

**Примерные вопросы к экзамену  
по МДК.02.01 «Инфокоммуникационные системы и сети»  
для студентов группы П-1-15  
(2-й семестр 2017/2018 учебного года)**

Преподаватель, составитель: Караваев Сергей Владимирович ([s.v.karavaev@mpt.ru](mailto:s.v.karavaev@mpt.ru)).

В каждом экзаменационном билете — два теоретических вопроса и одно практическое задание.

Экзамен будет проведён в период с 01 по 07 июня 2018 г. включительно.

**Теоретические вопросы**

1. Гипертекст. Гипертекстовые системы. Разделение данных и формы их представления. Справка: методическое указание «[isis-01.pdf](#)».
2. HTML5: стандарты W3C и WhatWG, структура документа, процесс валидации разметки. Справка: [https://developer.mozilla.org/ru/docs/Learn/HTML/Введение\\_в\\_HTML/Начало\\_работы](https://developer.mozilla.org/ru/docs/Learn/HTML/Введение_в_HTML/Начало_работы), <https://validator.w3.org>.
3. Форма в HTML5: атрибуты формы, однострочное и многострочное поля ввода, флажок () , радиокнопка () , поле со списком, поле выбора файла, кнопки. Справка: 1) методическое указание «[04-blade.html](#)» (см. `{ { Form::model( . . . ) } }`); 2) : <http://htmlbook.ru/samhtml5/formy>.
4. SEO-friendly (user-friendly) URL: определение, способы реализации, восприятие пользователем (user experience), влияние на продвижение веб-ресурса. Справка: [https://ru.wikipedia.org/wiki/Семантический\\_URL](https://ru.wikipedia.org/wiki/Семантический_URL).
5. HTTP/1.1: позиция в стеке протоколов Интернета, структура сообщений (запросов и ответов). Справка: 1) <https://developer.mozilla.org/ru/docs/Web/HTTP>, 2) <https://developer.mozilla.org/ru/docs/Web/HTTP/Messages>.
6. Статусные коды HTTP/1.1: 200, 201, 301–304, 307, 400, 401, 403–406, 500. Справка: <https://developer.mozilla.org/ru/docs/Web/HTTP/Status>.
7. Заголовки запроса HTTP/1.1: 1) Accept; 2) Accept-Language; 3) Host; 4) If-Modified-Since. Справка: [https://ru.wikipedia.org/wiki/Список\\_заголовков\\_HTTP](https://ru.wikipedia.org/wiki/Список_заголовков_HTTP).
8. Заголовки ответа HTTP/1.1: 1) Content-Language; 2) Content-Length; 3) Content-Type; 4) Date; 5) Location. Справка: [https://ru.wikipedia.org/wiki/Список\\_заголовков\\_HTTP](https://ru.wikipedia.org/wiki/Список_заголовков_HTTP).
9. Метод HTTP GET: назначение, ограничения, идемпотентность. Привести пример сообщения. Справка: <http://lib.ru/WEBMASTER/rfc2068/section-9.html>.
10. Метод HTTP POST: назначение, ограничения, неидемпотентность. Сравнить типы содержимого application/x-www-form-urlencoded, multipart/form-data, text/plain. Привести пример сообщения. Справка: [https://ru.wikipedia.org/wiki/POST\\_\(HTTP\)](https://ru.wikipedia.org/wiki/POST_(HTTP)).
11. Шаблон проектирования веб-приложений POST/Redirect/GET. Привести пример реализации. Справка: [https://ru.bmstu.wiki/PRG\\_\(Post/Redirect/Get\)](https://ru.bmstu.wiki/PRG_(Post/Redirect/Get)).
12. Массивы в PHP: особенности реализации типа данных, функции обработки, сериализация (в JSON) и десериализация (из JSON), примеры. Справка: 1) <https://secure.php.net/manual/ru/language.types.array.php>; 2) <https://secure.php.net/manual/ru/ref.array.php>.
13. Объектная модель PHP. Понятия класса (class) и объекта (object). Особенности реализации инкапсуляции (public, protected, private), полиморфизма (interface, implements), наследования (extends). Справка: <https://secure.php.net/manual/ru/language.oop5.php>.
14. Особенности работы с файловой системой сервера при разработке и эксплуатации веб-приложения. Справка: 1) методическое указание «[file.pdf](#)»; 2) <https://laravel.ru/docs/v5/filesystem>.

15. Миграция схемы базы данных: определение, особенности реализации в веб-приложениях, примеры. Справка: 1) методическое указание «02-migrations.html»; 2) <https://laravel.ru/docs/v5/migrations>.
16. Модель. Объектно-реляционное несоответствие импеданса. Реализация ORM в веб-приложениях. Справка: 1) методические указания «05-1-to-m.html», «06-m-to-m.html»; 2) <https://laravel.ru/docs/v5/eloquent>.
17. Модель. ORM. Методы, реализующие вставку (INSERT), обновление (UPDATE), удаление (DELETE) кортежей из базы данных. Справка: <https://laravel.ru/docs/v5/eloquent>.
18. Модель. ORM. Выборка с условием (WHERE, BETWEEN, IN). Справка: <https://laravel.ru/docs/v5/queries>.
19. Модель. ORM. Упорядочение (ORDER BY), предел (LIMIT). Справка: <https://laravel.ru/docs/v5/queries>.
20. Автоматизированное интеграционное тестирование веб-приложения и системы управления базами данных (СУБД). Справка: <https://laravel.ru/docs/v5/database-testing>.
21. Контроллер (англ. controller) в архитектуре веб-приложения. Справка: 1) методическое указание «03-controller.html»; 2) <https://laravel.ru/docs/v5/controllers>.
22. Маршрутизация (англ. routing) в веб-приложениях. Справка: 1) методическое указание «03-controller.html»; 2) <https://laravel.ru/docs/v5/routing>.
23. REST: определение, методы, преимущества и недостатки, понятие RESTful API. Привести пример. Справка: 1) методическое указание «03-controller.html»; 2) <http://www.restapitutorial.ru/lessons/httpmethods.html>.
24. Шаблонизация в веб-разработке. Принцип работы шаблонизатора. Привести пример использования шаблонизатора. Справка: 1) методическое указание «04-blade.html»; 2) <https://laravel.ru/docs/v5/blade>.
25. Верификация данных в веб-приложениях: средства проверки, спецификация требований, Perl-совместимые регулярные выражения, примеры. Справка: 1) методическое указание «Request.pdf»; 2) <https://laravel.ru/docs/v5/validation>; 3) [http://nickolay.info/php/php\\_regexp.html](http://nickolay.info/php/php_regexp.html).
26. Cookie: определение, уязвимости. Приведите пример установки и аннулирования cookie. Справка: 1) <https://developer.mozilla.org/ru/docs/Web/HTTP/Куки>; 2) <https://secure.php.net/manual/ru/function.setcookie.php>.
27. Сеанс (сессия): определение, способы передачи идентификатора, уязвимости, примеры. Справка: 1) методическое указание «04-blade.html» (см. `$request->flash`); 2) <https://laravel.ru/docs/v5/session>.
28. SQL-инъекции I и II порядков: векторы атаки через веб-приложение, методика противодействия, примеры. Справка: [https://ru.wikipedia.org/wiki/Внедрение\\_SQL-кода](https://ru.wikipedia.org/wiki/Внедрение_SQL-кода).
29. Отражённые (reflected) и хранимые (stored) XSS: классификация, векторы атаки, методика противодействия. Справка: 1) методическое указание «04-blade.html»; 2) [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)).
30. Подделка межсайтового запроса (CSRF): векторы атаки, методика противодействия, примеры. Справка: 1) [https://ru.wikipedia.org/wiki/Межсайтовая\\_подделка\\_запроса](https://ru.wikipedia.org/wiki/Межсайтовая_подделка_запроса); 2) <https://laravel.ru/docs/v5/csrf>.
31. MITM: векторы атаки, противодействие. Справка: [https://ru.wikipedia.org/wiki/Атака\\_посредника](https://ru.wikipedia.org/wiki/Атака_посредника).
32. HTTP over TLS (HTTP/TLS): назначение, задействованные уровни стека протоколов Интернета, схема https.

### Практические задания

1. Сверстайте документ HTML5, реализующий принципы адаптивного веб-дизайна. Справка: методическое указание «[isis-01.pdf](#)».

2. С применением каркаса веб-приложения (Laravel или аналогичного) реализуйте модуль миграции схемы таблицы students, имеющей строковые атрибуты surname (фамилия), name (имя) и patronymic (отчество), атрибут birth\_date (дата рождения). Продемонстрировать миграцию и её откат. Справка: методическое указание «[02-migrations.html](#)».

3. С применением каркаса веб-приложения (Laravel или аналогичного) реализуйте модуль миграции схемы таблицы products, имеющей как минимум атрибут title (наименование) строкового типа и атрибут price (цена) денежного типа. Продемонстрировать миграцию и её откат. Справка: методическое указание «[02-migrations.html](#)».

4. С применением каркаса веб-приложения (Laravel или аналогичного) реализуйте модуль миграции схем двух таблиц реляционной базы данных, связанных отношением «один-ко-многим». Продемонстрируйте миграцию и её откат. Справка: методические указания «[02-migrations.html](#)», «[05-1-to-m.html](#)».

5. С применением каркаса веб-приложения (Laravel или аналогичного) реализуйте модуль миграции схем двух таблиц реляционной базы данных, связанных отношением «многие-ко-многим». Продемонстрируйте миграцию и её откат. Справка: методические указания «[02-migrations.html](#)», «[06-m-to-m.html](#)».

6. Реализуйте модуль верификации данных, передаваемых веб-приложению в запросе HTTP. Справка: 1) методическое указание «[Request.pdf](#)»; 2) <https://laravel.ru/docs/v5/validation>.

7. С использованием ORM произведите вставку и удаление из таблицы products кортежа с некоторыми значениями атрибутов title и price. Справка: <https://laravel.ru/docs/v5/eloquent>.

8. С использованием ORM произведите выборку из таблицы products одного кортежа с указанным значением первичного ключа.

9. С использованием ORM произведите выборку по значению первичного ключа произвольного кортежа из таблицы products и всех связанных с ним отношением «один-ко-многим» кортежей из таблицы images.

10. С использованием ORM произведите упорядоченную выборку из таблицы products всех кортежей, значения атрибута price которых находятся в установленном интервале («от и до включительно»). Справка: <https://laravel.ru/docs/v5/queries>.

11. С использованием ORM произведите выборку из таблицы products всех кортежей, значения атрибута title которых являются элементами множества {Молоко, Хлеб}. Справка: <https://laravel.ru/docs/v5/queries>.

12. Проведите автоматизированное интеграционное тестирование веб-приложения и системы управления базами данных (СУБД). Справка: <https://laravel.ru/docs/v5/database-testing>.

13. Реализуйте модуль веб-приложения, позволяющий загрузить по HTTP и сохранить на сервере произвольный файл. Справка: методическое указание «[file.pdf](#)».

14. Спроектируйте и реализуйте RESTful API для операций CRUD (create, read, update, delete) над сущностью «Книга»; укажите возможные URL и методы HTTP, сопоставьте их с методами контроллера. Справка: 1) методическое указание «[03-controller.html](#)»; 2) <http://sap.pitroff.ru/tehnologii/rest/rest-eto-ne-pro-otdyih-chast-pervaya-chto-takoe-rest>; 3) <https://laravel.ru/docs/v5/routing>.

15. Разработайте шаблон формы HTML5 для подачи запроса на создание нового кортежа в таблице products. Шаблон должен обеспечивать защиту от XSS и CSRF. Справка: 1) методическое указание «04-blade.html»; 2) <https://laravel.ru/docs/v5/blade>.

16. Выполняется запрос: `SELECT * FROM users WHERE username = '$username' AND password = '$password'`; (значения \$username и \$password приняты по HTTP и интерполируются без фильтрации). Укажите, каким образом может быть проведена SQL-инъекция I порядка, которая приведёт к выборке всех кортежей, если существует хотя бы один, и какие меры необходимо предпринять для устранения уязвимости.