Балаковский инженерно-технологический институт – филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»

Факультет атомной энергетики и технологий
Кафедра «Информационные системы и технологии»

РАБОЧАЯ ПРОГРАММА

по дисциплине «Основы web-программирования»

Направления подготовки/специальность
«09.03.02 Информационные системы и технологии»

Основная профессиональная образовательная программа
«Информационные системы и технологии»

Квалификация выпускника
Бакалавр

Форма обучения
Очная

Балаково

Цель освоения дисциплины

Целями освоения дисциплины «Основы web-программирования» в области обучения, воспитания, развития, соотнесенные с общими целями ООП ВО и требованиями профессиональных стандартов («06.016. Руководитель проектов в области информационных технологий», «06.015. Специалист по информационным системам») являются: формирование профессиональной ответственности в области проектирования, разработки, эксплуатации и сопровождения информационных систем и технологий, используемых для информационного обеспечения деятельности различных предприятий и организаций, в том числе организаций атомной отрасли; формализация и алгоритмизация поставленных задач; написание программного кода с использованием языков программирования, определения и манипулирования данными; проверка и отладка программного кода; проверка работоспособности программного обеспечения; анализ требований к программному обеспечению.

Место дисциплины в структуре ООПВО

Необходимыми условиями для освоения дисциплины являются знания, умения и практические навыки по предшествующим дисциплинам и практикам в соответствии с требованиями освоения компетенций:

Интерактивное программирование web-приложений / Создание интерактивных приложений для интернет

Управление данными

Знания, умения и практические навыки, полученные при освоении дисциплины, необходимы при изучении следующих дисциплин и прохождения практик:

Методы и средства проектирования информационных систем и технологий

Современные технологии интернет-программирования

Информационный менеджмент

Электронный документооборот / Системы электронного документооборота на предприятии

Производственная практика (проектно-технологическая)

Производственная практика (преддипломная)

Государственная итоговая аттестация

При освоении данной дисциплины студент сможет частично продемонстрировать следующие обобщенные трудовые функции: Разработка архитектуры ИС, Проектирование и дизайн ИС, Сбор информации для инициации проекта в соответствии с полученным заданием

При освоении данной дисциплины студент сможет частично продемонстрировать следующие обобщенные трудовые функции: Разработка архитектуры ИС, Проектирование и дизайн ИС, Сбор информации для инициации проекта в соответствии с полученным заданием

Компетенции обучающегося, формируемые в результате освоениядисциплины

В процессе освоения данной дисциплины у студента формируются следующие компетенции:

профессиональные

Задачи профессиональной деятельности (ЗПД)	Объект или область знания	Код и наименование компетенции	Индикаторы достижения компетенции
проектирование базовых и прикладных информационных технологий	информационные процессы, технологии, системы и сети, их инструментальное (программное, техническое, организационное) обеспечение, способы и методы проектирования, отладки, производства и эксплуатации информационных технологий и систем в различных областях и сферах деятельности.	ПК-6.1 Способен разрабатывать и адаптировать прикладное программное обеспечение	З-ПК-6.1 Знать: принципы и методы разработки, внедрения и адаптации прикладного программного обеспечения У-ПК-6.1 Уметь: программировать приложения и создавать программные прототипы решения прикладных задач В-ПК-6.1 Владеть: инструментальными средствами проектирования и разработки прикладного программного обеспечения
выбор исходных данных для проектирования	информационные процессы, технологии, системы и сети, их инструментальное (программное, техническое, организационное) обеспечение, способы и методы проектирования, отладки, производства и эксплуатации информационных технологий и систем в различных областях и сферах деятельности	ПК-12 Способен проводить выбор исходных данных для проектирования с учетом требований заказчика	З-ПК-12 Знать: требования к разработке технического задания, его структуру и принципы составления У-ПК-12 Уметь: анализировать исходную документацию заказчика В-ПК-12 Владеть: методикой составления технического задания

Задачи воспитания, реализуемые в рамках освоения дисциплины

Направление/цели	Создание условий, обеспечивающих	Использование воспитательного потенциала учебной дисциплины	Вовлечение в разноплановую внеучебную деятельность
Профессиональное воспитание	формирование профессиональной ответственности в области проектирования, разработки, эксплуатации и сопровождения	1. Использование для формирования чувства личной ответственности в области проектирования, разработки, эксплуатации и сопровождения информационных систем и технологий, используемых для информационного обеспечения деятельности различных предприятий и организаций, в том числе организаций атомной отрасли	1. Организация и проведение научно-практических конференций, форумов, круглых столов, вебинаров по вопросам профессиональной деятельности 2. Участие в

	информационных систем и технологий, используемых для информационного обеспечения деятельности различных предприятий и организаций, в том числе организаций атомной отрасли (В26).	воспитательного потенциала блока профессиональных дисциплин: Управление данными; Интерактивное программирование web-приложений; Создание интерактивных приложений для интернет; Технологии создания web –ресурсов; Основы web-программирования; Информационная теория управления 2. Развитие навыков творческого мышления путем содействия и поддержки участия студентов в научно-практических мероприятиях внутривузовского регионального и/или всероссийского уровня по информатике и информационным технологиям.	студенческих олимпиадах и конкурсах научных проектов, творческих мероприятиях, конкурсах профессионального мастерства, в том числе по стандартам WorldSkills. 3. Участие в подготовке публикаций в периодических научных изданиях; 4. Участие в деятельности студенческого научного общества

Структура и содержание учебнойдисциплины

Дисциплина преподается студентам на очной форме обучения в 5-ом семестре. Общая трудоемкость дисциплины составляет 5 зачетных единицы, 180 ак. часа.

Календарный план

№ Раздела	№ Темы	Наименование раздела (темы) дисциплины	Виды учебной деятельности (в часах)					Аттестация раздела (форма)	Максимальный балл за раздел
			Всего	Лекции	Лабораторные	Практические	СРС		
1	**Основы программирования на языке PHP**							КИ	**25**
	1	Переменные и константы. Условные операторы и ветвления. Циклы. Массивы и функции для работы с массивами.	24	2	-	2	20		
	2	Функции и создание собственных функций. Получение данных из HTML форм. Метод Get и Post.	28	2	-	6	20		
	3	Язык SQL и работа с базами данных. Операторы UPDATE, операторы DELETE.	30	2	-	8	20		
2	**Взаимодействие PHP с XML. ООП.**							КИ	**25**
	4	Работа с XML и файлами	32	4/2	-	8	20		
	5	Объектно-ориентированное программирование. Классы, геттеры, сеттеры. Инкапсуляция. Наследование.	30	6/4	-	8	16		
Вид промежуточной аттестации			**36**					**Э**	**50**
Итого			**180/16**	**16/6**	**-**	**32/10**	**96**		**100**

КИ- контроль по итогам
* - занятия в интерактивной форме

Сокращенное наименование форм текущего контроля и аттестации разделов:

Обозначение	**Полное наименование**
КИ	Контроль по итогам
Э	Экзамен

Содержание лекционного курса

Темы лекции. Вопросы, отрабатываемые на лекции	**Всего часов**	**Учебно-методическоеобеспечение**
1	**2**	**3**
Лекция 1. Переменные и константы. Условные операторы и ветвления. Циклы. Массивы и функции для работы с массивами. 1. Переменные в PHP. 2. Переменные PHP и HTML. 3. Типы переменных и константы. 4. Константы. 5. Ветвления. Условные операторы if else. 6. Условный оператор switch. 7. Цикл while. 8. Цикл do while. 9. Цикл for. 10. Массивы в PHP. 11. Функция count() и циклы. 12. Цикл foreach. 13. Функции для работы с массивами. 14. Уникальные элементы массива. 15. Вставка элементов в массив. 16. Функции implode и explode. 17. Многомерные массивы в php. 18. Способы создания многомерных массивов.	2	1-2
Лекция 2. Функции и создание собственных функций. Получение данных из HTML форм. Метод Get и Post. 1. Получение данных из HTML формы. Методы GET и POST. 2. Функции для работы со строками. Защита формы сайта. 3. Создание собственных функций. Валидация функций 4. Функции. Взаимодействие функций между собой	2	1-2
Лекция 3. Язык SQL и работа с базами данных. Операторы UPDATE, операторы DELETE. 1. MySQL и PHPMYADMIN. Начало работы 2. Вывод данных из MySQL. Оператор SELECT 3. Вставка данных в MySQL. Оператор INSERT 4. Обновление данных в таблице MySQL. Оператор UPDATE 5. Удаление данных из таблицы MySQL. Оператор DELETE 6. Проектирование базы данных. Хранение данных в разных таблицах 7. SQL запросы. INNER JOIN	4	1-2

8. SQL запросы. LEFT JOIN и RIGHT JOIN		
Лекция 4. Работа с XML и файлами. 1. Работа с XML. Чтение XML и вывод в браузер в виде массива 2. Работа с XML. Создадим функцию для записи данных из XML в MySQL 3. Работа с XML. Запишем данные из XML в базу 4. Работа с файлами с помощью PHP	4	1-2
Лекция 5. Объектно-ориентированное программирование (ООП) 1. ООП. Классы, Геттеры и Сеттеры 2. ООП. Инкапсуляция 3. ООП. Наследование 4. ООП. Полиморфизм. Интерфейсы 5. ООП. Полиморфизм. Абстрактные классы	6	1-2

Перечень практических занятий

Тема практического занятия. Вопросы, отрабатываемые на практическом занятии	Всего часов	Учебно-методическое обеспечение
1	**2**	**3**
Математические операции с использованием PHP. Функции пользователя в PHP	4	1-2
Циклы, массивы, функции в PHP. Работа с файлами в PHP	4	1-2
Создание базы данных MySQL. Простой вывод данных в PHP	4	1-2
Ввод и правка данных с помощью формы в PHP	4	1-2
Работа с заметками в PHP	4	1-2
Страница статистики inform.php	4	1-2
Реализация поиска по сайту	4	1-2
Передача файлов на сервер	4	1-2

Перечень лабораторных работ – не предусмотрен учебным планом

Задания для самостоятельной работы студентов

Вопросы для самостоятельного изучения (задания)	Всего часов	Учебно-методическое обеспечение
1	**2**	**3**
Основы синтаксиса PHP. Строки и числа. Переменные. Операция присваивания. Комментарии. Функции. Функции echo. Математические операции. Специальные значения. Значения true, falce. Значение null. Функция var_dump. PHP внутри HTML. Сложение строк в PHP. Получение определенного символа строки. Сложности с операцией присваивания. Операции инкремента и декремента. Операции +=, -=, *=, .=. Операция .=. Типизация переменных. Вывод отдельного элемента массива. Ассоциативный массив. Хитрость с ключами. Методы создания массива.	20	1-2

Многомерный массив. Конструкция if-else. Сокращенный синтаксис. Равенство по значению и типу. Равенство по значению и типу. Не равно. Все операции сравнения. Функция empty. Функция isset. Несколько условий сразу: or и end. Работа с логическими переменными. Работа вместе с HTML. Вложенные if. Конструкция elseif. Несколько if. Конструкция switch-case. Цикл foreach. Цикл while. Бесконечный цикл. Цикл for. Цикл без тела. Несколько команд в цикле for. Инструкция break. Инструкция continue.		
Работа со строковыми функциями в PHP. Функции работы с массивами. Создания форм на HTML. Поля формы. Получение данных из формы в PHP. Сохранение значений полей формы после отправки. Функция time, формат timestamp. Функция mktime. Функция date. Второй параметр функции date. Функция strtotime. Добавление и отнимание даты. Работа с пользовательскими функциями. Синтаксис функция пользователя. Вызов функции в коде. Параметры пользовательской функции. Инструкция return. Частая ошибка с return. Приёмы работы с return. Флаги. Флаги в функции. Логические операции без if. Цикл в цикле. Строка 'xxxxxxxxxx'. While без заданного количества итераций.	20	1-2
PhpMyAdmin. AUTO_INCREMENT. Типы переменных. Как работать с MySQL через PHP. Установка соединения с БД. Посылаем запросы к базе данных. Отлавливаем ошибки базы данных. Проблемы с кодировкамию. Тестирование работоспособности. Получение результата. Команда SELECT - выборка из БД. Примеры работы с SELECT. Более сложная логика: OR и AND. Группировка условий. Выбор столбцов. Команда INSERT - вставка данных в БД. Другой синтаксис INSERT. Массовая вставка через INSERT. Команда DELETE - удаление записей. ORDER BY - сортировка. LIMIT - ограничение количества. Команда COUNT - считаем количество. Команда LIKE - реализуем поиск. Кавычки ' '.	20	1-2
Объектная модель XML - документа. Расширения SAX и DOM XML. Установка расширения DOM XML. Взаимодействие PHP и XML посредством DOM XML. Перевод данных XML-файла в объекты и классы PHP. Добавление новых элементов в XML-документ. Работа с атрибутами. Теги с дефисами. Перебор циклом. Из объекта в нормальный массив. Парсинг на основе sitemap.xml.	20	1-2
Класс PHP. Свойства класса в PHP. Конструкторы для классов. Методы для классов. Объекты в PHP. Инкапсуляция. Уровни доступа. Доступ public. Доступ private. Доступ protected. Наследование. Полиморфизм. Константы классов. Автоматическая загрузка классов. Конструкторы и деструкторы. Область видимости. Наследование. Оператор разрешения области видимости. Ключевое слово static. Абстрактные классы. Интерфейсы объектов. Трейты. Анонимные классы. Перегрузка. Итераторы объектов. Магические методы. Ключевое слово final. Клонирование объектов. Сравнение объектов. Позднее статическое связывание.	16	1-2

Расчетно-графическая работа не предусмотрена учебным планом

Курсовая работа не предусмотрена учебным планом

Образовательные технологии

При реализации учебного материала курса используются различные образовательные технологии, способствующие созданию атмосферы свободной и творческой дискуссии как между преподавателем и студентами, так и в студенческой группе. Целью при этом является выработка у студентов навыков и компетенций, позволяющих самостоятельно вести исследовательскую и научно-педагогическую работу.

Практическая подготовка при реализации учебной дисциплины организуется путем проведения практических занятий, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

Практическая подготовка также включает в себя занятия лекционного типа, которые предусматривают передачу учебной информации обучающимся, необходимой для последующего выполнения работ, связанных с будущей профессиональной деятельностью.

Аудиторные занятия проводятся в виде лекций, практических занятий с использованием ПК и компьютерного проектора. Самостоятельная работа студентов проводится под руководством преподавателей, с оказанием консультаций и помощи при подготовке к практическим занятиям.

Фонд оценочных средств

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

№ п/п	Наименование контролируемых разделов (темы)	Код и наименование индикатора достижения компетенций	Наименование оценочного средства
Входной контроль			
1	Входной контроль		Вопросы входного контроля (письменно)
Аттестация разделов, текущий контроль успеваемости			
2	Реализация сценариев JavaScript	З-ПК-6.1, У-ПК-6.1, В-ПК-6.1, З-ПК-12, У-ПК-12, В-ПК-12	Контроль итогов в форме тестирования (письменно)
3	Реализация сценариев jQuery	З-ПК-6.1, У-ПК-6.1, В-ПК-6.1, З-ПК-12, У-ПК-12, В-ПК-12	Контроль итогов в форме тестирования (письменно)
Промежуточная аттестация			
4	Экзамен	З-ПК-6.1, У-ПК-6.1, В-ПК-6.1, З-ПК-12, У-ПК-12, В-ПК-12	Экзаменационный тест (письменно)

Входной контроль предназначен для выявления пробелов в знаниях студентов и готовности их к получению новых знаний. Оценочные средства для входного контроля представляют собой вопросы, которые задаются студентам в письменной форме.

Перечень вопросов входного контроля

1. Основные этапы в развитии HTML.
2. Теговая модель и базовая структура HTML-документов.
3. Основные требования к заглавной части HTML.
4. Дерево элементов HTML. Родственные связи между элементами. Принципы наследования.
5. Основные элементы HTML для форматирования текста.
6. Дополнительные (вспомогательные) элементы HTML для форматирования текста.
7. Основные элементы HTML для вставки изображений и создания гиперссылок.
8. Основные элементы HTML для работы со списками.
9. Основные элементы HTML для работы с таблицами.
10. Блочные и строчные элементы HTML. Определения и основные особенности.
11. Универсальные элементы HTML. Назначение и принципы использования.
12. Атрибуты элементов HTML. Принципы наследования. Универсальные атрибуты.
13. Адресация в HTML. Варианты и примеры абсолютной и относительной адресации.
14. Каскадные таблицы стилей CSS. Предпосылки появления и история развития.
15. Основы синтаксиса CSS. Назначение и особенности использования.
16. Методы определения CSS. Встраивание, вложение и связывание.
17. Методы определения CSS. Принципы каскадирования и наследования стилей.
18. Единицы измерения в CSS. Перечень абсолютных и относительных единиц измерения.
19. Способы задания цвета в CSS. Цветовые таблицы (палитры). Принципы подбора цвета.
20. Шрифтовое оформление в CSS. Гарнитуры. Семейство и тип шрифта. Понятие о «безопасных» шрифтах.
21. Шрифтовое оформление в CSS. Настройка типа, размера, начертания и модификации шрифта.
22. Собирательное шрифтовое оформление.
23. Оформление текста в CSS. Выравнивание, отступы и промежутки, трансформация, интервалы
24. и декорация.
25. Базовый синтаксис CSS. Селекторы тегов.
26. Базовый синтаксис CSS. Классы и идентификаторы.
27. Базовый синтаксис CSS. Контекстные, соседние и дочерние селекторы.
28. Базовый синтаксис CSS. Селекторы атрибутов.
29. Блочная модель CSS. Рамки, поля и отступы.
30. Блочная модель CSS. Позиционирование элементов.
31. Блочная модель CSS. Многослойность, выравнивание и обтекание.

Текущий контроль – это непрерывно осуществляемый мониторинг уровня усвоения знаний и формирования умений и навыков в течение семестра. Текущий контроль знаний, умений и навыков студентов осуществляется в ходе учебных (аудиторных) занятий, проводимых по расписанию. Формами текущего контроля выступают используются коллоквиум, выполнение практических

контрольных заданий.

Перечень вопросов коллоквиума

1. Какая разница между $this и self в PHP?
2. Сколько типов данных в PHP?
3. Типизация данных в PHP?
4. Что такое static функция и чем она отличается от "обычной" (не static)?
5. Что такое конструктор?
6. Привести пример конструктора.
7. Обязательно ли писать закрывающий тег ?> в конце скрипта?
8. В каких случаях ?> лучше не использовать?
9. Поддерживает ли PHP множественное наследование?
10. Какая разница между require(), require_once(), include() и include_once()?
11. Какая разница между функциями echo и print в PHP?
12. Чем отличается цикл while от do while?
13. Как перевернуть массив?

Есть массив array('h', 'e', 'l', 'l', 'o'), как из него получить array('o', 'l', 'l', 'e', 'h')?

14. Как перевернуть строку?
15. Что такое рекурсия?
16. Напиши пример рекурсивной функции, которая вычисляет факториал

числа.

17. Как вывести ряд чисел Фибоначчи?
18. Преинкремент и постинкремент. В чем между ними разница?
19. Что работает быстрее: преинкремент или постинкремент?
20. Есть ли разница между одинарными и двойными кавычками в PHP?
21. Нужно определить длину строки функцией strlen(). Откуда она узнает

сколько в строке символов?

22. Проход массива. Как вывести все элементы массива на экран?
23. Что такое ассоциативный массив?
24. Нарисуй форму для отправки файла.
25. Что такое магические методы?
26. Назови по памяти любые функции для работы со строками, массивами и файлами.
27. Какая разница между функциями sort(), asort() и ksort()?
28. Что такое динамические переменные?
29. Как можно переадресовать пользователя на другую страницу?
30. Для чего в PHP 5 используется тип данных "указатель"?
31. Что такое MVC?
32. Что за что отвечает в MVC?
33. Что такое шаблоны (паттерны) проектирования?
34. ООП знаешь? Что это такое?
35. Расскажи основные принципы ООП.
36. Напиши пример реализации полиморфизма.
37. Зачем нужна инкапсуляция?
38. Что такое класс?
39. А что такое объект?
40. Чем отличается класс от объекта?
41. Что такое область видимости переменной?
42. Чем локальные переменные отличаются от глобальных?
43. Чем отличается процедурный подход от объектно-ориентированного?

44. Какие еще есть парадигмы (модели, подходы) программирования кроме ООП?
45. Что такое абстрактный класс?
46. Можно ли создать экземпляр абстрактного класса?
47. Какая разница между абстрактным классом и интерфейсом?
48. Зачем нужен интерфейс, если есть абстрактный класс?
49. Что такое первичный ключ?
50. Что такое нормализация и денормализация?
51. Что такое SQL-инъекция?
52. Чем отличается WHERE от HAVING?
53. Что можешь сказать про команду GROUP BY?
54. Что делает команда EXPLAIN?
55. Как вывести все поля из таблицы my_table?

Перечень практических заданий

1. Дан массив. Выведите его элементы на экран следующим образом:

1 4 7

2 5 8

3 6 9

Массив может быть любой длины. Таблица всегда должна получаться квадратной. Если элементов не хватает - в последнем столбце будет пару пустых ячеек.

Таблица должна сформироваться через тег <table>.

2. Скрипт выводит сам себя.

Напишите PHP скрипт, который будет выводить на экран свой исходный код. То есть: когда мы обращаемся к этому скрипту в браузере - на экран вываливается код этого скрипта.

Код должен выводиться в красиво оформленном виде - как набран в редакторе, так и должен появиться на экране.

Запрещено использовать константу __FILE__.

3. Найти НОК.

Сделайте функцию, которое будет принимать 2 числа, а возвращать их НОК - наименьшее общее кратное. НОК двух чисел - это самое маленькое число, которое делится и на одно, и на второе число. Пример: числа 12 и 15 имеют НОК 60. Число 60 делится и на 12, и на 15 и это самое минимальное такое число.

4. Противоположность array_unique.

Реализуйте функцию-противоположность array_unique. Наша функция должна оставлять элементы, имеющие дубли.

5. Удаление пустых элементов.

Дан массив вида [1, '', 2, '', '', 3] - то есть в нем есть пустые строки. Удалите все такие элементы из этого массива.

6. Разложение числа на простые множители.

Дано число. Разложите его на простые множители. Пример: 12 - это 2 * 2 * 3.

7. Удалить третий пробел.

Дана строка. Удалите из нее третий пробел. Пример: строка '12 34 56 78' станет '12 34 5678'.

8. Подсчет цифр в массиве.

Дан массив с числами. Подсчитайте количество цифр 3 в данном массиве. Пример: [13, 35, 3, 443] - в массиве 4 цифры 3.

9. Добрый array_unique.

Сделайте аналог функции array_unique, который будет удалять дубли, встречающиеся более 3-х раз.

10. Дата масленницы.

Определите дату ближайшей масленницы (последнее воскресенье зимы). Если в этом году она уже была - покажите за следующий год.

11. Случайный цвет.

Сделайте функцию, которая будет генерировать случайный цвет в hex формате (типа #ffffff).

12. Самое длинное слово.

Дана строка с текстом. Выведите на экран самое длинное слово из этого текста (или слова если несколько слов имеют одинаковую длину).

13. Увеличение цены.

Дана строка с текстом. В этом тексте есть цены в долларах - 'текст 10$ текст'. Найдите все цены в этой и увеличьте их на 10%. Верните строку с увеличенными ценами.

14. Количество цифр.

Найдите суммарное количество единиц в числах от 1 до 100.

15. Из textarea в базу.

Дан textarea. В него вводятся имена юзеров, каждый юзер с новой строки. По нажатию на кнопку сохраните введенных юзеров в базу данных - каждого юзера в новую запись в базе. При этом к базе данных должен сделаться только 1 запрос для вставки всех юзеров.

16. Промежуток времени в секунды.

Сделайте функцию, которая будет принимать промежуток времени в формате '1 день', '2 дня', '2 дня и 3 месяца' и тп в количество секунд, соответствующего указанному промежутку.

17. Перестановка.

Дана строка '12345678'. Сделайте из нее '21436587'. Строка, конечно же, может быть любой.

18. Варианты товаров.

Дан массив товаров с ценами. Дана сумма. Выведите все возможные комбинации товаров, которые можно купить на эту сумму.

19. Сумма целых чисел.

Найдите сумму целых чисел от 1 до 10...00 самым незатратным по ресурсам и самым быстро работающим способом.

20. Треугольник Паскаля.

Реализуйте треугольник Паскаля произвольного размера (должен нарисоваться в браузере).

Аттестация раздела по дисциплине проводится в форме тестирования. Тест содержит от 10 вопросов. На выполнение задания отводится от 20 минут. Тест – это форма контроля, направленная на проверку уровня освоения контролируемого теоретического и практического материала по дидактическим единицам дисциплины (терминологический аппарат, основные методы).

Примерный перечень тестовых заданий:

Тестовые задания 1. (Т1)

1. При использовании метода POST, переменные отображаются в URL.

a) Нет

b) Да

2. Выберите способ открытия файла 'sample.txt' для чтения:

a) open('sample.txt');

b) fopen('sample.txt','r+');

c) fopen('sample.txt','r');

d) open('sample.txt','read');

3. Как можно вывести 'Hello World' используя PHP?

a) print ('Hello World');

b) Document.Write('Hello World');

c) echo 'Hello World';

d) 'Hello World';
4. Как декларировать функцию в PHP?
a) function myFunction()
b) create myFunction()
c) new_function myFunction()
5. PHP код может быть выполнен из Microsoft Windows IIS (Internet Information Server):
a) Нет
b) Да
6. PHP позволяет пользователям отправлять e-mail, непосредственно из скрипта:
a) Да
b) Нет
7. Чем заканчивается любое PHP-выражение?
a) ;
b) .
c) </cr>
d) </php>

8. PHP скрипт располагается между:
a) <&>...</&>
b) <?php...?>
c) <?php>...</?>
d) <script>...</script>
9. С какого символа начинаются имена переменных в PHP?
a) #
b) &
c) $
d) !
10. Какой правильный способ увеличить на 1 значение переменной $count?
a) $count++;
b) $count =+1
c) count++;
d) ++count

Тестовые задания 2 (Т2)
1. Функции die() и exit() в PHP делают одно и тоже:
a) Да
b) Нет
2. Выберите способ подключения к базе данных MySQL:
a) mysqli_open(host,username,password,dbname);
b) mysqli_connect(host,username,password,dbname);
c) mysqli_db(host,username,password,dbname);
d) db_open(host,username,password,dbname);
3. Как получаются данные из формы отправленные методом 'get'?
a) Request.QueryString;
b) $_GET[];
c) Request.Form;
d) $_POST[];

4. При создании строк в PHP можно пользоваться как одинарными кавычками ('), так и двойными (").

a) Нет
b) Да

5. Выберите правильный способ загрузки файла 'sample1.inc':

a) <?php include 'samle1.inc'; ?>
b) <!-- include file='samle1.inc' -->
c) <?php include file='samle1.inc'; ?>
d) <?php include: 'samle1.inc'; ?>

6. Какое значение сейчас имеет аббревиатура PHP?

a) Private Home Page
b) Personal Hypertext Processor
c) PHP: Hypertext Preprocessor

7. В какой из переменных используется запрещенное имя?

a) $my-Var
b) $myVar
c) $my_Var
d) $_MyVar

8. Как обозначить комментариев в PHP?:

a) /* ... */
b) *...*
c) <!-- ... -->
d) <comment> ... </comment>

9. Синтаксис PHP является самым близким к:

a) JavaScript
b) Perl и C
c) VBScript

10. Включаемые файлы с INCLUDE должны иметь расширение '.inc':

a) Нет
b) Да

Критерии оценки тестовых заданий:

1. Полнота знаний теоретического контролируемого материала.
2. Количество правильных ответов.

Промежуточная аттестация осуществляется в форме экзамена. Экзамен проводится в форме тестирования. Тест содержит от 15 вопросов. Тест – это форма контроля, направленная на проверку уровня освоения контролируемого теоретического и практического материала по дидактическим единицам дисциплины (терминологический аппарат, основные методы).

Примерный перечень экзаменационных тестовых заданий:

Экзаменационное тестовое задание (Т1).

1. Что выведет такой код:

```
<?php
  $x = 10;
  $x += --$x;
  echo $x;
?>
```

a) 19
b) 18
c) 20
d) 21

2. Каков результат выполнения у данного скрипта:

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

a) Переменная $x = 5
b) Ошибка, поскольку строки с переменными должны быть всегда в одинарных кавычках.
c) Переменная x = 5
d) Переменная 5 = 5

3. Какая функция в PHP позволяет найти вхождение подстроки?

a) strpos()
b) strtok()
c) str_repeat()
d) substr()

4. Параметр session.cookie_lifetime задает длительность жизни cookies в секундах. Какое значение имеет этот параметр по умолчанию:

a) по умолчанию это «10», т.е. данные в cookies считаются правильными ровно 10 секунд
b) по умолчанию это «0», т.е. данные в cookies считаются правильными до закрытия окна браузера
c) по умолчанию это «60»

5. Уничтожить текущую сессию целиком можно командой session_destroy(); К чему приведет уничтожение сессии:

a) после уничтожения сессии уничтожается ее идентификатор, мы больше не можем ни регистрировать переменные, ни вообще производить какие-либо действия с сессией
b) уничтожения сессии очищается массив $_SESSION и больше ничего не происходит
c) после уничтожения сессии массив $_SESSION заполняется нулями

6. С помощью какой функции можно получить идентификатор сессии:

a) id_session
b) session_id
c) session_identificator

7. Какова структура регулярного выражения:

a) общая структура регулярного выражения: шаблон, выделенный с помощью специального символа разделителя, модификатор, влияющий на способ обработки регулярного выражения
b) структура регулярного выражения: шаблон, выделенный с помощью специального символа разделителя
c) общая структура регулярного выражения: шаблон, заключенный в круглые скобки, и функция для его обработки

8. Назначение метасимвола “-“:

a) вычисляет символьный класс
b) задает диапазон символов
c) отрицание класса, но только если это первый символ

9. Каков синтаксис оператора удаления таблицы:

a) REMOVE TABLE [IF EXISTS] имя_таблицы [, имя_таблицы,…] [RESTRICT | CASCADE]
b) DELETE TABLE [IF EXISTS] имя_таблицы [, имя_таблицы,…] [RESTRICT | CASCADE]
c) DROP TABLE [IF EXISTS] имя_таблицы [, имя_таблицы,…] [RESTRICT | CASCADE

10. Составить запрос к таблице описаний статей (Articles): изменить название статьи (title), автор (author) которой «Петров» на название «Второе название»:

a) UPDATE Articles SET title=”Второе название”;

b) UPDATE Articles SET title=”Второе название” WHERE author=”Петров”;

c) UPDATE Articles SET title=”Второе название” WHERE title=” Первое название”;

11. Каковы синтаксис и семантика функции explode:

a) эта функция без параметров. Ее семантика неизвестна

b) массив explode (строка string). Эта функция разбивает строку string на части с помощью разделителя » » и возвращает массив полученных строк

c) массив explode(строка separator, строка string [, int limit]). Эта функция разбивает строку string на части с помощью разделителя separator и возвращает массив полученных строк

12. С помощью какой функции можно проверить, существует ли файл:

a) is_readable()

b) file_exists()

c) is_writable()

13. Что делает функция move_uploaded_file (временное имя файла, место назначения):

a) проверяет, загружен ли файл, и если да, то перемещает его из временной директории в указанную директорию для хранения

b) удаляет загруженный файл

c) перемещает файл из временной директории в указанную директорию для хранения

14. Каким HTTP методом по умолчанию отправляются данные из веб-формы

a) OPTIONS

b) GET

c) PUT

15. Какую функция возвращает метку времени (timestamp):

a) localtime()

b) date()

c) time()

16. Имеется объект ($obj) какого-то класса. Требуется получить список всех свойств класса, которому принадлежит объект.

a) $get_class_vars(get_class->$obj);

b) get_class_var(get_class($obj));

c) $vars = get_class_vars(get_class($obj));

17. Как можно программно узнать имя класса, представителем которого является объект?

a) $(get_class->объект);

b) get_class(объект);

c) class(объект);

18. Имеется следующая форма:

```
<form action=task3.php>
Запись номер 1
<input type=checkbox name=id[] value=10>
<br>
Запись номер 2
<input type=checkbox name=id[] value=20>
<br>Запись номер 3
<input type=checkbox name=id[] value=30>
<br>
<input type=submit value=»»»Отправить»»»>
</form>
```

Изменить значения переданных элементов, увеличив их в 10 раз. Использовать функцию array_walk(). Вывести значения элементов до и после применения array_walk

a) <?print_r($_GET);array_walk($_GET,»»test»»);function test(&$val,$key){foreach ($val as $k=> $v)$val[$k] = $v*10;}print_r($_GET);?>

b) <?print_r($_GET);array_walk($_GET,»»test»»);function test(&$val,$key){$val = $val*10;}print_r($_GET)?>

c) <?print_r($_POST);array_walk($_POST,»»test»»);function test(&$val,$key){foreach ($val as $k=> $v)$val[$k] = $v*10;}print_r($_POST);?>

19. Дана строка «<h2>Наука – двигатель прогресса!</h2>». Как можно вывести ее в браузер именно в таком виде, без форматирования согласно стандарту HTML?

a) echo htmlspecialchars(«»<h2>Наука — двигатель прогресса!</h2>»»);

b) echo htmlentities(«»<h2>Наука – двигатель прогресса!</h2>»»);

c) echo htmlspecialchars(«»<h2>Наука — двигатель прогресса!</h2>»», ENT_COMPAT);

20. Дана строка текста: «PHP – очень простой язык. Я знаю PHP!!!». Заменить в ней все вхождения слова «PHP» на слово «Perl».

a) <?php$str = «»PHP – очень простой язык. Я знаю PHP!!!»»;echo str_replace($str, «»PHP»», «»Perl»»);?>

b) <?php$str = «»PHP – очень простой язык. Я знаю PHP!!!»»;echo str_replace(«»PHP»», «»Perl»», $str);?>

c) <?php$str = «»PHP – очень простой язык. Я знаю PHP!!!»»;echo substr_replace(«»PHP»», «»Perl»», $str);?>

Экзаменационное тестовое задание (Т2).

1. Какие парадигмы программирования поддерживает PHP?

a) логическую
b) процедурную
c) объектно-ориентированную

2. Написать программу, которая выводит «жирными» буквами (тег) строку «Добро пожаловать!» с использованием языка PHP.

a) <?phpecho «» Добро пожаловать! «»;?>
b) <?phpecho «»Добро пожаловать!»»?>
c) <?echo «»<bold>Добро пожаловать!</bold>»»?>

3. Известно, что настройки PHP можно сохранять/изменять не только в php.ini, но и в самих PHP скриптах, в файлах .htaccess и в файле настроек сервера httpd.conf. Где можно установить значение опции session.auto_start?

a) в php.ini или httpd.conf
b) в скрипте пользователя
c) в файле .htaccess

4. Какой оператор обозначает равенство значений в языке PHP?

a) :=
b) =
c) =:
d) ==

5. Как можно задать массив в языке PHP?

a) $arr[«a»,»b»,»c»] = «q»;
b) $arr («0″=> «a»);
c) $arr = array(«a»,»b»,»c»);
d) $arr[0] = «a»;

6. Какие из утверждений относительно оператора require верны?

a) require выполняет код указанного в нем файла только один раз

b) require используется для включения в программу содержимого другого файла

c) при использовании внутри условных блоков require не нужно заключать в фигурные скобки

7. Каким будет результат выполнения следующей программы

<?phpfor ($j=1; ?{ $i = round (9/$j); switch ($i) { case 5: echo «»+»»; break; case 9: echo «»-«»; break; case 3: echo «»!»»; break 2; default: echo $i; break; } $j++;}?>

a) +-!

b) ошибка синтаксиса

c) бесконечный цикл

d) -+!

8. Дана команда: if(!$var) echo «Hello»; В каком случае на экран будет выведено слово «Hello»?

a) если $var == true

b) если $var = 0

c) если $var преобразуется к логическому false

9. Какие из операторов switch записаны правильно с точки зрения синтаксиса?

a) switch ($par){«»1″»: echo «»1″»;»»2″»: echo 2;}

b) switch ($par){case «»1″»: echo «»1″»;case 2: echo 2;default: echo 3;}

c) switch ($par):case «»1″»: echo «»1″»;break;case 2: echo 2;break;endswitch;

10. Файл vars.php:

<?php$a = 1;?>

Файл index.php:

<?phpfor ($I=0; $I < 3; $I++){ include(«»vars.php»»); echo $a++;}?>

a) 12

b) 1234

c) 123

d) 111

11. В каком случае выполняется блок действий цикла

for (expr1; expr2; expr3) {// блок действий}

a) если второе выражение (expr2) вычисляется как true

b) если третье выражение (expr3) вычисляется как true

c) если первое выражение (expr1) вычисляется как true

12. Какое из утверждений относительно оператора include верно?

a) файл, включаемый с помощью include, должен быть правильным HTML-файлом

b) include используется для включения в программу содержимого другого файла

c) файл, включаемый с помощью include, может быть любым файлом

13. Как можно узнать виртуальный путь до выполняющегося в данный момент скрипта?

a) getenv(‘SCRIPT_NAME’)

b) $_SERVER[‘SCRIPT_NAME’]

c) $_SERVER[‘PHP_SELF’]

d) с помощью константы SCRIPT_NAME

14. Чем отличается клиент от сервера?

a) клиент посылает запросы, а сервер обрабатывает их

b) клиент отображает данные на экране компьютера пользователя, а сервер предоставляет данные

c) сервер выполняет запросы клиента, специализируясь на эффективном решении задач определенного класса

d) сервер есть часть компьютерной архитектуры сервер-сервер, а клиент – архитектуры клиент-клиент

e) сервер создает соединение, а клиент, пользуясь им, передает свой запрос

15. Как можно получить адрес страницы, с которой был послан запрос?

a) $_SERVER[‘HTTP_REFERER’]
b) getenv(‘REMOTE_HOST’)
c) getenv(‘HTTP_REFERER’)
d) с помощью константы HTTP_REFERER

16. В чем состоят функции сервера?

a) обрабатывать запросы клиента
b) запускать процессы, запрошенные клиентом, и возвращать клиенту результаты
c) отображать данные на экране компьютера пользователя
d) инициировать соединение с клиентом

17. Дана следующая html форма:

<form name=»»my_form»» method=»»get»»>
Field 1
<input type=text name=»»f1″»>

Select field
<select name=»»f2″»>
<option value=»»1″»>
first
<option value=»»2″»>
second
</select>

<input type=submit>
</form>

Как можно получить переданные клиентом значения из этой формы?

a) $f1 и $f2, если register_globals=off
b) $_POST[«f1»] и $_POST[«f2»]
c) $_GET[«f1»] и $_GET[«f2»]
d) $POST[«f1»] и $POST[«f2»]
e) $_REQUEST[«f1»] и $_REQUEST[«f2»]

18. Создать класс A, расширяющий класс B с помощью метода add() и переменной $a.

a) class B{var $b;function B(){}}class B extends A {var $a;function add($b){$this->a += $b;}}
b) class B{var $b;function B(){}}class A extends B {var $a;function add($b){$this->a += $b;}}
c) class A extends B {var $a;function add($b){$this-> += $b;}}

19. Как можно получить и вывести на экран список всех методов класса, представителем которого является объект?

a) get_methods(get_class(объект));
b) get_class_methods(get_class(объект));
c) get_class_methods(get_class -> объект);

20. Класс MyClass задан следующим образом:

<?phpclass MyClass{var $a;function MyClass() { $this->a = «»hello»»;}function MyFunc($b) { return $b . «» «» . $this->a;}}?>

Создать представителя класса MyClass. Получить и вывести значения всех свойств этого класса. Вызвать методы данного класса.

a) $myObj = new MyClass();echo $myObj->a;$str = $myObj->MyFunc(«»people»»);

Конструктор MyClass() вызывается автоматически при создании представителя класса.

b) new MyClass();echo $MyClass->a;$str = $MyClass->MyFunc(«»people»»);

Конструктор MyClass() вызывается автоматически при создании представителя класса.

c) $myObj = new MyClass(«»test»»);$b = «»a»»;echo $myObj->$b;MyFunc($b);

Шкалы оценки образовательных достижений

Баллы (итоговой рейтинговой оценки)	**Оценка (балл за ответ на экзамене)**	**Требования к знаниям**
90-100	Отлично	Ответы на поставленные вопросы полные, четкие и развернутые. Решения задач логичны, доказательны и демонстрируют аналитические и творческие способности студента.
70-89	Хорошо	Даются полные ответы на поставленные вопросы. Показано умение выделять причинно-следственные связи. При решении задач допущены незначительные ошибки, исправленные с помощью «наводящих» вопросов преподавателя.
60-69	Удовлетво-рительно	Ответы на вопросы и решения поставленных задач недостаточно полные. Логика и последовательность в решении задач имеют нарушения. В ответах отсутствуют выводы.

Учебно-методическое и информационное обеспечение учебной дисциплины

Основная литература:

1. Диков, А. В. Клиентские технологии веб-программирования: JavaScript и DOM : учебное пособие / А. В. Диков. — Санкт-Петербург : Лань, 2020. — 124 с. https://e.lanbook.com/reader/book/126934/#1

Дополнительная литература:

2. Филиппов, С. А. Основы современного веб-программирования : учебное пособие / С. А. Филиппов. — Москва : НИЯУ МИФИ, 2011. — 160 с. https://e.lanbook.com/reader/book/75795/#1"

Программное обеспечение и Интернет-ресурсы:

1. Электронно-библиотечная система «Лань»
2. Электронно-библиотечная система «Юрайт»
3. Российская национальная библиотека - http://www.nlr.ru/
4. Поисковая система - http://www.rambler.ru.
5. Поисковая система - http://www.yandex.ru.
6. Гарант - http://base.garant.ru/.
7. Интернет-Университет Информационных Технологий -http://www.intuit.ru

Для проведения практических занятий и выполнения самостоятельной работы используются учебные компьютерные классы с выходом в Интернет и лицензионным программным обеспечением.

Материально-техническое обеспечение учебной дисциплины

Лекционные занятия проводятся в аудиториях: учебная мебель, учебная доска, комплект мультимедийного оборудования

Практические занятия проводятся в компьютерных классах: учебная мебель, учебная доска, комплект мультимедийного оборудования, персональные компьютеры.

Для самостоятельной работы обучающихся имеется: читальный зал с выходом в сеть Интернет: Учебная мебель, комплект мультимедийного оборудования, персональные компьютеры, МФУ

Учебно-методические рекомендации для студентов

1. Указания для прослушивания лекций

Перед началом занятий внимательно ознакомиться с учебным планом проведения лекций и списком рекомендованной литературы.

Перед посещением очередной лекции освежить в памяти основные концепции пройденного ранее материала. Подготовить при необходимости вопросы преподавателю. Не надо опасаться, что вопросы могут быть простыми.

На лекции основное внимание следует уделять не формулам и математическим выкладкам, а содержанию изучаемых вопросов, определениям и постановкам задач.

В процессе изучения лекционного курса необходимо по возможности часто возвращаться к основным понятиям и методам решения задач (здесь возможен выборочный контроль знаний студентов).

Желательно использовать конспекты лекций, в которых используется принятая преподавателем система обозначений.

Для более подробного изучения курса следует работать с рекомендованными литературными источниками и вновь появляющимися источниками.

2. Указания для участия в практических занятиях

Перед посещением уяснить тему практического занятия и самостоятельно изучить теоретические вопросы.

В конце практического занятия при необходимости выяснить у преподавателя неясные вопросы.

Основные результаты выполнения работы необходимо распечатать.

3. Указания для выполнения самостоятельной работы

Получить у преподавателя задание и список рекомендованной литературы. Изучение теоретических вопросов следует проводить по возможности самостоятельно, но при затруднениях обращаться к преподавателю.

Подготовить письменный отчет о проделанной работе.

При выполнении заданий по усмотрению преподавателя работа может быть оценена без письменного отчета на основе ответов на контрольные вопросы, при условии активной самостоятельной работы.

Подготовить ответы на контрольные вопросы.

Методические рекомендации для преподавателей

1. Указания для проведения лекций

На первой вводной лекции сделать общий обзор содержания курса и отметить новые методы и подходы к решению задач, рассматриваемых в курсе, довести до студентов требования кафедры, ответить на вопросы.

При подготовке к лекционным занятиям необходимо продумать план его проведения, содержание вступительной, основной и заключительной части лекции, ознакомиться с новинками учебной и методической литературы, публикациями периодической печати по теме лекционного занятия. Уточнить план проведения семинарского занятия по теме лекции. Перед изложением текущего лекционного материала напомнить об основных итогах, достигнутых на предыдущих лекциях. С этой целью задать несколько вопросов аудитории и осуществить выборочный контроль знания студентов.

В ходе лекционного занятия преподаватель должен назвать тему, учебные вопросы, ознакомить студентов с перечнем основной и дополнительной литературы по теме занятия. Раскрывая содержание учебных вопросов, акцентировать внимание студентов на основных категориях, явлениях и процессах, особенностях их протекания. Раскрывать сущность и содержание различных точек зрения и научных подходов к объяснению тех или иных явлений и процессов.

Следует аргументировано обосновать собственную позицию по спорным теоретическим вопросам. Приводить примеры. Задавать по ходу изложения лекционного материала риторические вопросы и самому давать на них ответ. Это способствует активизации мыслительной деятельности студентов, повышению их внимания и интереса к материалу лекции, ее содержанию. Преподаватель должен руководить работой студентов по конспектированию лекционного материала, подчеркивать необходимость отражения в конспектах основных положений изучаемой темы, особо выделяя, категориальный аппарат. В заключительной части лекции необходимо сформулировать общие выводы по теме, раскрывающие содержание всех вопросов, поставленных в лекции. Объявить план очередного практического занятия, дать краткие рекомендации по подготовке студентов к семинару. Определить место и время консультации студентам, пожелавшим выступить на занятии с докладами.

На последней лекции уделить время для обзора наиболее важных положений, рассмотренных в курсе.

2. Указания для проведения практических занятий

Четко обозначить тему занятия.

Обсудить основные понятия, связанные с темой.

В процессе решения задач вести дискуссию со студентами о правильности применения теоретических знаний.

Отмечать студентов, наиболее активно участвующих в решении задач и дискуссиях.

В конце занятия задать аудитории несколько контрольных вопросов.

3.Указания по контролю самостоятельной работы студентов

По усмотрению преподавателя задание на самостоятельную работу может быть индивидуальным или фронтальным.

При использовании индивидуальных заданий требовать от студента письменный отчет о проделанной работе.

При применении фронтальных заданий вести коллективные обсуждения со студентами основных теоретических положений.

С целью контроля качества выполнения самостоятельной работы требовать индивидуальные отчеты (допустимо вместо письменного отчета применять индивидуальные контрольные вопросы).

Программа составлена в соответствии с требованиями ОС НИЯУ МИФИ по направлению 09.03.02 Информационные системы и технологии

Рабочую программу составил

ст.пр. кафедры А.Г. Мотков

Рецензент: доцент И.А. Штырова

Программа одобрена на заседании УМКН «Информационные системы и технологии» от 4.07.2023 года, протокол №5.

Председатель учебно-методической комиссии О.В. Виштак