фонд оценочных средств
(приложение к рабочей программе модуля)
«WEB-ТЕХНОЛОГИИ»

основной профессиональной образовательной программы специалитета
по специальности

**10.05.03 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ АВТОМАТИЗИРОВАННЫХ
СИСТЕМ**

Специализация

«БЕЗОПАСНОСТЬ ОТКРЫТЫХ ИНФОРМАЦИОННЫХ СИСТЕМ»

ИНСТИТУТ
РАЗРАБОТЧИК

цифровых технологий
кафедра информационной безопасности

1. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Таблица 1 – Планируемые результаты обучения по дисциплине, соотнесенные с установленными индикаторами достижения компетенций

| Код и наименование компетенции | Индикаторы достижения компетенции | Дисциплина | Результаты обучения (владения, умения и знания), соотнесенные с компетенциями/индикаторами достижения компетенции |
|---|--|----------------|--|
| ОПК-7: Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ. | ОПК-7.3: Использует различные методы программирования при разработке веб-сервисов. | Web-технологии | <p><u>Знать:</u> структуру и общепринятые элементы веб-страниц различных видов и назначений; методы обеспечения доступа к страницам веб-сайтов.</p> <p><u>Уметь:</u> создавать, использовать и оптимизировать изображения для веб-сайтов; создавать SQL-запросы; разрабатывать объектно-ориентированный программный код.</p> <p><u>Владеть:</u> навыками разработки веб-сервисов с применением PHP, Python, XML; различными методами программирования.</p> |

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПОЭТАПНОГО ФОРМИРОВАНИЯ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ (ТЕКУЩИЙ КОНТРОЛЬ) И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

2.1 Для оценки результатов освоения дисциплины используются:

- оценочные средства текущего контроля успеваемости;
- оценочные средства для промежуточной аттестации по дисциплине.

2.2 К оценочным средствам текущего контроля успеваемости относятся:

- тестовые задания;
- задания и контрольные вопросы по лабораторным работам;

2.3 К оценочным средствам для промежуточной аттестации по дисциплине, проводимой в форме зачета, относятся:

- промежуточная аттестация в форме зачета проходит по результатам прохождения всех видов текущего контроля успеваемости.

3. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

3.1 Тестовые задания предназначены для оценки в рамках текущего контроля успеваемости знаний, приобретенных студентами на лекционных и лабораторных занятиях, и для измерения соответствующих индикаторов достижения компетенции.

Содержание теста определяется в соответствии с содержанием дисциплины пропорционально учебному времени, отведенному на изучение разделов, перечисленных в рабочей программе модуля.

Время выполнения теста 70 мин.

Типовые варианты тестовых заданий приведены в Приложении №1, ключи к ним – в Приложении №4.

Шкала оценивания тестовых заданий основана на системе зачет/незачет.

| незачет | зачет |
|-------------------------------|-----------------------------|
| Менее 50% правильных ответов. | 51-100% правильных ответов. |

Результаты измерений индикатора считаются положительными при правильном выполнении не менее 50% заданий.

3.2 Темы и образцы заданий для лабораторных занятий приведены в Приложении №2. В Приложении №3 приведены задания для защиты лабораторных работ.

Результаты измерений индикатора считаются положительными при положительной оценке за выполнение задания.

4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

4.1 Промежуточная аттестация по дисциплине проводится в форме зачета (6 семестр).

Промежуточная аттестация проходит по результатам прохождения всех видов текущего контроля успеваемости.

Компетенции в той части, в которой они должны быть сформированы в рамках изучения дисциплины, могут считаться сформированными в случае, если студент получил зачет.

5. СВЕДЕНИЯ О ФОНДЕ ОЦЕНОЧНЫХ СРЕДСТВ И ЕГО СОГЛАСОВАНИИ

Фонд оценочных средств для аттестации по дисциплине «Web-технологии» представляет собой компонент основной профессиональной образовательной программы специалитета по специальности 10.05.03 Информационная безопасность автоматизированных систем (специализация «Безопасность открытых информационных систем»).

Фонд оценочных средств рассмотрен и одобрен на заседании кафедры информационной безопасности 20.04.2022 г. (протокол № 7).

Заведующий кафедрой



Н.Я. Великите

Приложение № 1

ТЕСТОВЫЕ ЗАДАНИЯ ПО ДИСЦИПЛИНЕ

| ВАРИАНТ 1 | |
|------------------|---|
| 1. | В иерархии объектов самым высшим объектом является: 1. window 2. document 3. forms |
| 2. | При создании иерархии можно пренебречь объектом: 1. document 2. все объекты необходимо прописать 3. window |
| 3. | Свойство <i>value</i> соответствует: 1. ссылке 2. введенному тексту 3. графическому изображению |
| 4. | При работе в иерархической системе объектам давать имена: 1. необязательно, можно обратиться к объектам по их нумерованным названиям 2. обязательно 3. именованье объектов вообще не предусмотрено |
| 5. | Объектам даются имена для: 1. корректной работы программы 2. создания функций, обрабатывающих движение мыши 3. удобства программиста |
| 6. | Путь к объекту прослеживается с помощью: 1. тире 2. точки 3. запятой |
| 7. | Объект <i>location</i> содержит: 1. адрес загруженного HTML-документа 2. графическое изображение 3. текстовое поле |
| 8. | Объект <i>images</i> содержит: 1. адрес загруженного HTML-документа 2. графическое изображение 3. текстовое поле |
| 9. | Объект <i>window</i> необходимо указывать в иерархии, если сайт состоит из: 1. одной страницы 2. одной страницы и содержит несколько форм 3. нескольких страниц |
| 10. | События инициируются: 1. браузером, независимо от действий пользователя 2. действиями пользователя 3. протоколом, независимо от действий пользователя |
| 11. | Событие <i>MouseOver</i> происходит: 1. в результате щелчка по кнопке 2. если указатель мыши покидает ссылку гипертекста 3. если указатель мыши пересекает ссылку гипертекста |

| | |
|------------------|---|
| 12. | Событие <i>MouseOut</i> происходит: 1. в результате щелчка по кнопке 2. если указатель мыши покидает ссылку гипертекста 3. если указатель мыши пересекает ссылку гипертекста |
| 13. | Событие <i>onClick</i> происходит: 1. в результате щелчка по кнопке 2. если указатель мыши покидает ссылку гипертекста 3. если указатель мыши пересекает ссылку гипертекста |
| 14. | Атрибут <i>target</i> при работе с фреймами: 1. запрещает переход между фреймами 2. дает имя фрейму 3. осуществляет обращение к именованному фрейму |
| 15. | Навигация по странице может быть организована с помощью: 1. фреймов 2. таблиц 3. форм |
| ВАРИАНТ 2 | |
| 1. | Функция <i>setTimeout</i> в JavaScript: 1. устанавливает время существования сайта 2. обрабатывает нажатие на кнопку 3. устанавливает интервал времени работы другой функции |
| 2. | Функция <i>statbar</i> в JavaScript: 1. обрабатывает нажатие на кнопку 2. выполняет запись в строку состояния 3. создает выпадающее меню |
| 3. | Функция <i>scroll</i> в JavaScript: 1. программирует прокрутку страницы 2. выполняет запись в строку состояния 3. создает выпадающее меню |
| 4. | Содержимое текстового поля можно проверить с помощью свойства: 1. <i>target</i> 2. <i>value</i> 3. <i>text</i> |
| 5. | Способ организации информации на web-сервере называется: 1. файлом 2. мультимедиа 3. web-сайтом |
| 6. | Для тега <i>form</i> обязательный параметр: 1. <i>target</i> 2. <i>action</i> 3. <i>type</i> |
| 7. | Страницу, созданную с помощью фрейма, от искажения защищает свойство: 1. <i>target</i> 2. <i>noresize</i> 3. <i>text</i> |
| 8. | Если при создании окон используется команда: <code>myWin= open("abc.htm", "dispWindow", "width=400,height=300,status=no,toolbar=no,menubar=no");</code> то именем окна является переменная: |

| | |
|------------------|--|
| | <ol style="list-style-type: none"> 1. myWin 2. dispWindow 3. обе переменные могут быть именем окна |
| 9. | <p>Окно закрывается с помощью функции:</p> <ol style="list-style-type: none"> 1. erase() 2. clos_it() 3. close() |
| 10. | <p>В JavaScript не является предопределенным объект:</p> <ol style="list-style-type: none"> 1. Massiv 2. Date 3. Math |
| 11. | <p>В предопределенном объекте Date содержатся методы:</p> <ol style="list-style-type: none"> 1. getHours(), postHours(), getMinutes(), postMinutes() 2. actHours(), setHours(), actMinutes(), setMinutes() 3. getHours(), setHours(), getMinutes(), setMinutes() |
| 12. | <p>Правильное обращение к методу предопределенного объекта:</p> <ol style="list-style-type: none"> 1. Math_random() 2. Math.random() 3. Math=random() |
| 13. | <p>В предопределенном объекте Math не содержится метод:</p> <ol style="list-style-type: none"> 1. sin() 2. cos() 3. array() |
| 14. | <p>Схема drag&drop означает:</p> <ol style="list-style-type: none"> 1. «перетащил и оставил» 2. «нашел и уничтожил» 3. «перетащил и переименовал» |
| 15. | <p>Используя механизм drag&drop можно, нажав клавишу мыши:</p> <ol style="list-style-type: none"> 1. только помечать объекты 2. перетаскивать объекты 3. только удалять объекты |
| ВАРИАНТ 3 | |
| 1. | <p>RНР поддерживает парадигму программирования:</p> <ol style="list-style-type: none"> 1. объектно-ориентированную 2. логическую 3. функциональную |
| 2. | <p>При передаче ассоциативного массива в функцию sort() результат выполнения функции: массив отсортируется по.....</p> <ol style="list-style-type: none"> 1. значениям; ключи элементов отсортированы не будут и не будут сохранены + 2. значениям; ключи элементов отсортированы не будут, но будут сохранены 3. ключам, значения элементов отсортированы не будут |
| 3. | <p>Правильное объявление массива в языке RНР:</p> <ol style="list-style-type: none"> 1. \$arr[«a»,»b»,»c»] = «q»; 2. \$arr («0"=> «a»); 3. \$arr = array(«a»,»b»,»c»); |
| 4. | <p>Отличие клиента от сервера:</p> <ol style="list-style-type: none"> 1. сервер создает соединение, а клиент, пользуясь им, передает свой запрос 2. сервер есть часть компьютерной архитектуры сервер-сервер, а клиент – архитектуры клиент-клиент |

| | |
|-----|---|
| | 3. клиент посылает запросы, а сервер обрабатывает их |
| 5. | Одна из функций сервера: 1. обрабатывать запросы клиента 2. отображать данные на экране компьютера пользователя 3. инициировать соединение с клиентом |
| 6. | Если имеется объект (\$obj) какого-то класса, то получить список всех свойств класса можно: 1. <code>get_class_var(get_class(\$obj))</code> 2. <code>\$get_class_vars(get_class->\$obj);</code> 3. <code>\$vars = get_class_vars(get_class(\$obj));</code> |
| 7. | Узнать имя класса, представителем которого является объект, можно: 1. <code>get_class(объект);</code> 2. <code>\$(get_class->объект);</code> 3. <code>class(объект);</code> |
| 8. | Текущее смещение чтения/записи файла можно получить с помощью функции: 1. <code>fread</code> 2. <code>ftell()</code> 3. <code>fseek()</code> |
| 9. | Добавляет новый элемент в начало массива функция: 1. <code>array_shift()</code> 2. <code>array_pop()</code> 3. <code>array_unshift()</code> |
| 10. | Существование файла можно проверить с помощью функции: 1. <code>is_readable()</code> 2. <code>file_exists()</code> 3. <code>is_writable()</code> |
| 11. | Правильная запись оператора удаления таблицы: 1. <code>REMOVE TABLE [IF EXISTS] имя_таблицы [, имя_таблицы,...] [RESTRICT CASCADE]</code> 2. <code>DELETE TABLE [IF EXISTS] имя_таблицы [, имя_таблицы,...] [RESTRICT CASCADE]</code> 3. <code>DROP TABLE [IF EXISTS] имя_таблицы [, имя_таблицы,...] [RESTRICT CASCADE]</code> |
| 12. | Правильный запрос к таблице описаний статей (Articles) и изменение названия статьи (title), автор (author) которой «Петров» на название «Второе название»: 1. <code>UPDATE Articles SET title=»Второе название»;</code> 2. <code>UPDATE Articles SET title=»Второе название» WHERE author=»Петров»;</code> 3. <code>UPDATE Articles SET title=»Второе название» WHERE title=» Первое название»;</code> |
| 13. | После уничтожения сессии командой <code>session_destroy();</code> : 1. уничтожается ее идентификатор, мы больше не можем ни зарегистрировать переменные, ни вообще производить какие-либо действия с сессией 2. очищается массив <code>\$_SESSION</code> и больше ничего не происходит 3. массив <code>\$_SESSION</code> заполняется нулями |
| 14. | С помощью какой функции можно получить идентификатор сессии: 1. <code>id_session</code> 2. <code>session_id</code> 3. <code>session_identificator</code> |

| | |
|-----|---|
| 15. | Общая структура регулярного выражения – шаблон: 1. выделенный с помощью специального символа разделителя, модификатор, влияющий на способ обработки регулярного выражения 2. выделенный с помощью специального символа разделителя 3. заключенный в круглые скобки и функция для его обработки |
|-----|---|

Приложение № 2

ТЕМЫ И ОБРАЗЦЫ ЗАДАНИЙ ДЛЯ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Тема 1. Иерархия объектов, события, функции.

Пример 1.1

Написать сценарий, определяющий площадь прямоугольного треугольника по заданным катетам. Сценарий разместим в разделе `<body>` HTML-документа.

Пример 1.2

Для трех заданных значений a , b , c написать сценарий, определяющий максимальное значение.

Тема 2. Формы

Пример 2.1

Ввести в текстовое поле формы номер телефона, дополнить скрипт процедурой проверки цифр и некоторых символов (специальных символов, букв).

Пример 2.2

Создать программу ввода информации в текстовые поля формы и отправки этой информации на сервер. Предусмотреть проверку ввода символа `@`.

Тема 3. Предопределенные объекты

Пример 3.1

С помощью метода `Date()` создать программу работающих часов.

Пример 3.2

С помощью метода `Array()` создать одномерный массив и найти в нем максимальный и минимальный элементы.

Тема 4. Окна и динамически создаваемые документы

Пример 4.1

Создать динамически открывающееся окно и записать в него html-страницу. Предусмотреть закрытие окна.

Пример 4.2

Открыть объект `document` для последующей печати, сгенерировать новый документ, закрыть документ (но не окно!)

Тема 5. Drag & Drop

Пример 5.1

Создать объект `window`, используя метод `captureEvent()`, перехватить сигнал о событии `Click`.

Пример 5.2

Используя метод `MouseMove`, отобразить текущие координаты курсора мыши в окне состояния.

Приложение №3.

ЗАДАЧИ ДЛЯ ЗАЩИТЫ ЛАБОРАТОРНЫХ РАБОТ

1. Стандартный объект *Math* имеет метод *max*, который определяет наибольшее значение двух аргументов. С помощью метода *max* определить максимальное из трех введенных в текстовое поле чисел.
2. Написать сценарий, определяющий площадь квадрата по заданной стороне.
3. Написать сценарий, который меняет местами значения двух введенных переменных (использовать событие *Blur*).
4. Вводится последовательность из пяти чисел. Напишите сценарий, в котором определяется число максимальных элементов.
5. Написать сценарий, который реализует обмен рисунков в документе.
6. Написать сценарий, реализующий вертикальное графическое меню. При наведении курсора мыши на пункт меню меняется цветовая палитра, соответствующая выделенному пункту меню.
7. Написать сценарий картинки с «эффектом приближения», т.е. увеличения размеров как реакция на попадание курсора мыши в поле рисунка (использовать свойства *width* и *height*).
8. Написать сценарий, который предоставляет возможность пользователю задавать значения параметров, определяющих, к какому полю окна (левому или правому) прижимается изображение, и, соответственно, с какой стороны текст его обтекает.
9. Написать сценарий обработки анкеты слушателя курсов. Пользователь может выбрать курс, его продолжительность, язык, на котором он готов работать с преподавателем, и форму отчетности.
10. Написать сценарий, который позволяет посчитать стоимость предполагаемой покупки. Задается список продуктов, цена за единицу товара и количество экземпляров.
11. Создайте документ, разбивающий окно просмотра с помощью фреймов на две прямоугольные области: верхнюю и нижнюю. В верхней области поместите оглавление в виде списка, при выборе пунктов которого соответствующий раздел должен появляться в нижней части окна.
12. Напишите программу, определяющую все делители заданного натурального числа.
13. Написать сценарий, с помощью которого определяются все даты в указанном году, приходящиеся на пятницу, 13 число.
14. Написать сценарий, который по заданной дате определяет номер недели в году.
15. Слова в заданном тексте разделяются пробелами. Напишите программу, которая определяет количество слов в тексте.
16. Создать простейший мультипликационный сюжет с использованием массива.
17. Задан одномерный массив вещественных чисел. Напишите сценарий, который определяет число положительных элементов массива.
18. Задан одномерный массив вещественных чисел. Напишите сценарий, позволяющий найти максимальный элемент в массиве.
19. Создать динамически открывающийся документ и загрузить в него изображение.
20. С помощью механизма Drag & Drop создать движущийся объект.