Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(МИ ВлГУ)

Кафедра *ИС*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
21.05.2024

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Web-технологии

Направление подготовки *09.04.02 Информационные системы и технологии*

Профиль подготовки *Системы обработки информации*

Семестр	Трудоемкость, час./зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль, час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
1	**54 / 1,5**	**16**		**32**	**1,6**	**0,25**	**49,85**	**4,15**	**Зач.**
2	**126 / 3,5**	**16**	**14**	**32**	**3,6**	**2,35**	**67,95**	**22,4**	**Экз.(35,65)**
Итого	**180 / 5**	**32**	**14**	**64**	**5,2**	**2,6**	**117,8**	**26,55**	**35,65**

Муром, 2024 г.

1. Цель освоения дисциплины

Цель дисциплины: изучить принципы и основные направлениях развития web-технология

понимать методологию создания информационных систем с использованием средств World Wide Web

знать перспективные направления развития средств и методов обработки данных

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Web-технологии» является необходимым компонентом образования магистров. Содержание курса включает такие вопросы, которые при должном рассмотрении и активном изучении дают ключ к разработке крупных, сложных, высокоавтоматизированных технических систем. В ходе изучения дисциплины учащиеся должны приобрести знания методов, процессов и средств, используемых на практике для достижения главной цели – создания в заданные сроки эффективной системы, отвечающей требованиям заинтересованных лиц. Курс базируется на знаниях, полученных студентами на уровне бакалавриата при изучении следующих дисциплин: информационные технологии, управление данными, моделирование информационных систем, представление знаний в информационных системах, теория информационных процессов и систем, проектирование информационных систем.

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ПК-3 Способен распределять задания по выполнению разработки программного обеспечения, осуществлять общее руководство и контроль выполнения заданий	ПК-3.1 Применяет современные технологии разработки программного обеспечения	(ПК-3.1)	Устный опрос,Вопросы к лабораторным работам,Вопросы промежуточного контроля,Вопросы к лабораторным и практическим работам
	ПК-3.2 Распределяет задания в группе разработчиков и осуществляет общее руководство	Умением разрабатывать новые методы и средства проектирования информационных систем (ПК-3.2) Умением проводить разработку и исследование теоретических и экспериментальных моделей объектов профессиональной деятельности в областях: машиностроение, приборостроение, наука, техника, образование, медицина, административное управление, юриспруденция, бизнес, предпринимательство, коммерция, менеджмент, банковские системы, безопасность информационных систем, управление технологическими процессами, механика, техническая физика,	

		энергетика, ядерная энергетика, силовая электроника, металлургия, строительство, транспорт, железнодорожный транспорт, связь, телекоммуникации, управление инфокоммуникациями, почтовая связь, химическая промышленность, сельское хозяйство, текстильная и легкая промышленность, пищевая промышленность, медицинские и биотехнологии, горное дело, обеспечение безопасности подземных предприятий и производств, геология, нефтегазовая отрасль, геодезия и картография, геоинформационные системы, лесной комплекс, химико-лесной комплекс, экология, сфера сервиса, системы массовой информации, дизайн, медиаиндустрия, а также предприятия различного профиля и все виды деятельности в условиях экономики информационного общества (ПК-3.2)	
	ПК-3.3 Разрабатывает программные продукты в группе и ведет контроль выполнения заданий	(ПК-3.3)	

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зачетных единиц, 180 часов.

4.1. Форма обучения: очная

Уровень базового образования: высшее.

Срок обучения 2г.

4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Введение	1	2							4	Посещаемость лекций
2	Понятие о Web 1.0, 2.0, 3.0	1	2							0,15	Посещаемость лекций
3	Технологии HTML 5	1	4		16						Отчет по лабораторной работе
4	Технология ECMAScript	1	4								Посещаемость лекций
5	Технология JavaScript	1	4		16						Отчет по лабораторной работе
Всего за семестр		54	16		32			1,6	0,25	4,15	Зач.
6	Технология AJAX	2	4		4					3,85	Промежуточная контрольная работа
7	Технология ActionScript	2	4	8	20					2	Отчет по лабораторной работе и практическим занятиям
8	Технологии 3D визуализации в Web	2	4	4	4					2	Отчет по лабораторной работе и практическим занятиям
9	Технология XML	2	4	2	4					14,55	Отчет по лабораторной работе и практическим занятиям
Всего за семестр		126	16	14	32		+	3,6	2,35	22,4	Экз.(35,65)
Итого		180	32	14	64			5,2	2,6	26,55	35,65

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 1

Раздел 1. Введение

Лекция 1.

Введение (2 часа).

Раздел 2. Понятие о Web 1.0, 2.0, 3.0

Лекция 2.

Понятие о Web 1.0, 2.0, 3.0 (2 часа).

Раздел 3. Технологии HTML 5

Лекция 3.

Технологии HTML 5 (2 часа).

Лекция 4.

Технологии HTML 5 (2 часа).

Раздел 4. Технология ECMAScript

Лекция 5.

Технология ECMAScript (2 часа).

Лекция 6.

Технология ECMAScript (2 часа).

Раздел 5. Технология JavaScript

Лекция 7.

Технология JavaScript (2 часа).

Лекция 8.

Технология JavaScript (2 часа).

Семестр 2

Раздел 6. Технология AJAX

Лекция 9.

Технология AJAX (2 часа).

Лекция 10.

Технология AJAX (2 часа).

Раздел 7. Технология ActionScript

Лекция 11.

Технология ActionScript (2 часа).

Лекция 12.

Технология ActionScript (2 часа).

Раздел 8. Технологии 3D визуализации в Web

Лекция 13.

Технологии 3D визуализации в Web (2 часа).

Лекция 14.

Технологии 3D визуализации в Web (2 часа).

Раздел 9. Технология XML

Лекция 15.

Технология XML (2 часа).

Лекция 16.

Технология XML (2 часа).

4.1.2.2. Перечень практических занятий

Семестр 2

Раздел 7. Технология ActionScript

Практическое занятие 1

Реализация секундомера с использованием технологии AJAX (2 часа).

Практическое занятие 2

Выборка информации из БД с использование технологи AJAX (2 часа).

Практическое занятие 3

Динамическое отображение загрузки процессора ПК (2 часа).

Практическое занятие 4

Реализация управляющих действий во flash (2 часа).

Раздел 8. Технологии 3D визуализации в Web

Практическое занятие 5

Реализация перемещения трехмерного объекта (2 часа).

Практическое занятие 6

Реализация свободного вращения трехмерного объекта (2 часа).

Раздел 9. Технология XML

Практическое занятие 7

Разработка страницы просмотра информации XML файла (2 часа).

4.1.2.3. Перечень лабораторных работ

Семестр 1

Раздел 3. Технологии HTML 5

Лабораторная 1.

Ознакомление и апробация новых управляющих конструкций стандарта HTML 5 (4 часа).

Лабораторная 2.

Ознакомление с возможностями стандарта HTML5 по работе с мультимедиа данными (4 часа).

Лабораторная 3.

Ознакомление с возможностями стандарта HTML5 по работе с динамическими данными (4 часа).

Лабораторная 4.

Ознакомление с возможностями стандарта HTML5 по интеграции с корпоративными системами (4 часа).

Раздел 5. Технология JavaScript

Лабораторная 5.

Организация доступа к БД с использованием JavaScript (4 часа).

Лабораторная 6.

Получение сведений о системе средствами JavaScript (4 часа).

Лабораторная 7.

Макетирование форм документов посредством JavaScript (4 часа).

Лабораторная 8.

Разработка динамического пользовательского интерфейса посредством JavaScript (4 часа).

Семестр 2

Раздел 6. Технология AJAX

Лабораторная 9.

Реализация информационного Java-апплета (4 часа).

Раздел 7. Технология ActionScript

Лабораторная 10.

Разработка простейшего HTML-редактора с использованием технологии AJAX (4 часа).

Лабораторная 11.

Реализация погодного информера (4 часа).

Лабораторная 12.

Реализация простейшего flash-калькулятора (4 часа).

Лабораторная 13.

Реализация flash формы выборки данных из БД (4 часа).

Лабораторная 14.

Реализация модели комнаты с динамическим изменением точки обзора (4 часа).

Раздел 8. Технологии 3D визуализации в Web

Лабораторная 15.

Реализация системы экспорта/импорта информации из БД (4 часа).

Раздел 9. Технология XML

Лабораторная 16.

Реализация шаблона документа на XML (4 часа).

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Историческое развитие web-технологий.
2. Перспективы развития web-технологий.
3. Новые управляющие элементы HTML 5. Перспективы применения HTML 5 при разработке информационных и медийных web-ресурсов.
4. ECMAScript – практика применения.
5. Синтаксис JavaScript.
6. Внедрение технологии AJAX в web-ресурс.
7. Создание интерактивных средств с использованием ActionScript.
8. Практика использования 3D визуализации в web.
9. Практика использования XML как средства обмена информацией.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

1. Разработка системы просмотра классификатора адресов России (КЛАДР).
2. Разработка системы «Кулинарная книга».
3. Разработка системы заказа билетов.
4. Разработка системы обмена текстовыми сообщениями.
5. Разработка системы отображения рельефа местности.
6. Разработка системы «Виртуальный выставочный зал».
7. Разработка системы «Сетевой калькулятор».
8. Разработка системы «Навигация».

4.2 Форма обучения: заочная

Уровень базового образования: высшее.

Срок обучения 2г 6м.

Семестр	Трудоемкость, час./ зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль,час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
1	**54 / 1,5**	**6**		**6**	**3**	**0,5**	**15,5**	**34,75**	**Зач.(3,75)**
2	**126 / 3,5**	**6**	**6**	**4**	**3**	**2,35**	**21,35**	**96**	**Экз.(8,65)**
Итого	**180 / 5**	**12**	**6**	**10**	**6**	**2,85**	**36,85**	**130,75**	**12,4**

4.2.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Введение	1	2							30	Посещаемость лекций
2	Понятие о Web 1.0, 2.0, 3.0	1	2							4,75	Посещаемость лекций
3	Технологии HTML 5	1			4					0	Отчет по лабораторной работе
4	Технология ECMAScript	1	2							0	Посещаемость лекций
5	Технология JavaScript	1			2					0	Отчет по лабораторной работе
Всего за семестр		54	6		6	+		3	0,5	34,75	Зач.(3,75)
6	Технология AJAX	2								5,25	Промежуточная контрольная работа
7	Технология ActionScript	2	2	2	4					10	Отчет по лабораторной работе и практическим занятиям
8	Технологии 3D визуализации в	2	2	2						10	Отчет по практическим

	Web										занятиям
9	Технология XML	2	2	2						70,75	Отчет по практическим занятиям
Всего за семестр		126	6	6	4		+	3	2,35	96	Экз.(8,65)
Итого		180	12	6	10			6	2,85	130,75	12,4

4.2.2. Содержание дисциплины

4.2.2.1. Перечень лекций

Семестр 1

Раздел 1. Введение

Лекция 1.

Введение, Понятие о Web 1.0, 2.0, 3.0, 5.0 (2 часа).

Раздел 2. Понятие о Web 1.0, 2.0, 3.0

Лекция 2.

Технология ECMAScript (2 часа).

Раздел 4. Технология ECMAScript

Лекция 3.

Технологии JavaScript, AJAX (2 часа).

Семестр 2

Раздел 7. Технология ActionScript

Лекция 4.

Технология ActionScript (2 часа).

Раздел 8. Технологии 3D визуализации в Web

Лекция 5.

Технологии 3D визуализации в Web (2 часа).

Раздел 9. Технология XML

Лекция 6.

Технология XML (2 часа).

4.2.2.2. Перечень практических занятий

Семестр 2

Раздел 7. Технология ActionScript

Практическое занятие 1.

Выборка информации из БД с использование технологи AJAX (2 часа).

Раздел 8. Технологии 3D визуализации в Web

Практическое занятие 2.

Реализация перемещения трехмерного объекта (2 часа).

Раздел 9. Технология XML

Практическое занятие 3.

Разработка страницы просмотра информации XML файла (2 часа).

4.2.2.3. Перечень лабораторных работ

Семестр 1

Раздел 1. Технологии HTML 5

Лабораторная 1.

Ознакомление и апробация новых управляющих конструкций стандарта HTML 5 (4 часа).

Раздел 2. Технология JavaScript

Лабораторная 2.

Получение сведений о системе средствами JavaScript (2 часа).

Семестр 2
Раздел 3. Технология ActionScript
Лабораторная 3.

Реализация погодного информера (4 часа).

4.2.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Историческое развитие web-технологий.
2. Перспективы развития web-технологий.
3. Новые управляющие элементы HTML 5. Перспективы применения HTML 5 при разработке информационных и медийных web-ресурсов.
4. ECMAScript – практика применения.
5. Синтаксис JavaScript.
6. Внедрение технологии AJAX в web-ресурс.
7. Создание интерактивных средств с использованием ActionScript.
8. Практика использования 3D визуализации в web.
9. Практика использования XML как средства обмена информацией.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.2.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

1. Общее устройство сети интернет.
2. Понятие домена и управление доменами.
3. Протоколы интернет.
4. Выбор технологий web-разработки.
5. Web-приложения и их разновидности.
6. Назначение и логика применения HTML.
7. Структура HTML-документа.
8. Структура HTML-тэга.
9. Основные структурные тэги HTML-документа.
10. Основные оформляющие тэги HTML-документа.
11. Организация взаимосвязи HTML-документов.
12. Логика действия HTML-формы.
13. Понятие стиля и основные стили.
14. Каскадная таблица стилей.
15. Необходимость программирования сервера.
16. Логика действия PHP.
17. Установка и настройка PHP.
18. Синтаксис «встраивания» PHP.
19. Выражения и операции в PHP.
20. Типы данных в PHP.
21. Функции в PHP.
22. Сессии в PHP.
23. Передача и приём параметров в скрипт PHP.
24. Обработка форм с помощью PHP.
25. Структура web-приложения.
26. Авторизация пользователей в web-приложениях.
27. Обмен информацией между модулями в web-приложении.
28. Использование внешних данных в web-приложении.
29. Понятие и назначение языка SQL.
30. Установка MySQL и доступ к базам данных.
31. Использование MySQL в веб-приложении на PHP.
32. Основные виды запросов в MySQL.

33. Динамика пользовательского интерфейса web-приложения.
34. Синтаксис внедрения javascript.
35. Необходимость и логика подключения библиотек javascript.
36. Понятие и общий синтаксис JQuery.
37. Понятие Ajax и общая логика его применения.
38. Общая методика разработки web-сайта.
39. Методика развёртывания web-сайта.
40. Проектная документация при web-разработке.

4.2.2.6. Примерный перечень тем курсовых работ (проектов)

1. Разработка системы просмотра классификатора адресов России (КЛАДР).
2. Разработка системы «Кулинарная книга».
3. Разработка системы заказа билетов.
4. Разработка системы обмена текстовыми сообщениями.
5. Разработка системы отображения рельефа местности.
6. Разработка системы «Виртуальный выставочный зал».
7. Разработка системы «Сетевой калькулятор».
8. Разработка системы «Навигация».

5. Образовательные технологии

- занятия лекционного типа;
- занятия с использованием мультимедийных технологий;
- занятия в виде лабораторных и практических работ;
- оценка промежуточных знаний студентов с использованием специальных программных средств тестирования;
- занятия с привлечением специалистов в области разработки программных средств и информационных технологий.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Сычев, А. В. Web-технологии : учебное пособие / А. В. Сычев. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 407 c. — ISBN 978-5-4497-0292-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/89412.html (дата обращения: 14.09.2023). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/89412

2. Основы web-технологий : учебное пособие / П. Б. Храмцов, С. А. Брик, А. М. Русак, А. И. Сурин. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 374 c. — ISBN 978-5-4497-0673-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/97560.html (дата обращения: 14.09.2023). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/97560

3. Web-технологии : учебное пособие (лабораторный практикум) / составители С. В. Говорова. — Ставрополь : Северо-Кавказский федеральный университет, 2019. — 163 c. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/92671.html (дата обращения: 14.09.2023). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/92671

4. Моргунов, А. В. Web-технологии : учебно-методическое пособие / А. В. Моргунов. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2022. — 101 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/126668.html (дата обращения: 14.09.2023). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/126668

5. Рындин, Н. А. Технологии разработки клиентских WEB-приложений на языке JavaScript : учебное пособие / Н. А. Рындин. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 54 с. — ISBN 978-5-7731-0888-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/108188.html (дата обращения: 14.09.2023). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/108188

6. Никитченко, И. И. Основы web-технологий : учебное пособие / И. И. Никитченко, К. Н. Мезенцев, О. В. Зинюк. — Москва : Российская таможенная академия, 2020. — 140 с. — ISBN 978-5-9590-1126-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/105689.html (дата обращения: 14.09.2023). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/105689

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Серова, Е. А. Использование web-технологий при создании информационных систем : учебно-методическое пособие / Е. А. Серова, Л. А. Шилова, В. С. Евстратов. — Москва : МИСИ-МГСУ, ЭБС АСВ, 2020. — 55 с. — ISBN 978-5-7264-2203-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/101866.html (дата обращения: 14.09.2023). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/101866

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:
http://php.net/manual/ru/ - PHP
http://htmlbook.ru/ - HTML, CSS
http://jquery.com/ - jQuery
http://jquery.page2page.ru – русскоязычная wiki о jQuery
http://javascript.ru/ - JavaScript
Программное обеспечение:
LibreOffice (Mozilla Public License v2.0)
Foxit Reader (Foxit EULA)
Double Commander (GNU GPL 2+)

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru
php.net
htmlbook.ru

jquery.com
jquery.page2page.ru – русскоязычная wiki о jQuery
javascript.ru
mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория интерфейсов, телекоммуникационных технологий и сетей

1 мультимедийный микрокомпьютер 3Q; стенд лабораторный «Телекоммуникационные линии связи» ТЛС-02; генератор сигналов специальной формы АКИП-3407/4А; осциллограф GOS-652G; стенд учебно-лабораторный «Локальные компьютерные сети» LAN-1; стенд учебно-лабораторный «Интерфейсы периферийных устройств» IPU; интерактивная доска SMART Board 480 со встроенным проектором V25; проектор Benq; экран настенный Lumien Master Picture.

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

На практических занятиях пройденный теоретический материал подкрепляется решением задач по основным темам дисциплины. Занятия проводятся в компьютерном классе, используя специальное программное обеспечение. Каждой подгруппе обучающихся преподаватель выдает задачу, связанную с разработкой и программной реализацией алгоритмов обработки информации. В конце занятия обучающие демонстрируют полученные результаты преподавателю и при необходимости делают работу над ошибками.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводятся в компьютерном классе. Обучающиеся выполняют индивидуальную задачу компьютерного моделирования в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

Курсовая работа выполняется в соответствии с методическими указаниями на курсовую работу. Обучающийся выбирает одну из указанных в перечне тем курсовых работ, исходя из своих интересов, наличия соответствующих литературных и иных источников. В ходе выполнения курсовой работы преподаватель проводит консультации обучающегося. На заключительном этапе обучающийся оформляет пояснительную записку к курсовой работе и выполняет ее защиту в присутствии комиссии из преподавателей кафедры.

Форма заключительного контроля при промежуточной аттестации – экзамен. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.04.02 Информационные системы и технологии* и профилю подготовки *Системы обработки информации*
Рабочую программу составил *д.т.н., зав.каф. Андрианов Д.Е.*_____

Программа рассмотрена и одобрена на заседании кафедры *ИС*

протокол № 18 от 07.05.2024 года.
Заведующий кафедрой *ИС* _____*Андрианов Д.Е.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 17.05.2024 года.
Председатель комиссии ФИТР _____ *Рыжкова М.Н.*
(Подпись) (Ф.И.О.)

Приложение

Фонд оценочных материалов (средств) по дисциплине
Web-технологии

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

Достоинства и недостатки существующих технологий.
Особенности разработки Web-проектов.
Создание Web – страниц при помощи HTML.
Правила оформления HTML документов.
Создание HTML документов.
Абсолютный и относительный URL.
Гипертекстовые ссылки.
Наследование свойств в HTML.
JavaScript и объектная модель.
Объекты Window, Document, Location, History.
Динамический HTML.
Стили.
Введение в расширяемый язык разметки XML.
Описание языка XML.
Иерархическая структура XML документа.
Стилевые таблицы XSL.
CGI – интерфейс.
Структура HTTP запросов и ответов в CGI.
Базы данных и виды доступа.
Некоторые аспекты безопасности и серверные вставки.
ISAPI – расширения.
Межплатформенные Интернет-технологии.
Приложения ASP. NET.
. NET Framework.
Объекты и языки . NET Framework.
Типы данных в ASP. NET.
Web – формы и компоненты в ASP. NET.
Что такое HTML? Как создаются HTML документы? Структура HTML документа.
Что такое тэг? Тэги заголовков, абзаца, преформатирования.
Тэги списков. Гипертекстовые ссылки.
Тэги стилей, управления шрифтом и цветом, линии.
Таблицы в HTML. Фреймы и формы.
Графические изображения и гипертекстовые ссылки на его части.
Особенности создания сценариев JavaScript. Структура. Объявление переменных.
Операторы в JavaScript.
Числа и операции в JavaScript. Логические данные.
Создание каскадных таблиц стилей
Реализация скриптов на PHP.
Доступ к БД по средствам PHP.

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	Контрольная работа, отчеты по лабораторным работам	15
Рейтинг-контроль 2	Контрольная работа, отчеты по лабораторным работам	15
Рейтинг-контроль 3	Контрольная работа, отчеты по лабораторным работам	15

Посещение занятий студентом	Журнал посещения	5
Дополнительные баллы (бонусы)	Устный опрос	5
Выполнение семестрового плана самостоятельной работы	Устный опрос	5

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой.
Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

Достоинства и недостатки существующих технологий.
Особенности разработки Web-проектов.
Создание Web – страниц при помощи HTML.
Правила оформления HTML документов.
Создание HTML документов.
Абсолютный и относительный URL.
Гипертекстовые ссылки.
Наследование свойств в HTML.
JavaScript и объектная модель.
Объекты Window, Document, Location, History.
Динамический HTML.
Стили.
Введение в расширяемый язык разметки XML.
Описание языка XML.
Иерархическая структура XML документа.
Стилевые таблицы XSL.
CGI – интерфейс.
Структура HTTP запросов и ответов в CGI.
Базы данных и виды доступа.
Некоторые аспекты безопасности и серверные вставки.
ISAPI – расширения.
Межплатформенные Интернет-технологии.
Приложения ASP. NET.
. NET Framework.
Объекты и языки . NET Framework.
Типы данных в ASP. NET.
Web – формы и компоненты в ASP. NET.
Что такое HTML? Как создаются HTML документы? Структура HTML документа.
Что такое тэг? Тэги заголовков, абзаца, преформатирования.
Тэги списков. Гипертекстовые ссылки.
Тэги стилей, управления шрифтом и цветом, линии.
Таблицы в HTML. Фреймы и формы.
Графические изображения и гипертекстовые ссылки на его части.
Особенности создания сценариев JavaScript. Структура. Объявление переменных.
Операторы в JavaScript.
Числа и операции в JavaScript. Логические данные.
Создание каскадных таблиц стилей
Реализация скриптов на PHP.
Доступ к БД по средствам PHP.

Методические материалы, характеризующие процедуры оценивания

https://www.mivlgu.ru/iop/course/view.php?id=3756

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	***Уровень сформированности компетенций***
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	***Высокий уровень***
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	***Продвинутый уровень***
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	***Пороговый уровень***
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	***Компетенции не сформированы***

3. Задания в тестовой форме по дисциплине

Примеры заданий:
ПК-2
Блок 1 (знать)
Какой будет результат выполнения программного кода

```
$var1 = "World";
$hello = "var1";
echo $$hello;
```

- var1
- World
- hello
- hello World

Укажите правильный селектор для всех элементов с классом «tab»

```
<ul>
   <li class="feedtab news active">
      <a href="/news/">
         <b class="tab word">Новости</b>
      </a>
   </li>
   <li class="feedtab notifications">
      <a href="/notifications/">
         <b class="tab word">Ответы</b>
      </a>
   </li>
   <li class="feedtab comments">
      <a href="/comments/">
         <b class="tab word">Комментарии</b>
      </a>
   </li>
</ul>
```

- .tab .word
- b .tab.word
- .feedtab .tab.word
- .feedtab > .tab .word

Сколько записей будет иметь объект jQuery после выполнения следующей функции

```
$(".feedtab.active").parents().find(".tab")
```

HTML

```
<ul>
   <li class="feedtab news active">
      <a href="/news/">
         <b class="tab word">Новости</b>
      </a>
   </li>
   <li class="feedtab notifications">
      <a href="/notifications/">
         <b class="tab word">Ответы</b>
```

```
        </a>
    </li>
    <li class="feedtab comments">
        <a href="/comments/">
            <b class="tab word">Комментарии</b>
        </a>
    </li>
</ul>
```

•
0
•
1
•
2
•
3

Что произойдёт после нажатия на кнопку Find

```
$(function() {
    function Form(selector) {
        var self = this;
        this.selector = selector;
        this.field;

        this.setField = function(field) {
            this.field = field;
            return this;
        }

        this.onSubmit = function(){
            if (self.field) {
                alert(self.field.attr('name'));
            }
        };

        this.selector.submit(function() {
            self.onSubmit.call(self);
            return false;
        })
    }

    var search = new Form($("form"));
    search.setField(search.selector.find(">input"));
})
```

HTML

```
<form>
    <input type="hidden" name="journal" />

    <fieldset>
        <input type="text" name="query" placeholder="Search" />
    </fieldset>
```

```
    <fieldset>
        <button type="submit">
            <span>Find</span>
        </button>
    </fieldset>
</form>
```

•
Ничего
•
Появится сообщение с текстом «journal»
•
Появится сообщение с текстом «query»
•
Появится сообщение с текстом «journal, query»

Какой будет результат выполнения программного кода

```
abstract class Test {
    public $a;
    public $b;
    public $result;

    public function __construct($a, $b) {
        $this->a = $a;
        $this->b = $b;
    }

    abstract function calc();

    final public function getResult() {
        return $this->result;
    }
}
class Test1 extends Test {
    public function calc() {
        $this->result = $this->a + $this->b;
    }

}
class Test2 extends Test1 {
    public function calc() {
        parent::calc();
        $this->result *= 2;
    }
}

$obj = new Test1(2,3);
$obj->calc();
echo $obj->getResult();
```

•
Ошибка выполнения
•
5
•
10

•
0
Выберите правильное определение функции внутри основного кода:
•
var func = function();
•
function test(){}
•
func: function(){};
•
function(){}
Какие стили будут добавлены к элементу input после компиляции LESS файла
```
.padding(@value: 0) {
   padding: ~"@{value}px";
}

.calc(@a; @b; @c) {
   .padding(@a + @b);
}

.calc(@a; @b; @c) when (@c = "+") {
   .padding(@a + @b);
}

.calc(@a; @b; @c) when (@c = "-") {
   .padding(@a - @b);
}

.calc(@a; @b; @c) when (@c = "/") {
   .padding(@a / @b);
}

.calc(@a; @b; @c) when (@c = "*") {
   .padding(@a * @b);
}

input {
   .calc(20, 5, "/");
}
```
•
padding: 20px;
•
padding: 4px;
•
padding: 25px;
•
padding: 0px;
Выберите правильный формат инициализации переменных
•
var test = 2;
•
test := 2;
•
$test = 2;

•
var $test := 2;
Какие стили будут добавлены к элементу input после компиляции LESS файла
```
.padding(@vert: 0px; @hor: 0px) {
   padding: @vert @hor;
}

.calc(@a; @b: 5) {
   .padding(~"@{a}px"; ~"@{b}px");
}

input {
   .calc(10);
}
```
•
.calc(10)
•
padding: 10
•
padding: 10px 0px
•
padding: 10px 5px

ПК-2, ПК-8
Блок 2, уметь
Вопрос 10
Какой будет результат выполнения программного кода

```
class TestParent {
   public $result;

   public static function calc($a, $b) {
      return $a + $b;
   }

   public function getResult() {
      return $this->result;
   }
}
class Test extends TestParent {
   public function __construct($a, $b) {
      $this->result = parent::calc($a, $b);
   }

   public static function calc($a, $b) {
      $this->result = $a * $b;
      return $this->result;
   }
}

$obj = new Test(2,3);
echo $obj->getResult();
```
•
Ошибка выполнения

•
5
•
23
•
6
Выберите результат, который вернёт функция, после выполнения следующего кода

```
function Test(param1, param2) {
   this.result = param1 + param2;

   this.getResult = function() {
      return this.result;
   }
}

var test1 = new Test(10, 2);
test1.getResult()
```

•
NaN
•
102
•
12
•
0
Выберите корректный альтернативный селектор, описывающий строку с классом «.level1»

```
.block.pricelist .table > .header .column {
  font-weight: bold;
}
.block.pricelist .table > .content .row.level1 .column:first-child {
  padding-left: 15px;
}
.block.pricelist .table > .content .row.level2 .column:first-child {
  padding-left: 30px;
}
```

•
.block .table.content .level1
•
.block .table > .row.level1
•
.block > .content > .level1
•
.block .table .row.level1
Выберите правильный вариант подключения интерфейса к классу
•
class Test include iTest
•
class Test implements iTest()
•
class Test implements iTest
•
class Test interface iTest

Выберите правильный вариант подключения нескольких интерфейсов к классу

- class Test include iTest, iTest2
- class Test implements iTest() implements iTest2
- class Test implements iTest, iTest2
- class Test interface iTest and iTest2

Выберите корректный альтернативный селектор, описывающий только колонку внутри заголовка таблицы

```
.block.pricelist .table > .header .column {
  font-weight: bold;
}
.block.pricelist .table > .content .row.level1 .column:first-child {
  padding-left: 15px;
}
.block.pricelist .table > .content .row.level2 .column:first-child {
  padding-left: 30px;
}
```

- .table > .column
- .block.table .column
- .table .column
- .block .table .header .column

Какой будет результат выполнения программного кода

```
class TestParent {
    public $result;

    public static function calc($a, $b) {
        return $a + $b;
    }

    public function getResult() {
        return $this->result * 2;
    }
}
class Test extends TestParent {
    public function __construct($a, $b) {
        $this->result = self::calc($a, $b);
    }

    public static function calc($a, $b) {
        return $a * $b;
    }

    public function getResult() {
        return $this->result;
    }
}
```

```
$obj = new Test(2,3);
echo $obj->getResult();
```

•

Ошибка выполнения

•

5

•

12

•

6

Выберите результат, который вернёт функция, после выполнения следующего кода

```
function Test(param1, param2) {
   this.result = param1 + param2;

   this.getResult = function() {
      return this.result;
   }
}

var test1 = new Test(10);
test1.getResult()
```

•

NaN

•

10

•

undefined

•

0

Выберите корректный альтернативный селектор, описывающий все колонки внутри таблицы

```
.block.pricelist .table > .header .column {
  font-weight: bold;
}
.block.pricelist .table > .content .row.level1 .column:first-child {
  padding-left: 15px;
}
.block.pricelist .table > .content .row.level2 .column:first-child {
  padding-left: 30px;
}
```

•

.table .column

•

.header .column

•

.table.column

•

.table > .column

Выберите правильный вариант объявления публичного метода класса

•

```
public function test() {}
```

•

class function test() {}
•
method test() {}
•
public method test() {}
Выберите результат, который вернёт функция, после выполнения следующего кода

```
function Test(param1, param2) {
   this.result = param1 + param2;

   this.getResult = function() {
      return this.result;
   }
}

var test1 = new Test("test", 2);
test1.getResult()
```

•
NaN
•
2
•
undefined
•
test2
Какой будет результат выполнения программного кода

```
class TestParent {
   public $result;

   public static function calc($a, $b) {
      return $a + $b;
   }

   public function getResult() {
      return $this->result * 2;
   }
}
class Test extends TestParent {
   public function __construct($a, $b) {
      $this->result = parent::calc($a, $b);
   }

   public static function calc($a, $b) {
      return $a * $b;
   }

   public function getResult() {
      return $this->result;
   }
}

$obj = new Test(2,3);
echo $obj->getResult();
```

•
Ошибка выполнения
•
5
•
10
•
6
Выберите результат, который вернёт функция, после выполнения следующего кода

```
function Test(param1, param2) {
   this.result = param1 + param2;

   this.getResult = function() {
      return this.result;
   }
}

var test1 = new Test(2, "test");
test1.getResult()
```

•
NaN
•
2
•
undefined
•
2test
Выберите правильный вариант объявления абстрактного класса
•
abstract public function test();
•
abstract public function test() {}
•
abstract method test();
•
abstract method test;
Выберите правильный вариант доступа к не статичным переменным класса из не статичной функции
•
$this->$param1;
•
$this::$param1;
•
$this->param1;
•
$this::param1;
Выберите результат, который вернёт функция, после выполнения следующего кода

```
function Test(param1, param2) {
   this.result = param1 + param2;

   this.getResult = function() {
      return this.result;
```

```
    }
}

var test1 = new Test("test", "test");
test1.getResult()
```

- NaN
- 0
- testtest
- undefined

Выберите результат, который вернёт функция, после выполнения следующего кода

```
function Test(param1, param2) {
    this.result = param1 + param2;

    this.getResult = function() {
        return this.result;
    }
}

var test1 = new Test("test", [1,2]);
test1.getResult()
```

- NaN
- 0
- test1,2
- undefined

Выберите правильный вариант доступа к статичным переменным класса из статичной функции

- $this::$param1;
- $this::param1;
- self::$param1;
- self::param1;

Выберите результат, который вернёт функция, после выполнения следующего кода

```
function Test(param1, param2) {
    this.result = param1 + param2;

    this.getResult = function() {
        return this.result;
    }
}

var test1 = new Test([1,2], [3,4]);
```

test1.getResult()

- NaN
- [1,2,3,4]
- []
- 1,23,4

Выберите правильный вариант вызова функции от родительского объекта

- parent::test();
- self::test();
- child::test();
- $this->test();

Выберите правильный вариант вызова не статичной, публичной функции от созданного объекта

- $obj->test();
- $obj::test();
- $obj->$test();
- $obj::$test();

Выберите результат, который вернёт функция, после выполнения следующего кода

```
function Test(param1, param2) {
   this.result = param1 + param2;

   this.getResult = function() {
      return this.result;
   }
}

var test1 = new Test({abc: 1}, {cde: 2});
test1.getResult()
```

- NaN
- {abc:1, cde:2}
- {}
- [object Object][object Object]

Выберите правильный вариант вызова статичной, публичной функции от созданного объекта

- $obj->test();
- $obj::test();

•
$obj->$test();
•
$obj::$test();
Выберите результат, который вернёт функция, после выполнения следующего кода

```
function Test(param1, param2) {
   this.result = param1 + param2;

   this.getResult = function() {
      return this.result;
   }
}

var test1 = new Test(2, "2");
test1.getResult()
```

•
NaN
•
2
•
22
•
4
Сколько записей будет иметь объект jQuery после выполнения следующей функции

```
$("ul a:lt(2)")
```

HTML

```
<ul>
   <li class="feedtab news active">
      <a href="/news/">
         <b class="tab word">Новости</b>
      </a>
   </li>
   <li class="feedtab notifications">
      <a href="/notifications/">
         <b class="tab word">Ответы</b>
      </a>
   </li>
   <li class="feedtab comments">
      <a href="/comments/">
         <b class="tab word">Комментарии</b>
      </a>
   </li>
</ul>
```

•
0
•
1
•
2
•

3
Что произойдёт после нажатия на кнопку Find

```
$(function() {
   function Form(selector) {
      var self = this;
      this.selector = selector;
      this.field;

      this.setField = function(field) {
         this.field = field;
         return this;
      }

      this.onSubmit = function(){
         if (self.field) {
            alert(self.field.attr('name'));
         }
      };

      this.selector.submit(function() {
         self.onSubmit.call(self);
         return false;
      })
   }

   var search = new Form($("form"));
   search.setField(search.selector.find(">input"));
   search.onSubmit = function() {

   }
})
```

HTML

```
<form>
   <input type="hidden" name="journal" />

   <fieldset>
      <input type="text" name="query" placeholder="Search" />
   </fieldset>

   <fieldset>
      <button type="submit">
         <span>Find</span>
      </button>
   </fieldset>
</form>
```

- Ничего
- Появится сообщение с текстом «journal»
- Появится сообщение с текстом «query»

•
Появится сообщение с текстом «journal, query»
Укажите правильный селектор для активного элемента

```
<ul>
   <li class="feedtab news active">
      <a href="/news/">
         <b class="tab word">Новости</b>
      </a>
   </li>
   <li class="feedtab notifications">
      <a href="/notifications/">
         <b class="tab word">Ответы</b>
      </a>
   </li>
   <li class="feedtab comments">
      <a href="/comments/">
         <b class="tab word">Комментарии</b>
      </a>
   </li>
</ul>
```

•
.ul li.news.active
•
ul > .active
•
li .feedtab.active
•
.li.active
Какой будет результат выполнения программного кода

```
$var1 = "hello";
$hello = "var1";
$hello_world = "Hello world";
echo ${$$hello."_world"};
```

•
var1
•
World
•
hello
•
hello World

ПК-8
Блок 2 (уметь)
Какие стили будут добавлены к элементу input после компиляции LESS файла

```
.padding(@vert: 0px; @hor: 0px) {
   padding: @vert @hor;
}

.calc(@a; @b: 5) {
   .padding(~"@{a}px"; 10px);
}
```

```
input {
   .calc(10; 15);
}
```
•
padding: 10px 15px
•
padding: 10 15
•
padding: 10px 10px
•
padding: 10px 5px

Какие стили будут добавлены к элементу input после компиляции LESS файла

```
.padding(@value: 0) {
   padding: ~"@{value}px";
}

.calc(@a; @b; @c) {
   .padding(@a + @b);
}

.calc(@a; @b; @c) when (@c = "+") {
   .padding(@a + @b);
}

.calc(@a; @b; @c) when (@c = "-") {
   .padding(@a - @b);
}

.calc(@a; @b; @c) when (@c = "/") {
   .padding(@a / @b);
}

.calc(@a; @b; @c) when (@c = "*") {
   .padding(@a * @b);
}

input {
   .calc(20, 5, "%");
}
```
•
padding: 25px;
•
padding: 4px;
•
Стили не будут добавлены
•
padding: 0px;

Выберите неправильный вариант оформления цикла FOR
•
for(var $i = 1; $i < 10; $i++);
•
for($j = 1; $j < 10; $j++);
•
for(;;);

•
for($i = 1, $j = 1;;$i+=5);
Выберите неправильное определение цикла FOR
•
for(var i in values);
•
for(var i=0;;i++);
•
for(var i in key=>value);
•
for(var i=0, j=0; i<10, j<10; i++, j++);
Какой будет результат выполнения программного кода

```
abstract class Test {
   public $a;
   public $b;
   public $result;

   public function __construct($a, $b) {
      $this->a = $a;
      $this->b = $b;
   }

   abstract function calc();

   final public function getResult() {
      return $this->result;
   }
}
class Test1 extends Test {
   public function calc() {
      $this->result = $this->a + $this->b;
   }

}
class Test2 extends Test1 {
   public function calc() {
      parent::calc();
      $this->result *= 2;
   }
}

$obj = new Test2(2,3);
$obj->calc();
echo $obj->getResult();
```

•
Ошибка выполнения
•
5
•
10
•
0
Выберите не правильное определение цикла WHILE

•
while(true);
•
while(i < 20);
•
while(i in [1,2,3]);
•
while;
Какие стили будут добавлены к элементу input после компиляции LESS файла
```
.padding(@vert: 0px; @hor: 0px) {
   padding: @vert @hor;
}

.calc(@a; @b: 5) {
   .padding(~"`@{a}+@{b}`px");
}

input {
   .calc(10; 15);
}
```
•
padding: 10px 15px
•
padding: 10px 5px
•
padding: 15px 0px
•
padding: 25px 0px
Какой будет результат выполнения программного кода

```
function test($a, $b) {
   return $a + $b;
}
$func = "test";
$args = array(2, 3);
echo call_user_func_array($func, $args);
```
•
Ошибка выполнения
•
5
•
23
•
test23
Укажите правильный селектор для ссылки с адресом /news/
```
<ul>
   <li class="feedtab news active">
      <a href="/news/">
         <b class="tab word">Новости</b>
      </a>
   </li>
   <li class="feedtab notifications">
      <a href="/notifications/">
         <b class="tab word">Ответы</b>
```

```
        </a>
    </li>
    <li class="feedtab comments">
        <a href="/comments/">
            <b class="tab word">Комментарии</b>
        </a>
    </li>
</ul>
```

•
.feedtab[href=’/news/’]
•
li a [href=’/news/’]
•
ul a[href=’/news/’]
•
ul > [href=’/news/’]

Какие стили будут добавлены к элементу input после компиляции LESS файла

```
.padding(@value: 0) {
    padding: ~"@{value}px";
}

.calc(@a; @b; @c) when (@c = "+") {
    .padding(@a + @b);
}

.calc(@a; @b; @c) when (@c = "-") {
    .padding(@a - @b);
}

.calc(@a; @b; @c) when (@c = "/") {
    .padding(@a / @b);
}

.calc(@a; @b; @c) when (@c = "*") {
    .padding(@a * @b);
}

.calc(@a; @b; @c) {
    .padding(@a + @b);
}

input {
    .calc(20, 5, "-");
}
```

•
padding: 15px;
•
padding: 25px;
•
padding: 20px;
•
padding: 0px;

Выберите неправильный вариант оформления цикла WHILE

•

while();
•
while(1);
•
while($i < 10);
•
while($i < test());
Сколько записей будет иметь объект jQuery после выполнения следующей функции

```
$("li.news a[href='/comments/'] b").parents()
```

HTML

```
<ul>
   <li class="feedtab news active">
      <a href="/news/">
         <b class="tab word">Новости</b>
      </a>
   </li>
   <li class="feedtab notifications">
      <a href="/notifications/">
         <b class="tab word">Ответы</b>
      </a>
   </li>
   <li class="feedtab comments">
      <a href="/comments/">
         <b class="tab word">Комментарии</b>
      </a>
   </li>
</ul>
```

•
0
•
1
•
2
•
3
Что произойдёт после нажатия на кнопку Find

```
$(function() {
   function Form(selector) {
      var self = this;
      this.selector = selector;
      this.field;

      this.setField = function(field) {
         this.field = field;
         return this;
      }

      this.onSubmit = function(){
         if (self.field) {
            alert(self.field.attr('name'));
```

```
        }
    };

    this.selector.submit(function() {
        self.onSubmit.call(self);
        return false;
    })
}

var search = new Form($("form"));
search.setField(search.selector.find(">input"));
search.onSubmit = function() {
    if (this.field) {
        alert(this.field.attr('type'));
    }
}})
```

HTML

```
<form>
    <input type="hidden" name="journal" />

    <fieldset>
        <input type="text" name="query" placeholder="Search" />
    </fieldset>

    <fieldset>
        <button type="submit">
            <span>Find</span>
        </button>
    </fieldset>
</form>
```

- Ничего
- Появится сообщение с текстом «journal»
- Появится сообщение с текстом «query»
- Появится сообщение с текстом «hidden»

Какой будет результат выполнения программного кода

```
abstract class Test {
    public $a;
    public $b;
    public $result;

    public function __construct($a, $b) {
        $this->a = $a;
        $this->b = $b;
    }

    abstract function calc();
```

```
    final public function getResult() {
        return $this->result;
    }
}
class Test1 extends Test {
    public function calc() {
        $this->result = $this->a + $this->b;
    }

}
class Test2 extends Test1 {
    public function calc() {
        parent::calc();
        $this->result *= 2;
    }
}
class Test3 extends Test {

}

$obj = new Test3(2,3);
$obj->calc();
echo $obj->getResult();
```

•
Ошибка выполнения
•
5
•
10
•
0

Какой будет результат выполнения программного кода

```
function test($a, $b) {
    return $a + $b;
}
$func = "test";
$args = array(2, 3);
echo call_user_func_array($$func, $args);
```

•
Ошибка выполнения
•
5
•
23
•
test23

Какие классы будут доступны в HTML после компиляции LESS файла

```
.percent (@value) when (@value >= 0) {
    .w@{value} {
        width: ~"{@value}%";
    }
    .h@{value} {
        height: ~"{@value}%";
```

```
    }
    .percent(@value - 5);
}
.percent (0) {}
.percent (100);
```

- классы «w» и «h»
- классы «w0», «w100», «h0», «h100»
- Классы не будут доступны в HTML
- Классы от «w0» до «w100» с интервалом 5, и классы от «h0» до «h100» с интервалом 5

Что будет показано на экране после выполнения следующего кода

```
var type = "test3";
switch(type) {
    case "test1":
        alert("test1");
    case "test2":
        alert("test2");
        break;
    case "test3":
        alert("test3");
    default:
        alert("default");
        break;
}
```

- test3
- test1, test2, test3
- test3, default
- default

Какой будет результат выполнения программного кода

```
abstract class Test {
    public $a;
    public $b;
    public $result;

    public function __construct($a, $b) {
        $this->a = $a;
        $this->b = $b;
    }

    abstract function calc();

    final public function getResult() {
        return $this->result;
    }
}
class Test1 extends Test {
```

```
    public function calc() {
        $this->result = $this->a + $this->b;
    }

}
class Test2 extends Test1 {
    public function calc() {
        parent::calc();
        $this->result *= 2;
    }

    public function getResult() {
        return $this->result;
    }
}

$obj = new Test2(2,3);
$obj->calc();
echo $obj->getResult();
```

•
Ошибка выполнения
•
5
•
10
•
0

Сколько записей будет иметь объект jQuery после выполнения следующей функции

```
$(".word:lt(2)").parents("li").find("b")
```

HTML

```
<ul>
    <li class="feedtab news active">
        <a href="/news/">
            <b class="tab word">Новости</b>
        </a>
    </li>
    <li class="feedtab notifications">
        <a href="/notifications/">
            <b class="tab word">Ответы</b>
        </a>
    </li>
    <li class="feedtab comments">
        <a href="/comments/">
            <b class="tab word">Комментарии</b>
        </a>
    </li>
</ul>
```

•
0
•
1

•
2
•
3
Выберите неправильный вариант оформления цикла FOREACH
•
foreach(array(1,2,3) as $value);
•
foreach(array(1,2,3));
•
foreach(array(1,2,3) as $value => $value);
•
foreach(array(1,2,3) as &$value);
Какие стили будут добавлены к элементу input после компиляции LESS файла
@defaultFontSize: 14px;

.padding(@value: 0px) {
  padding: @value;
}

.margin(@value: 0px) {
  margin: @value;
}

.font-size(@value: @defaultFontSize) {
  font-size: @value;
}

.search(@value: 0px) {
  .padding(@value);
  .margin(@value / 2);
  .font-size(@defaultFontSize * 1.5);
}

input {
  .search(10px);
}
•
.search(10px)
•
padding: 0px
margin: 0px
font-size: 14px;
•
padding: 10px
margin: 10px
font-size: 21px;
•
padding: 10px
margin: 5px
font-size: 21px;
Что произойдёт после нажатия на кнопку Find

$(function() {

```
function Form(selector) {
    var self = this;
    this.selector = selector;
    this.field;

    this.setField = function(field) {
        this.field = field;
        return this;
    }

    this.onSubmit = function(){
        if (self.field) {
            alert(self.field.attr('name'));
        }
    };

    this.selector.submit(function() {
        self.onSubmit.call(self);
        return false;
    })
}

var search = new Form($("form"));
search.setField(search.selector.find("input:last"));
search.onSubmit = function() {
    if (this.field) {
        alert(this.field.attr('type'));
    }
}})
```

HTML

```
<form>
    <input type="hidden" name="journal" />

    <fieldset>
        <input type="text" name="query" placeholder="Search" />
    </fieldset>

    <fieldset>
        <button type="submit">
            <span>Find</span>
        </button>
    </fieldset>
</form>
```

- Ничего
- Появится сообщение с текстом «text»
- Появится сообщение с текстом «query»
- Появится сообщение с текстом «hidden»

Выберите верное утверждение

```
ul {
   li {
      &.active {
         a {
            color: red;
            text-decoration: none;
         }
      }
      a {
         text-transform: uppercase;
      }
   }
}
```

- Все ссылки внутри списка будут красного цвета и не подчёркнуты
- Все ссылки внутри списка будут в верхнем регистре и не подчёркнуты
- Все ссылки внутри активных элементов списка будут красного цвета и не подчёркнуты
- Все ссылки внутри списка будут красного цвета, а внутри активных элементов также не подчёркнуты

Блок 3 (владеть)

Можно ли вызвать «.w100» после объявления классов

```
.percent (@value) when (@value >= 0) {
   .w@{value} {
      width: ~"{@value}%";
   }
   .h@{value} {
      height: ~"{@value}%";
   }
   .percent(@value - 5);
}
.percent (0) {}
.percent (100);

.w100;
```

- Да.
- Нет. Вызвать можно только внутри селектора.
- Нет. Созданные классы доступны только в HTML.
- Нет. Такой класс не был создан.

Что произойдёт после нажатия на кнопку Find

```
$(function() {
   function Form(selector) {
      var self = this;
      this.selector = selector;
      this.field;
```

```
        this.setField = function(field) {
            this.field = field;
            return this;
        }

        this.onSubmit = function(){
            if (self.field) {
                alert(self.field.attr('name'));
            }
        };

        this.selector.submit(function() {
            self.onSubmit.call(self);
            return false;
        })
    }

    var search = new Form($("form"));
    search.setField(search.selector.find("input:last"));
    search.onSubmit = function() {

    }})
```

HTML

```
<form>
    <input type="hidden" name="journal" />

    <fieldset>
        <input type="text" name="query" placeholder="Search" />
    </fieldset>

    <fieldset>
        <button type="submit">
            <span>Find</span>
        </button>
    </fieldset>
</form>
```

- Ничего
- Появится сообщение с текстом «text»
- Появится сообщение с текстом «query»
- Появится сообщение с текстом «hidden»

Выберите правильное объяснение конструкции &.active

```
ul {
    li {
        &.active {
            a {
                color: red;
                text-decoration: none;
            }
```

```
        }
        a {
            text-transform: uppercase;
        }
    }
}
```

- Все элементы «li» имеющие класс «active»
- Все элементы «a» имеющие класс «active»
- Чётные элементы, имеющие класс «active»
- Все элементы «ul» имеющие класс «active»

Сколько записей будет иметь объект jQuery после выполнения следующей функции

```
$(".word").parents('ul')
```

HTML

```
<ul>
    <li class="feedtab news active">
        <a href="/news/">
            <b class="tab word">Новости</b>
        </a>
    </li>
    <li class="feedtab notifications">
        <a href="/notifications/">
            <b class="tab word">Ответы</b>
        </a>
    </li>
    <li class="feedtab comments">
        <a href="/comments/">
            <b class="tab word">Комментарии</b>
        </a>
    </li>
</ul>
```

- 0
- 1
- 2
- 3

Выберите правильный вариант оформления функции

- function test(param1, param2);
- function test(param1, param2){};
- function test($param1, $param2 = array()){};
- function test($param1 = defaultValue, $param2 = "2"){};

Вопрос 5
Что будет показано на экране после выполнения следующего кода

```
var type = "test";
switch(type) {
   case "test1":
      alert("test1");
   case "test2":
      alert("test2");
      break;
   case "test3":
      alert("test3");
   default:
      alert("default");
      break;
}
```

- test1
- test1, test2, test3
- test2
- default

Какой будет результат выполнения программного кода

```
abstract class Test {
   public $a;
   public $b;
   public $result;

   public function __construct($a, $b) {
      $this->a = $a;
      $this->b = $b;
   }

   abstract function calc();

   final public function getResult() {
      return $this->result;
   }
}
class Test1 extends Test {
   public function calc() {
      $this->result = $this->a + $this->b;
   }

}
class Test2 extends Test1 {
   public function calc() {
      parent::calc();
      $this->result *= 2;
   }
}
```

```
class Test3 extends Test2 {
    public function __construct($a, $b) {
        $this->a = $a * 2;
        $this->b = $b * 2;
    }
}

$obj = new Test3(2,3);
$obj->calc();
echo $obj->getResult();
```

•
Ошибка выполнения
•
5
•
10
•
20

Какой будет результат выполнения программного кода

```
function test($a, $b) {
    return $a + $b;
}
$func = "test";
$args = array("2", "3");
echo call_user_func_array($func, $args);
```

•
Ошибка выполнения
•
5
•
23
•
test23

Вопрос 5

Какие стили будут добавлены к элементу input после компиляции LESS файла

```
@defaultFontSize: 14px;

.padding(@value: 0px) {
    padding: @value;
    .margin(@value / 2);
}

.margin(@value: 0px) {
    margin: @value;
    .font-size(@defaultFontSize * 1.5);
}

.font-size(@value: @defaultFontSize) {
    font-size: @value;
}

.search(@value: 0px) {
```

```
    .padding(@value);
}

input {
    .search(20px);
}
```
•
.search(20px)
•
padding: 20px;
margin: 10px;
font-size: 21px;
•
padding: 0px
•
padding: 20px
Что будет показано на экране после выполнения следующего кода

```
var type = "test1";
switch(type) {
    case "test1":
        alert("test1");
    case "test2":
        alert("test2");
        break;
    case "test3":
        alert("test3");
    default:
        alert("default");
        break;
}
```
•
test1
•
test1, test2, test3
•
test1, test2
•
default
Какой будет результат выполнения программного кода

```
function test($a, $b) {
    return $a . $b;
}
$func = "test";
$args = array(2, 3);
echo call_user_func_array($func, $args);
```
•
Ошибка выполнения
•
5
•
23
•

test23
Сколько записей будет иметь объект jQuery после выполнения следующей функции

$(".word").parents("li:not(.active)")

HTML

```
<ul>
   <li class="feedtab news active">
      <a href="/news/">
         <b class="tab word">Новости</b>
      </a>
   </li>
   <li class="feedtab notifications">
      <a href="/notifications/">
         <b class="tab word">Ответы</b>
      </a>
   </li>
   <li class="feedtab comments">
      <a href="/comments/">
         <b class="tab word">Комментарии</b>
      </a>
   </li>
</ul>
```

- 0
- 1
- 2
- 3

Выберите правильный вариант оформления пустого класса

- class Test() {};
- class Test {};
- class Test();
- class Test;

Можно ли вызвать класс «w97» в HTML

```
<div class="w97">Тест</div>
```

LESS:

```
.percent (@value) when (@value >= 0) {
   .w@{value} {
      width: ~"{@value}%";
   }
   .h@{value} {
      height: ~"{@value}%";
   }
   .percent(@value - 5);
}
.percent (0) {}
```

.percent (100);

- Да.
- Нет. Созданные классы доступны только в CSS.
- Нет. Созданный класс не применим к тегу DIV.
- Нет. Такой класс не был создан.

Что произойдёт после нажатия на кнопку Find

```
$(function() {
   function Form(selector) {
      var self = this;
      this.selector = selector;
      this.field;

      this.setField = function(field) {
         this.field = field;
         return this;
      }

      this.onSubmit = function(){
         if (self.field) {
            alert(self.field.attr('name'));
         }
      };

      this.selector.submit(function() {
         self.onSubmit.call(self);
         return false;
      })
   }

   var search = new Form($("form"));
   search.setField(search.selector.find("input:last"));
})
```

HTML

```
<form>
   <input type="hidden" name="journal" />

   <fieldset>
      <input type="text" name="query" placeholder="Search" />
   </fieldset>

   <fieldset>
      <button type="submit">
         <span>Find</span>
      </button>
   </fieldset>
</form>
```

-

Ничего
•
Появится сообщение с текстом «text»
•
Появится сообщение с текстом «query»
•
Появится сообщение с текстом «hidden»
Выберите корректный для CSS селектор, описывающий ссылку в активном элементе

```
ul {
   li {
      &.active {
         a {
            color: red;
            text-decoration: none;
         }
      }
      a {
         text-transform: uppercase;
      }
   }
}
```

•
ul.active a
•
ul li &.active a
•
li.active a
•
li > a.active
Какие стили будут добавлены к элементу input после компиляции LESS файла

```
@defaultFontSize: 14px;

.padding(@value: 0px) {
   padding: @value;
}

.margin(@value: 0px) {
   margin: @value;
   .font-size(@defaultFontSize * 1.5);
}

.font-size(@value: @defaultFontSize) {
   font-size: @value;
}

.search(@value: 0px) {
   .padding(@value);
}

input {
   .search();
}
```

•
.search()

•
padding: 0px;
margin: 0px;
font-size: 14px;
•
padding: 0px;
margin: 0px;
font-size: 21px;
•
padding: 0px;
Какой будет результат выполнения программного кода

```
abstract class Test {
   public $a;
   public $b;
   public $result;

   public function __construct($a, $b) {
      $this->a = $a;
      $this->b = $b;
   }

   abstract function calc();

   final public function getResult() {
      return $this->result;
   }
}
class Test1 extends Test {
   public function calc() {
      $this->result = $this->a + $this->b;
   }

}
class Test2 extends Test1 {
   public function calc() {
      parent::calc();
      $this->result *= 2;
   }
}

class Test3 extends Test1 {
   public function __construct($a, $b) {
      $this->a = $a * 2;
      $this->b = $b * 2;
   }
}

$obj = new Test3(2,3);
$obj->calc();
echo $obj->getResult();
```

•
Ошибка выполнения
•

5
•
10
•
20
Выберите корректное описание для селектора «form fieldset input»

```
<form>
   <input type="hidden" name="journal" />

   <fieldset>
      <input type="text" name="query" placeholder="Search" />
   </fieldset>

   <fieldset>
      <button type="submit">
         <span>Find</span>
      </button>
   </fieldset>
</form>
```

•
Указывает на все теги input внутри формы
•
Указывает на все теги fieldset внутри формы
•
Указывает на тег input внутри тега fieldset
•
Указывает на теги input и fieldset
Выберите правильный вариант оформления наследования класса
•
class Test() extends Test2, Test3 {};
•
class Test extends Test2 extends Test3 {};
•
class Test extends Test2 {};
•
class Test() parents Test2,Test3;
Какой будет результат выполнения программного кода

```
class TestParent {
   public $result;

   public static function calc($a, $b) {
      return $a + $b;
   }

   public function getResult() {
      return $this->result;
   }
}
class Test extends TestParent {
   public function __construct($a, $b) {
      $this->result = self::calc($a, $b);
   }
```

```
    public static function calc($a, $b) {
        return $a * $b;
    }
}

$obj = new Test(2,3);
echo $obj->getResult();
```

•
Ошибка выполнения
•
5
•
23
•
6

Выберите результат, который вернёт функция, после выполнения следующего кода

```
var func = function(param) {
    param = param || 0;
    param *= 2;
    return param;
}

func();
```

•
0
•
2
•
undefined
•
NaN

Выберите правильный вариант оформления пустого абстрактного класса

•
abstr class Test() {};
•
abstract class Test {};
•
abstr class Test();
•
abstract class Test;

Какой будет результат выполнения программного кода

```
class TestParent {
    public $result;

    public static function calc($a, $b) {
        return $a + $b;
    }

    public function getResult() {
        return $this->result;
    }
}
```

```
class Test extends TestParent {
   public function __construct($a, $b) {
      $this->result = parent::calc($a, $b);
   }

   public static function calc($a, $b) {
      return $a * $b;
   }
}

$obj = new Test(2,3);
echo $obj->getResult();
```

•
Ошибка выполнения
•
5
•
23
•
6

Выберите корректное описание для селектора «form [type=’hidden’]»

```
<form>
   <input type="hidden" name="journal" />

   <fieldset>
      <input type="text" name="query" placeholder="Search" />
   </fieldset>

   <fieldset>
      <button type="submit">
         <span>Find</span>
      </button>
   </fieldset>
</form>
```

•
Указывает на все теги input внутри формы
•
Указывает на первый тег input
•
Указывает на тег input внутри тега fieldset
•
Указывает на тег button внутри fieldset

Выберите результат, который вернёт функция, после выполнения следующего кода

```
var func = function(param) {
   param = param || 0;
   param *= 2;
   return param;
}

func(2);
```

•
0
•

4
•
undefined
•
NaN
Выберите результат, который вернёт функция, после выполнения следующего кода

```
var func = function(param) {
   param = param || 0;
   param *= 2;
   return param;
}

func("test");
```

•
0
•
test
•
undefined
•
NaN
Выберите корректное описание для селектора «form fieldset:first-child»

```
<form>
   <input type="hidden" name="journal" />

   <fieldset>
      <input type="text" name="query" placeholder="Search" />
   </fieldset>

   <fieldset>
      <button type="submit">
         <span>Find</span>
      </button>
   </fieldset>
</form>
```

•
Указывает на первый элемент внутри формы
•
Указывает на первый элемент fieldset внутри формы
•
Указывает на первый элемент внутри тега fieldset
•
Указывает на последний элемент fieldset внутри формы
Какой будет результат выполнения программного кода

```
class TestParent {
   public $result;

   public static function calc($a, $b) {
      return $a + $b;
   }

   public function getResult() {
```

```
        return $this->result;
    }
}
class Test extends TestParent {
    public function __construct($a, $b) {
        $this->result = self::calc($a, $b);
    }

    public static function calc($a, $b) {
        $this->result = $a * $b;
        return $this->result;
    }
}

$obj = new Test(2,3);
echo $obj->getResult();
```

•
Ошибка выполнения
•
5
•
23
•
6

Выберите правильный вариант оформления интерфейса

•
interface iTest {};
•
public interface iTest {};
•
interface iTest();
•
class interface iTest;

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/question/edit.php?courseid=3756

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.