

Б1.В.07

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА)

Учебная дисциплина	<i>Архитектура web-приложений</i>
По направлению подготовки	<i>09.03.03</i>
	<i>«Прикладная информатика»</i>
Профиль (программа бакалавриата)	<i>«Дизайн и разработка цифровых программных продуктов»</i>
Форма обучения	<i>Очная</i>

Оценочные материалы (средства) дисциплины рассмотрены (актуализированы) и утверждены на заседании кафедры прикладной информатики и высшей математики

Протокол заседания № 10 от «25» мая 2026 г.

Заведующий кафедрой Стрекалова Наталья Борисовна

ОГЛАВЛЕНИЕ

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....	3
2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	5
3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ.....	15
4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНОВАНИЯ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ.....	18
5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ.....	25
6. ЛИСТ СОГЛАСОВАНИЯ.....	41

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Оценочные материалы (средства) сформированы в соответствии с ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 «Прикладная информатика», утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922 (с изменениями и дополнениями от 26.11.2020, 08.02.2021). В соответствии с матрицей компетенций основной профессиональной образовательной программы «Дизайн и разработка цифровых программных продуктов» (уровень бакалавриата) в процессе обучения по дисциплине «Архитектура web-приложений» происходит формирование закрепленных за дисциплиной компетенций обучающихся. Оценка сформированности компетенций на каждом этапе обучения происходит через оценку планируемых результатов обучения по дисциплине (знаний, умений, навыков).

Результаты освоения образовательной программы (компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения по дисциплине	Этапы формирования компетенции (семестры и темы)	Оценочные средства
ПК-2. Способен проектировать цифровой программный продукт	ПК-2.1. - Разрабатывает модель цифрового программного продукта с учетом внедрения передовых ИТ-решений прикладной сферы	Знать: - понятие архитектуры программного обеспечения, традиционные и современные архитектуры информационных систем, архитектуры web-приложения; Уметь: - проектировать web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющихся цифровых и отраслевых решений; Владеть: - навыками моделирования и работы с современными средствами построения и визуализации моделей web-приложения, в том числе удаленного доступа (MS Visio, BP Win, MIRO);	Семестр 5 Тема 1. Архитектура как форма концептуального существования ПО. Эволюция архитектур Тема 2. Архитектуры «файл-сервер», «клиент-сервер» Тема 3. Облачные и распределенные вычисления Тема 4. Сервис-ориентированные архитектуры (SOA) Тема 5. Архитектура web-приложений Тема 6. Методология проектирования web-архитектуры	- устный опрос по теме 1; - выступление с докладом по темам 2-4; - практические работы по темам 2-5; - индивидуальное проектное задание по теме 6; - ответ на теоретический вопрос на экзамене.
	ПК-2.2. - Проектирует	Знать: - понятие базы	Тема 5. Архитектура	- практическая работа по

	базы данных, в том числе используя специализированные цифровые платформы	данных, распределенной базы данных облачных баз данных; Уметь: - проектировать базы данных для web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющихся цифровых и отраслевых решений; Владеть: - навыками моделирования и работы с современными средствами построения и визуализации моделей web-приложения и баз данных, в том числе удаленного доступа (MS Visio, BP Win, MIRO)	web-приложений Тема 6. Методология проектирования web-архитектуры	теме 5; - индивидуально проектное задание по теме 6; - ответ на теоретический вопрос на экзамене.
	ПК-2.4. Проводит анализ современных программно-технических средств, специализированных цифровых платформ, облачных и отраслевых решений для разработки цифрового программного продукта	Знать: - методологию разработки архитектуры web-приложения и необходимые для этого программно-аппаратные средства; Уметь: - осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации web-архитектуры; Владеть: - навыками анализа и оценки цифровых платформ, облачных решений и программно-аппаратных средств;	Семестр 5 Тема 1. Архитектура как форма концептуального существования ПО. Эволюция архитектур Тема 2. Архитектуры «файл-сервер», «клиент-сервер» Тема 3. Облачные и распределенные вычисления Тема 4. Сервис-ориентированная архитектура (SOA) Тема 6. . Методология проектирования web-архитектуры	- устный опрос по теме 1; - выступление с докладом по темам 2-4; индивидуально проектное задание по теме 6; - ответ на теоретический вопрос на экзамене.
ПК-3. Способен разрабатывать цифровой	ПК-3.2. Обеспечивает интеграцию	Знать: - особенности web-архитектуры и основные	Тема 5. Архитектура web-приложений	- практическая работа по теме 5;

программный продукт	программных модулей и компонентов цифрового программного продукта на основе облачных и отраслевых ИТ-решений	способы реализации современных web-архитектурных решений, их достоинства и недостатки, интеграционные сложности и риски; Уметь: - осуществлять подбор средств разработки web-приложения с учетом их интеграционных возможностей, потенциальных отказов и безопасности работы; Владеть: - навыками анализа и оценки потенциальных угроз в разработке web-приложений.	Тема 6. Методология проектирования web-архитектуры	- индивидуально е проектное задание по теме 6; - ответ на теоретический вопрос на экзамене
---------------------	--	---	---	---

2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Устные опросы

Примерные вопросы для проведения устного опроса по теме 1 «Архитектура как форма концептуального существования ПО. Эволюция архитектур»

1. Дайте понятие архитектуры программного обеспечения
2. Как согласуются модели жизненного цикла разработки программного обеспечения и вопросы разработки архитектуры?
3. Перечислите компоненты информационных систем
4. В чем особенность идеологического подхода к сути архитектуры программного обеспечения.
5. В чем особенность конструктивного подхода к сути архитектуры программного обеспечения
6. Перечислите уровни архитектуры web-приложений
7. Охарактеризуйте слой представления.
8. Охарактеризуйте слой доступа к данным.
9. Назначение бизнес-логики
10. Назначение подсистему управления
11. Назначение уровня обслуживания данных
12. Роль архитектурных решений для разработки web-приложений
13. Охарактеризуйте логическую архитектуру программного обеспечения
14. Охарактеризуйте физическую архитектуру программного обеспечения
15. Охарактеризуйте поведенческую архитектуру программного обеспечения
16. Охарактеризуйте логическую архитектуру программного обеспечения
17. Охарактеризуйте временную архитектуру программного обеспечения

Критерии оценивания устного опроса:

оценка «отлично» выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов, активно дополнял ответы других обучающихся или задавал им дополнительные вопросы;

оценка «хорошо» выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов или ответил кратко на ряд вопросов;

оценка «удовлетворительно» выставляется обучающемуся, если он кратко ответил на один из задаваемых вопросов или просто дополнял ответы других обучающихся, задавал им дополнительные вопросы;

оценка «неудовлетворительно» выставляется обучающемуся, если он не смог ответить правильно на заданный вопрос, либо не отвечал на вопросы и не участвовал в их обсуждении.

Подготовка и выступление с докладом

Примерные темы для подготовки докладов

по темам 2-4 «Архитектуры «файл-сервер», «клиент-сервер», «Облачные и распределенные вычисления», «Сервис-ориентированные архитектуры (SOA)»

1. Суть, достоинства и недостатки централизованной архитектуры ПО.
2. Суть, достоинства и недостатки «файл-серверной» архитектуры ПО.
3. Суть, достоинства и недостатки однозвенной «клиент-серверной» архитектуры ПО.
4. Суть, достоинства и недостатки двухуровневой «клиент-серверной» архитектуры ПО.
5. Суть, достоинства и недостатки трехуровневой «клиент-серверной» архитектуры ПО.
6. Суть, достоинства и недостатки вертикальной распределенной архитектуры ПО.
7. Суть, достоинства и недостатки горизонтальной распределенной архитектуры ПО.
8. Суть, достоинства и недостатки «облачной» архитектуры (бессерверной, монолитной).
9. Суть, достоинства и недостатки internet/intanet-архитектуры ПО.
10. Суть, достоинства и недостатки сервис-ориентированной архитектуры ПО.
11. Суть, достоинства и недостатки микросервисной архитектуры ПО.
12. Суть, достоинства и недостатки одностраничного приложения (SPA).
13. Суть, достоинства и недостатки многостраничного приложения (MPA).
14. Суть, достоинства и недостатки прогрессивной архитектуры (PWA).

Требования к докладу и презентации:

1. Материалы доклада должны «базировать» на современном представлении архитектур информационных систем.

2. В докладе должны быть освещены все вопросы: суть описываемой архитектуры, принципиальная схема взаимодействия компонентов ИС и подробное описание особенностей их взаимодействия, перечисление видов ПО, для которых такая архитектура наиболее характерна, представление хотя бы одного примера существующей информационной системы, спроектированной по данной архитектуре, описание достоинств ИС (не только перечисление, но и пояснения), описание недостатков (сложностей) ИС (не только перечисление, но и пояснения).

3. Доклад оформить по требованиям вуза с титульным листом, содержанием, введением, предметной частью, заключением, списком литературы, приложением (при необходимости)

4. Выступление с докладом не должно занимать более 8 минут и должно сопровождаться презентацией.

5. На слайдах презентации должны быть представлены ключевые моменты доклада, вспомогательные графические образы, дополняющие речь докладчика. Текст на слайды выносить в минимальном объеме, а его размер должен обеспечивать хорошую читаемость слайдов.

6. Доклад сдать в электронном и распечатанном виде преподавателю, презентацию – только в электронном виде.

Критерии оценивания доклада:

оценка «отлично» выставляется обучающемуся, если доклад полностью раскрывает тему, структура презентации логично выстроена, обучающийся хорошо ориентируется в материале и текст доклада, в основном, рассказывает (а не читает), отвечает на дополнительные вопросы; сопровождающая доклад презентация выполнена в соответствии с требованиями и дополняет доклад, докладчик уложился в регламент;

оценка «хорошо» выставляется обучающемуся, если доклад в целом раскрывает тему, но остаются не освещенные отдельные стороны, не на все дополнительные вопросы обучающийся может дать ответ; структура презентации логично выстроена, текст доклада обучающийся, в основном, рассказывает (а не читает); презентация выполнена в соответствии с требованиями, но есть ряд замечаний по ее оформлению, доклад и презентация дополняют друг друга, докладчик уложился в регламент;

оценка «удовлетворительно» выставляется обучающемуся, если его доклад не раскрывает тему в достаточной мере, не отвечает на дополнительные вопросы; к оформлению презентации есть много замечаний; презентация выполнена не по требованиям, доклад дублирует презентацию, а не дополняет ее, обучающийся не укладывается в регламент или наоборот;

оценка «неудовлетворительно» выставляется обучающемуся, если доклад отсутствует или совсем не раскрывает тему, не соответствует требованиям, обучающийся не может ответить на дополнительные вопросы.

Практические работы

Практическое задание по теме 2 «Архитектуры «файл-сервер», «клиент-сервер»»

Задача: разработать программу решения задачи на выбранном языке программирования в виде клиент-серверного приложения.

Технология работы:

1. На компьютере установить следующие средства разработки: Java Development Kit (jdk); IntelliJIDEA (CommunityEdition, свободно распространяемый вариант); СУБД на выбор: MS SQL Server, MySQL, PostgreSQL, Oracle.

2. Создать новый пустой проект (File — NewProject...). Записать его в каталог, название которого соответствует вашей фамилии. 2.1. При создании проекта необходимо указать, какое JDK вы будете использовать, если его не будет в предложенном меню, то вручную укажите путь до него. Например: C:\Program Files\Java\jdk1.7.0_15.

3. Создать пустой класс, содержащий статический метод выполнения (JavaApplication). Реализация метода может быть примерно следующей: `public static void main (String argc []) { System.out.println ("Hello, World!"); }`

4. Требования к выполнению вариативного задания.

- a. Вариативное задание состоит из трех задач варианта.
- b. Вариативное задание должно быть оформлено в отдельном классе.
- c. Класс должен реализовывать созданный вами интерфейс Result, который будет иметь один метод, возвращающий строку результата вычислений getResult () без входных параметров.
- d. Все входные параметры должны поступать в класс с помощью перегрузок конструкторов и должны совпадать с индивидуальным заданием (пример: в задании 1 должно быть описано три перегруженных конструктора, которые будут принимать String, Double [], List<Double>).
- e. В классе должно быть описано поле, в котором будут храниться распарсенные значения (пример: в задании 1 в конструктор класса поступает обычная строка «11, 32, 1, 22, 14», конструктор должен разбить данную строку по запятой на массив чисел и заполнить поле, с которым в дальнейшем мы будем работать для вычисления максимального значения).

5. Для отладки и проверки работоспособности выполненного вариативного задания создаем отдельный класс с методом main, который будет взаимодействовать с созданным ранее

классом (с классом, в котором выполнено вариативное задание) и позволит вводить данные (в консоли) для вычисления, после чего вывести результат на экран.

6. Запустить созданную программу на выполнение, убедиться, что на консоль (стандартный поток вывода) выводится надпись.

7. Используя класс `java.net.ServerSocket`, написать простейший echo-сервер. Технические требования, предъявляемые к серверу:

- a. Порт, на котором запускается сервер — 12345. Транспортный протокол — TCP.
- b. Количество одновременно подключенных клиентов — 1. То есть использование потоков (экземпляров класса `Thread`) на первом этапе не предусматривается.
- c. Сервер должен обеспечивать echo-функционал, то есть просто передавать обратно клиенту полученный буквенный символ.
- d. На консоль (с помощью метода `System.out.println ()`) должны выводиться основные этапы работы программы (клиент установил соединение, был получен и передан обратно символ, соединение разорвано).

8. Изменить написанный класс echo-сервера. Сервер должен производить вычисления в соответствии с индивидуальным заданием, выполненным в пункте 2.1.3.2. В ответ клиент должен получить результат выполненной операции.

9. Написать клиентскую часть `Socket`:

- a. Клиент должен будет подключиться к созданному ранее серверу.
- b. Клиенту нужно передать данные параметров в соответствии с индивидуальным заданием, в ответ получить результат действий.

10. Произвести сборку серверной и клиентской части в `jar` файл.

11. Запустить `jar`-файлы серверной и клиентской части.

12. оформить отчет, включив в него задание, блок-схему алгоритма (алгоритмов), текст программы и скриншот результата выполнения задачи.

Варианты задач:

1. Найти максимальное значение среди множества чисел (без использования класса `Math`). Числа должны поступать в виде: строки с некоторым разделителем (пример: «11, 32, 1, 22, 14»); в массиве; списком чисел.

2. Сортировка списка слов, без использования сторонних классов сортировки, например `Collections` метод `sort ()`. Слова должны поступать сплошным текстом с разделителем; списком; отдельными значениями данных.

3. Подсчет одинаковых слов. Слова должны поступать сплошным текстом и с разделителем; списком; отдельными значениями данных.

4. Подсчет четных и нечетных символов. Числа должны поступать в виде строки с некоторым разделителем (пример: «11, 32, 1, 22, 14»); в массиве; списком чисел.

5. Конвертер систем счисления. На вход поступает число, которое мы хотим конвертировать; система счисления конвертируемого числа; система счисления, в которую мы хотим преобразовать число. Числа должны поступать в виде строки с некоторым разделителем; в массиве; списком чисел.

6. Вывести уравнение прямой, проходящей через две точки. На вход поступает 4 числа: x_1 , y_1 , x_2 , y_2 . Числа должны поступать в виде строки с некоторым разделителем; в массиве; списком чисел.

7. Дан текст. Определите процентное отношение строчных и прописных букв к общему числу символов в нем. На вход поступает текст в виде строки.

8. Дана строка, состоящая из слов и чисел, отделенных друг от друга разделяющим символом (при этом один символ будет служебным, например «.», который будет характеризовать вещественные числа). Сформировать три строки, одна из которых содержит только целые числа, встречающиеся в исходной строке, вторая — только вещественные числа, а третья — оставшиеся слова. Текст должен поступать сплошным текстом с разделителем.

9. Подсчет одинаковых символов. Символы должны поступать сплошным текстом с разделителем; списком; отдельными значениями данных.

10. Наибольший общий делитель (НОД) чисел (Например: 3430 и 1365 — это 35. Другими словами, 35 — наибольшее число, на которое и 3430 и 1365 делятся без остатка). Числа должны поступать в виде строки с некоторым разделителем; в массиве; списком чисел.

11. Для каждого натурального числа в промежутке от m до n вывести все делители. Числа должны поступать в виде строки с некоторым разделителем; в массиве; отдельными значениями данных.

12. Перевод римских чисел в арабские (например: XIV — 13). Поступает в виде строки.

13. Калькулятор. Формула должна поступать в виде в виде текста (пример: «4/2»); отдельными значениями данных.

14. Перемножение матриц. Числа должны поступать в виде строки с некоторым разделителем; в массиве; отдельными значениями данных.

15. Пользователь вводит несколько целых чисел, представляющих собой последовательность. Требуется ее оценить: является ли последовательность возрастающей; есть ли в ней одинаковые элементы; является ли она знакопеременной (положительные и отрицательные числа чередуются). Числа должны поступать в виде строки с некоторым разделителем; в массиве; отдельными значениями данных.

Критерии оценивания:

оценка «отлично» выставляется обучающемуся, если задание выполнено в полном объеме – безошибочно решены задачи, правильно работают классы и корректно передаются данные между ними, алгоритм(ы) оформлены без ошибок, отчет содержит всю информацию и оформлен в соответствии с требованиями;

оценка «хорошо» выставляется обучающему, если задание выполнено в полном объеме, но есть незначительные ошибки в решении задач, классы работают правильно и корректно передаются данные между ними, алгоритм(ы) оформлены без ошибок или с незначительными нарушениями, отчет содержит всю информацию и оформлен в соответствии с требованиями или с незначительными отклонениями от них;

оценка «удовлетворительно» выставляется обучающему, если задание выполнено частично (например, решены не все задачи), между классами есть передача данных, но не весь код работает корректно, алгоритм(ы) оформлен с нарушениями(или отсутствует), отчет содержит не всю информацию и оформлен с отклонениями от требований;

оценка «неудовлетворительно» выставляется обучающему, если программа не представлена или между классами не передачи данных, демонстрирующих клиент-серверное взаимодействие, или совсем не представлен отчет о работе.

Практическое задание по теме 3 «Облачные и распределенные вычисления»

Задача: Разработать концепцию распределенной информационной системы для выбранной предметной области.

Список предметных областей:

1. Агрегатор по оказанию услуг
2. Информационный портал
3. Обучающая платформа
4. Онлайн-игра
5. Сетевой журнал заказов
6. Система продаж туров и экскурсий
7. Страхование
8. Электронное правительство
9. Электронный документооборот в компании

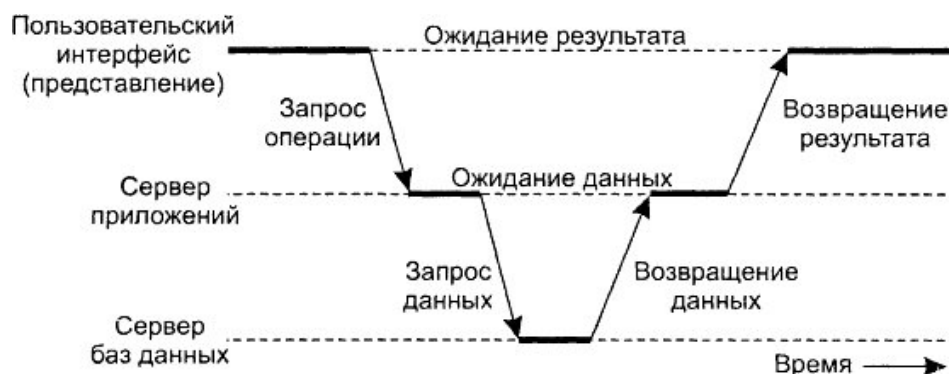
Технология работы:

1. Придумать идею информационной системы в выбранной предметной области
2. Зафиксировать назначение, цели и задачи системы
3. Рассмотреть аспекты распределенности системы

4. Схематично описать структуру распределенной системы и ее особенности (см. возможные варианты описания)



5. Разработать структуру связи для распределенной системы, указав не менее 10 сообщений и не менее 3 внутренних объектов.



6. Для каждого сервиса определить все сущности, требующие именования в системе, и ответить на вопросы: 1) как будет организовано адресное пространство для этих сущностей – плоско или иерархически; 2) будет ли использоваться несколько схем адресации для этих сущностей, если да, то как будет устанавливаться соответствие между этими адресами; 3) это сущности мобильные или стационарные, если мобильные, как к ним будет осуществляться доступ.

Пример:

Сущности клиники: пациент, врач, стационар, отделение, палата, койка

Адресное пространство: иерархичное, поскольку это удобней и привычней

Адресный путь к клинике: \город\район\клиника

Адресный путь(и) внутри клиники:

специальность(врача)\отделение\палата\койка\пациент\карта

Сущности город, район и клиника стационарные.

Сущность койка стационарная, содержит нулевое значение, если свободна, и ссылку на пациента, если занята.

Сущность пациент мобильная через идентификатор карты, содержащей год рождения, год поступления и id пациента

Критерии оценивания:

оценка «отлично» выставляется обучающемуся, если задание выполнено в полном объеме – безошибочно выполнены все 6 заданий, концепция сопровождается подробным поясняющим текстом, работа оформлена в соответствии с требованиями;

оценка «хорошо» выставляется обучающему, если задание выполнено в полном объеме (выполнены все 6 заданий), но есть незначительные ошибки в концептуальных решениях; концепция сопровождается поясняющим текстом, работа оформлена в соответствии с требованиями с незначительными отклонениями;

оценка «удовлетворительно» выставляется обучающему, если задание выполнено частично (но не менее четырех первых заданий), есть незначительные ошибки в концептуальных решениях; концепция сопровождается поясняющим текстом, работа оформлена в соответствии с требованиями с незначительными отклонениями;

оценка «неудовлетворительно» выставляется обучающему, если работа не выполнена или выполнено менее четырех заданий; или совсем не представлено описание концепции.

Практические задания по теме 5 «Архитектура web-приложений»

Задача: Создание веб-приложения со схемой разделения данных MVC.

Список предметных областей:

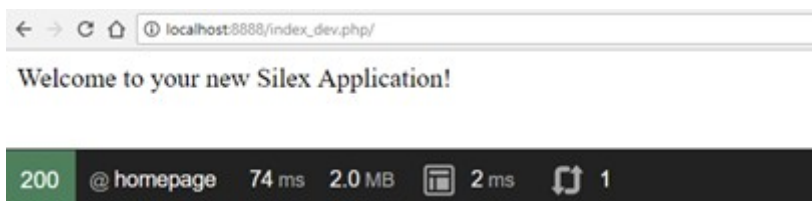
1. Система документооборота
2. Приложение стоматологическая клиника
3. Система шиномонтажа
4. Сайт кинотеатра
5. Приложение покупки билетов в кинотеатре
6. Приложение для боулинг-клуба
7. Приложение для гольф-клуба
8. Система управления персоналом
9. Система для управления заводом запчастей
10. Система учета медикаментов

Технология работы:

1. Устанавливаем фреймворк Silex через утилиту Composer с помощью командной строки: `composer create-project fabpot/silex-skeleton lab`

2. Запускаем встроенный PHP-сервер через Composer, предварительно настроив время ожидания процесса на бесконечность: `set "COMPOSER_PROCESS_TIMEOUT=0" composer run`

3. Проверяем результат настроек, запустив в браузере ссылку `http://localhost:8888/index_dev.php/` Если ошибок нет, мы увидим страницу-приветствие с текстом "Welcome to your new Silex Application!", а внизу debug-панель:



4. Создаем базу данных (по варианту). Например, список сотрудников с указанием, работает ли он в данный момент или уволен:

```
CREATE TABLE IF NOT EXISTS `employee` (`id` int(11) NOT NULL, `name` varchar(128) NOT NULL, `status` enum('active','inactive') NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
INSERT INTO `employee` (`id`, `name`, `status`) VALUES (1, 'Linus Torvalds', 'active'), (2, 'Eugene Kaspersky', 'inactive'), (3, 'Bill Gates', 'active'), (4, 'Steve Jobs', 'inactive'), (5, 'Sid Meier', 'active');
ALTER TABLE `employee` ADD PRIMARY KEY (`id`); ALTER TABLE `employee` MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

5. Для подключения к БД:

используем библиотеку Doctrine DBAL: `composer require doctrine/dbal:~2.2`

в файле `php.ini` следует включить расширение MySQL, раскомментировав строчку: `extension=php_pdo_mysql.dll`

добавляем в файл `config/prod.php` следующие строки, вписав свои параметры БД, логин и пароль:

```
$app['db.options'] = array_merge([ 'driver' => 'pdo_mysql', 'dbname'=> 'БД','host' => 'localhost','user' => 'ЛОГИН','password' => 'ПАРОЛЬ', 'charset' => 'utf8'], isset($app['db.options']) ? $app['db.options']: []);
$app->register(new Silex\Provider\DoctrineServiceProvider());
```

6. Создадим обработчик вывода списка активных или уволенных сотрудников, в зависимости от передаваемого статуса в GET-параметре. Добавляем в контроллер src/controllers.php следующие строки:

```
$app->get('/employee/{status}', function ($status) use ($app) {$sql = 'SELECT * FROM employee WHERE status =?';
$employeeList = $app['db']->fetchAll($sql, [$status]);
return $app['twig']->render('employees.html.twig', ['status' => $status, 'employees' => $employeeList,]);})
```

7. Через шаблонизатор Twig получаем HTML-представление. Для этого создаем соответствующий файл templates/employees.html.twig и заполняем его:

```
{% extends "layout.html.twig" %}
{% block content %}
    <h1>{{ status == 'active' ? 'Работают' : 'Уволенные' }}</h1>
    <table>
        <thead>
            <tr>
                <th>№</th>
                <th>Имя</th>
            </tr>
        </thead>
        <tbody>
            {# выводим весь список сотрудников #}
            {% for employee in employees %}
                <tr>
                    <td>{{ employee.id }}</td>
                    <td>{{ employee.name }}</td>
                </tr>
            {% endfor %}
        </tbody>
    </table>
{% endblock %}
```

8. Улучшим визуальное отображение шаблона. Для этого, в общем шаблоне templates/layout.html.twig можно посмотреть путь, по которому подключен CSS-файл, который находится по адресу web/css/main.css. Добавим стили в css-файл.

```
table { width: 100%; text-align: left;border-spacing: 0;}
table td {
    border-top: 1px solid black;}
```

9. Проверяем результат работы:

<http://localhost:8888/index.php/employee/inactive>.

<http://localhost:8888/index.php/employee/active><http://localhost:8888/index.php/employee>

10. Оформить отчет, включив в него задание, программные коды и скриншот результата выполнения.

Критерии оценивания:

оценка «отлично» выставляется обучающемуся, если задание выполнено в полном объеме и безошибочно, запросы к БД работают корректно, отчет содержит всю информацию и оформлен в соответствии с требованиями;

оценка «хорошо» выставляется обучающему, если задание выполнено в полном объеме, но есть незначительные ошибки, запросы к БД работают правильно и корректно передаются данные, отчет содержит всю информацию и оформлен в соответствии с требованиями или с незначительными отклонениями от них;

оценка «удовлетворительно» выставляется обучающему, если задание выполнено частично, запросы к БД работают не в полном объеме, а сама БД разработана примитивно, отчет содержит не всю информацию и оформлен с отклонениями от требований;

оценка «неудовлетворительно» выставляется обучающему, если работа или отчет не представлены или представленное приложение не работает.

Индивидуальное проектное задание «Интернет-портал организации»

по теме 6 «Методология проектирования web-архитектуры»

Индивидуальное проектное задание «Сайт организации» выполняется на основе кейс-материалов, содержащих структурную и функциональную модель предприятия (объекта автоматизации). В качестве кейс-материалов могут использоваться: 1) материалы предприятия, на котором обучающийся проходил учебную или производственную практику; 2) материалы предприятия, на котором работает обучающийся; 3) материалы кейс-ситуации, предложенные кафедрой. Индивидуальное проектное задание состоит из пяти заданий, соответствующих темам дисциплины и формируемым компетенциям. По итогам работ все выполненные задания собираются в единый документ, структурируются и оформляются в соответствии с требованиями вуза. Выполнение индивидуального проектного задания обязательно для всех обучающихся, т.к. является одним из условий допуска к экзамену. Оценивание индивидуального проектного задания производится целостно (после выполнения всех проектных заданий).

На кафедре разработаны кейсы для следующих предприятий:

1. Система документооборота.
2. Приложение стоматологическая клиника.
3. Система шиномонтажа.
4. Сайт кинотеатра.
5. Приложение покупки билетов в кинотеатре.
6. Приложение для боулинг-клуба.
7. Приложение для гольф-клуба.
8. Система управления персоналом.
9. Система для управления заводом запчастей.
10. Система учета медикаментов.

Индивидуальное проектное задание № 6.1

Задание: на основе приведенной структурной и функциональной модели объекта автоматизации построить диаграммы бизнес-процессов предприятия.

Технология работы:

1. Ознакомьтесь с информацией об объекте автоматизации.
2. Создайте модели основных бизнес-процессов, обеспечивающих и управленческих процессов (использовать можно любые автоматизированные средства Visio, BP Win, MIRO и др.).
3. Проведите декомпозицию процессов.
4. Определите цели и задачи автоматизации процессов.
5. В соответствии с целями и задачами автоматизации определите основные компоненты (модули, составляющие) интернет-портала организации (не менее 4-х).

Индивидуальное проектное задание № 6.2

Задание: на основе структурной модели объекта автоматизации сформулировать бизнес-требования, системные и функциональные требования к интернет-порталу организации. В качестве исходных данных выступает кейс-материалы по предприятию и результаты предыдущего задания.

Технология работы:

1. Ознакомьтесь с информацией об объекте автоматизации.
2. На основе описания объекта автоматизации сформулируйте цели и задачи разработки и использования интернет-портала организации.
3. Определите основных участников проекта, создайте соответствующих пользователей и группы пользователей интернет-портала организации.
4. На основе сформулированных целей разработки интернет-портала организации сформулируйте бизнес-требования к portalу
5. На основе приведенной модели объекта автоматизации и бизнес-требований сформулируйте функциональные требования к portalу.
6. На основе приведенной модели объекта автоматизации сформулируйте системные требования к portalу.
7. Сформируйте модель требований.
8. Сформируйте матрицу зависимостей между требованиями.

Индивидуальное проектное задание №6.3

Задание: на основе структурной модели объекта автоматизации сформировать программно-аппаратную часть портала - определить основные сетевые службы, требуемые для реализации интернет-портала, определить требования к аппаратному обеспечению, построить модель портала.

Технология работы:

1. Определите основные сетевые службы, которые понадобятся для работы портала (firewall, ftp-сервер, print- сервер, сервер БД, сервер приложений, web- сервер и т.д.).
2. Определите требования к аппаратному обеспечению портала.
3. Постройте модель интернет-портала (использовать можно любые автоматизированные средства Visio, BP Win, MIRO и др.). Обоснуйте свое решение.
4. Определите меры защиты интернет-портала. Укажите на диаграмме, отображающей модель корпоративной сети, соответствующие службы.
5. Определите меры по поддержанию работоспособности интернет-портала.

Индивидуальное проектное задание № 6.4

Задание: определить на основе существующих требований к интернет-порталу перечень типовых и/или облачных решений, которыми можно воспользоваться в работе интернет-портала (синтез из существующих на рынке типовых решений).

Технология работы:

1. Выделите подмножество требований, реализуемых с помощью тех или иных типовых / облачных решений.
2. Постройте модель интернет-портала на основе существующих типовых / облачных решений.
3. Проведите анализ двух моделей интернет-портала с точки зрения технологической и экономической эффективности, потенциальных рисков и угроз.

Критерии оценивания индивидуального проектного задания:

оценка «отлично» выставляется обучающемуся, если работа выполнена в срок и в полном объеме (100%), четко прослеживается преемственность выполненных заданий (каждое следующее задание базируется на результатах предыдущего задания); сделанные проектные предложения соответствуют современному уровню развития ИТ, работа представлена в «читабельном» и оформленном виде, отчет выполнен в полном соответствии с требованиями;

оценка «хорошо» выставляется обучающемуся, если работа выполнена в полном объеме, но присутствуют некоторые недочеты (не имеющие «критичных» последствий для последующих работ), существует преемственность выполненных заданий с отдельными незначительными отклонениями; в целом сделанные проектные предложения соответствуют современному уровню развития ИТ; работа представлена в «читабельном» виде, отчет оформлен в соответствии с требованиями с единичными нарушениями;

оценка «удовлетворительно» выставляется обучающемуся, если работа была выполнена частично, но не менее, чем на 75% (одна работа может отсутствовать), есть грубые ошибки, имеющие «критичные» последствия для последующих заданий и нарушается явная корреляция одних заданий с другими; сделанные проектные предложения не соответствуют современному уровню развития ИТ; отчет оформлен в соответствии с требованиями с единичными нарушениями;

оценка «неудовлетворительно» выставляется обучающемуся, если отчет не был представлен или работа была выполнена менее, чем на 75% (отсутствует более одной работы) и/или в работе очень много грубых ошибок (имеющие «критичных» последствий для последующих работ) и/или отсутствует какая-либо связь между заданиями; сделанные проектные предложения не соответствуют современному уровню развития ИТ; отчет не оформлен или его оформление не соответствует требованиям.

3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

ОЦЕНКА ЗНАНИЙ ОБУЧАЮЩИХСЯ НА ЭКЗАМЕНЕ

Список вопросов для подготовки к экзамену

Компетенция ПК-2 - Способен проектировать цифровой программный продукт

Индикатор ПК-2.1. Разрабатывает модель цифрового программного продукта с учетом внедрения передовых ИТ-решений прикладной сферы

Обучающийся знает: понятие архитектуры программного обеспечения, традиционные и современные архитектуры информационных систем, архитектуры web-приложения.

1. Компоненты информационных систем (пользовательский интерфейс, бизнес-логика, управление данными; подсистемы управления, функциональные подсистемы, информационные технологии).

2. Понятие архитектуры ИС и разные подходы к ее сути (идеологический, конструктивный). Значимость архитектуры и роль архитектурных решений для разработки ИС.

3. Виды архитектур (логическая, физическая; функциональная, поведенческая, временная).

4. Файл-серверные приложения: функции сервера, функции клиента, достоинства и недостатки.

5. Клиент-серверная архитектура: двухуровневая, смешанная двухуровневая, трехуровневая,

6. Сервисно-ориентированная архитектура как парадигма использования распределенных информационных ресурсов.

7. Микросервисная архитектура.

8. Архитектуры масштабируемых ИС.

9. Параллельные архитектуры ИС.

Компетенция ПК-2 - Способен проектировать цифровой программный продукт

Индикатор ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы

Обучающийся знает: понятие базы данных, распределенной базы данных облачных баз данных.

1. Преимущества и недостатки интеграции СУБД в среду web.

2. Понятие распределённых вычислений, особенности распределённых ИС, глобально распределённые ИС.

3. Internet/intranet-архитектура, internet/intranet-архитектура с мигрирующими программами,

4. Модель вычислений в облаке, структура облачных вычислений (инфраструктура, платформа, приложение).

5. «Облако» как многослойная модель SaaS (облачные приложения), PaaS (программная платформа), IaaS (виртуальные вычислительные мощности, физическое оборудование).

Компетенция ПК-2 - Способен проектировать цифровой программный продукт

Индикатор ПК-2.4. Проводит анализ современных программно-технических средств, специализированных цифровых платформ, облачных и отраслевых решений для разработки цифрового программного продукта

Обучающийся знает: методологию разработки архитектуры web-приложения и необходимые для этого программно-аппаратные средства.

1. Выявление требований (функциональных и нефункциональных) к web-приложению и их детализация в разрезе web-архитектуры.

2. Зависимости между видами требований к web-приложению, функциональности web-приложения и архитектурой web-приложения.

3. Специализированный инструментарий разработки web-архитектур (case-инструменты), традиционные нотации (DFD, IDEF, ER, EPC, ARIS, BPMN, UML, EIP).

4. Технологии и протоколы web (http, XML, JSON, REST, WebSoket).

5. Трёхзвенные архитектуры (web-ориентированные).

6. Типы серверов: www-сервер, ftpсервер, поисковый сервер.

7. Языки программирования web-приложений (обзор: Java, JavaScript, PHP и Python).

Компетенция ПК-3 - Способен разрабатывать цифровой программный продукт

Индикатор ПК-3.2. Обеспечивает интеграцию программных модулей и компонентов цифрового программного продукта на основе облачных и отраслевых ИТ-решений

Обучающийся знает: особенности web-архитектуры и основные способы реализации современных web-архитектурных решений, их достоинства и недостатки, интеграционные сложности и риски.

1. Эволюция архитектур ИС.

2. Риски реализации архитектуры и способы управления ими.

3. Преимущества и недостатки интеграции СУБД в среду web.

4. Архитектурные и проектные решения интеграции различных ИС между собой.

ОЦЕНКА УМЕНИЙ И НАВЫКОВ ОБУЧАЮЩИХСЯ НА ЭКЗАМЕНЕ

Компетенция ПК-2 - Способен проектировать цифровой программный продукт

Индикатор ПК-2.1. Разрабатывает модель цифрового программного продукта с учетом внедрения передовых ИТ-решений прикладной сферы

Обучающийся умеет: проектировать web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющихся цифровых и отраслевых решений.

Обучающийся владеет: навыками моделирования и работы с современными средствами построения и визуализации моделей web-приложения, в том числе удаленного доступа (MS Visio, BP Win, MIRO).

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения практического задания по теме 2 «Архитектуры «файл-сервер», «клиент-сервер»», по теме 3 «Облачные и распределённые вычисления», по теме 5 «Архитектура web-приложений», индивидуального проектного задания №6.1 (определение целей автоматизации и основных компонентов портала), №6.2 (формулирование требований к portalу), №6.3 (создание модели портала).

Компетенция ПК-2 - Способен проектировать цифровой программный продукт

Индикатор ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы

Обучающийся умеет: проектировать базы данных для web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющихся цифровых и отраслевых решений.

Обучающийся владеет: навыками моделирования и работы с современными средствами построения и визуализации моделей web-приложения и баз данных, в том числе удаленного доступа (MS Visio, BP Win, MIRO).

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения практического задания по теме 5 «Архитектура web-приложений», индивидуального проектного задания №6.1 (определение целей автоматизации и основных компонентов портала), №6.3 (создание модели портала).

Компетенция ПК-2 - Способен проектировать цифровой программный продукт

Индикатор ПК-2.4. Проводит анализ современных программно-технических средств, специализированных цифровых платформ, облачных и отраслевых решений для разработки цифрового программного продукта

Обучающийся умеет: осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации web-архитектуры.

Обучающийся владеет: навыками анализа и оценки цифровых платформ, облачных решений и программно-аппаратных средств.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выступления с докладом по темам 2-4, выполнения индивидуального проектного задания №6.3 (формирование программно-аппаратной платформы портала), №6.4 (определение перечня типовых / облачных решений, анализ эффективности и рисков).

Компетенция ПК-3 - Способен разрабатывать цифровой программный продукт

Индикатор ПК-3.2. Обеспечивает интеграцию программных модулей и компонентов цифрового программного продукта на основе облачных и отраслевых ИТ-решений

Обучающийся умеет: осуществлять подбор средств разработки web-приложения с учетом их интеграционных возможностей, потенциальных отказов и безопасности работы;

Обучающийся владеет навыками анализа и оценки потенциальных угроз в разработке web-приложений.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения индивидуального проектного задания №6.4 (определение перечня типовых / облачных решений, анализ эффективности и рисков).

ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

Экзаменационный билет состоит из теоретического вопроса.

Билет № 5

1. Зависимости между видами требований к web-приложению, функциональности web-приложения и архитектурой web-приложений.

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНОВАНИЯ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

ПРОЦЕДУРА ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Для контроля усвоения обучающимися данной дисциплины учебным планом предусмотрен экзамен в пятом семестре, который проводится в форме устного ответа на теоретический вопрос. При выставлении итоговой отметки за дисциплину преподаватель учитывает результаты текущего контроля (выполнения практических и индивидуальных заданий), отражающих сформированность практических навыков и умений обучающегося, а также участие обучающегося в устных опросах и выступление с докладом (который также участвует в формировании не только знаний, но и практических умений и навыков). Выполнение всех заданий текущего контроля (доклад, практические задания, индивидуальная проектная разработка) является обязательным для всех обучающихся. Обучающиеся, не выполнившие практические задания и индивидуальную проектную разработку, не допускаются к сдаче экзамена по данной учебной дисциплине.

КРИТЕРИИ ОЦЕНИВАНИЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
Компетенция ПК-2 - Способен проектировать цифровой программный продукт Индикатор ПК-2.1. Разрабатывает модель цифрового программного продукта с учетом внедрения передовых ИТ-решений прикладной сферы					
Знать: понятия архитектуры программного обеспечения, традиционных и современных архитектур информационных систем,	Не знает	Фрагментарные знания понятия архитектуры программного обеспечения, традиционных и современных архитектур информационных систем, архитектур web-	Общие, но не структурированные знания понятия архитектуры программного обеспечения, традиционных и современных архитектур информационных систем	Сформированные, но содержащие отдельные пробелы знания понятия архитектуры программного обеспечения, традиционных и современных архитектур	Сформированные систематизированные прочные знания понятия архитектуры программного обеспечения, традиционных и

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
архитектур web-приложения		приложения	нных систем, архитектур web-приложения	информационных систем, архитектур web-приложения	современных архитектур информационных систем, архитектур web-приложения
Уметь: проектировать web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющих цифровых и отраслевых решений	Не умеет	Частично освоенные умения проектировать web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющих цифровых и отраслевых решений	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения проектировать web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющих цифровых и отраслевых решений	В целом сформированные и самостоятельные выполняемые, но не всегда интеллектуально обоснованные умения проектировать web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющих цифровых и отраслевых решений	Успешно сформированные, самостоятельные и осознанно выполняемые умения проектировать web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющих цифровых и отраслевых решений
Владеть: - навыками моделирования и работы с современным и средствами построения и визуализации моделей web-	Не владеет	Фрагментарно сформированные навыки моделирования и работы с современными средствами построения и визуализации	В целом сформированные, но выполняемые на элементарном уровне навыки моделирования и работы с	Сформированные, но не устойчивые в сложных ситуациях навыки моделирования и работы с современным	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки моделирования и работы с

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
приложения, в том числе удаленного доступа (MS Visio, BP Win, MIRO)		моделей web-приложения, в том числе удаленного доступа (MS Visio, BP Win, MIRO)	современными средствами построения и визуализации моделей web-приложения, в том числе удаленного доступа (MS Visio, BP Win, MIRO)	и средствами построения и визуализации моделей web-приложения, в том числе удаленного доступа (MS Visio, BP Win, MIRO)	современным и средствами построения и визуализации моделей web-приложения, в том числе удаленного доступа (MS Visio, BP Win, MIRO)
Компетенция ПК-2 - Способен проектировать цифровой программный продукт Индикатор ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы					
Знать: понятия базы данных, распределенной базы данных, облачных баз данных	Не знает	Фрагментарные знания понятия базы данных, распределенной базы данных, облачных баз данных	Общие, но не структурированные знания понятия базы данных, распределенной базы данных, облачных баз данных	Сформированные, но содержащие отдельные пробелы знания понятия базы данных, распределенной базы данных, облачных баз данных	Сформированные систематизированные прочные знания понятия базы данных, распределенной базы данных, облачных баз данных
Уметь: проектировать базы данных для web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющихся цифровых и	Не умеет	Частично освоенные умения проектировать базы данных для web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющихся	В целом освоенные, но не подкрепляемые знаниями и самостоятельно выполнением умения проектировать базы данных для web-приложения с учетом современных тенденций развития информационных	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения проектировать базы данных для web-приложения с учетом современных тенденций развития	Успешно сформированные, самостоятельно и осознанно выполняемые умения проектировать базы данных для web-приложения с учетом современных тенденций развития информационных технологий,

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
отраслевых решений		цифровых и отраслевых решений	технологий, потребностей и возможностей заказчика, имеющихся цифровых и отраслевых решений	информационных технологий, потребностей и возможностей заказчика, имеющихся цифровых и отраслевых решений	потребностей и возможностей заказчика, имеющихся цифровых и отраслевых решений
Владеть: - навыками моделирования и работы с современным и средствами построения и визуализации моделей web-приложения и баз данных, в том числе удаленного доступа (MS Visio, BP Win, MIRO)	Не владеет	Фрагментарно сформированные навыки моделирования и работы с современными средствами построения и визуализации моделей web-приложения и баз данных, в том числе удаленного доступа (MS Visio, BP Win, MIRO)	В целом сформированные, но выполняемые на элементарном уровне навыки моделирования и работы с современными средствами построения и визуализации моделей web-приложения и баз данных, в том числе удаленного доступа (MS Visio, BP Win, MIRO)	Сформированные, но не устойчивые в сложных ситуациях навыки моделирования и работы с современным и средствами построения и визуализации моделей web-приложения и баз данных, в том числе удаленного доступа (MS Visio, BP Win, MIRO)	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки моделирования и работы с современным и средствами построения и визуализации моделей web-приложения и баз данных, в том числе удаленного доступа (MS Visio, BP Win, MIRO)
Компетенция ПК-2 - Способен проектировать цифровой программный продукт Индикатор ПК-2.4. Проводит анализ современных программно-технических средств, специализированных цифровых платформ, облачных и отраслевых решений для разработки цифрового программного продукта					
Знать: методологии разработки архитектуры web-приложения и необходимы	Не знает	Фрагментарные знания методологии разработки архитектуры web-приложения и	Общие, но не структурированные знания методологии разработки архитектуры web-приложения	Сформированные, но содержащие отдельные пробелы знания методологии разработки	Сформированные систематизированные прочные знания методологии разработки

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
е для этого программно-аппаратные средства		необходимых для этого программно-аппаратных средств	и необходимых для этого программно-аппаратных средств	архитектуры web-приложения и необходимых для этого программно-аппаратных средств	архитектуры web-приложения и необходимых для этого программно-аппаратных средств
Уметь: осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации web-архитектуры	Не умеет	Частично освоенные умения осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации web-архитектуры	В целом освоенные, но не подкрепляемые знаниями и самостоятельно выполнению умения осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации web-архитектуры	В целом сформированные и самостоятельные выполняемые, но не всегда интеллектуально обоснованные умения осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации web-архитектуры	Успешно сформированные, самостоятельные и осознанно выполняемые умения осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации web-архитектуры
Владеть: - навыками анализа и оценки цифровых платформ, облачных решений и программно-аппаратных средств	Не владеет	Фрагментарно сформированные навыки анализа и оценки цифровых платформ, облачных решений и программно-аппаратных средств	В целом сформированные, но выполняемые на элементарном уровне навыки анализа и оценки цифровых платформ, облачных решений и программно-аппаратных средств	Сформированные, но не устойчивые в сложных ситуациях навыки анализа и оценки цифровых платформ, облачных решений и программно-аппаратных средств	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки анализа и оценки цифровых платформ, облачных решений и программно-аппаратных средств

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
Компетенция ПК-3 - Способен разрабатывать цифровой программный продукт Индикатор ПК-3.2. Обеспечивает интеграцию программных модулей и компонентов цифрового программного продукта на основе облачных и отраслевых ИТ-решений					
Знать: особенности web-архитектуры и основные способы реализации современных web-архитектурных решений, их достоинства и недостатки, интеграционные сложности и риски	Не знает	Фрагментарные знания особенностей web-архитектуры, способов реализации современных web-архитектурных решений, их достоинств и недостатков, интеграционных сложностей и рисков	Общие, но не структурированные знания особенностей web-архитектуры, способов реализации современных web-архитектурных решений, их достоинств и недостатков, интеграционных сложностей и рисков	Сформированные, но содержащие отдельные пробелы знания особенностей web-архитектуры, способов реализации современных web-архитектурных решений, их достоинств и недостатков, интеграционных сложностей и рисков	Сформированные систематизированные прочные знания особенностей web-архитектуры, способов реализации современных web-архитектурных решений, их достоинств и недостатков, интеграционных сложностей и рисков
Уметь: осуществлять подбор средств разработки web-приложения с учетом их интеграционных возможностей, потенциальных отказов и безопасности работы	Не умеет	Частично освоенные умения осуществлять подбор средств разработки web-приложения с учетом их интеграционных возможностей, потенциальных отказов и безопасности работы	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения осуществлять подбор средств разработки web-приложения с учетом их интеграционных возможностей, потенциальных отказов и безопасности	В целом сформированные и самостоятельные выполняемые, но не всегда интеллектуальные умения осуществлять подбор средств разработки web-приложения с учетом их интеграционных возможностей, потенциальных отказов и безопасности	Успешно сформированные, самостоятельно и осознанно выполняемые умения осуществлять подбор средств разработки web-приложения с учетом их интеграционных возможностей, потенциальных отказов и безопасности

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
			работы	потенциальны ых отказов и безопасности работы	работы
Владеть: навыками анализа и оценки потенциаль ных угроз в разработке web- приложений;	Не владеет	Фрагментарно сформированны е навыки анализа и оценки потенциальны х угроз в разработке web- приложений	В целом сформированны е, но выполняемые на элементарном уровне навыки анализа и оценки потенциальны х угроз в разработке web- приложений	Сформирован ные, но не устойчивые в сложных ситуациях навыки анализа и оценки потенциаль ных угроз в разработке web- приложений	Успешно сформированн ые, устойчивые, технологическ и грамотно выполняемые навыки анализа и оценки потенциальн ых угроз в разработке web- приложений

КРИТЕРИИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

При выставлении итоговой оценки за дисциплину необходимо учитывать результаты текущего контроля и критерии оценивания сформированности компетенций.

Оценка **«отлично»** выставляется обучающемуся, если:

- при ответе на теоретический вопрос были продемонстрированы глубокие, систематизированные и прочные знания (в соответствии с критериями сформированности компетенций), умения логично и обоснованно излагать материал, быстро ориентироваться в дополнительных вопросах и давать полные ответы;

- индивидуальное проектное задание выполнено безошибочно и в полном соответствии с требованиями на оценку «отлично», в работе обучающийся демонстрирует понимание сути вопроса и отличное владение материалом предметной области, обосновывает проектные решения;

- обучающийся участвовал во всех видах текущего контроля и получал по ним максимальные баллы.

Все вышеперечисленное говорит о сформированности запланированных компетенций на высоком уровне.

Оценка **«хорошо»** выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано хорошее, но недостаточно полное изложение материала, с отдельными пробелами в знаниях (возможно, незначительными неточностями в употреблении терминов), логичное изложение материала, но недостаточно быстрая ориентация в нем при ответе на дополнительные вопросы;

- индивидуальное проектное задание выполнено на оценку «хорошо» (с незначительными замечаниями и/или ошибками), в работе обучающийся демонстрирует знание предметной области, но не все проектные решения может обосновать;

- обучающийся участвовал во всех видах текущего контроля и получал по ним в основном отметки «хорошо».

Все вышеперечисленное говорит о сформированности запланированных компетенций на достаточном уровне.

Оценка «**удовлетворительно**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано поверхностное владение материалом и фрагментарные знания, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, плохая ориентация в материале при ответе на дополнительные вопросы;

- индивидуальное проектное задание выполнено на оценку «удовлетворительно» (с ошибками и замечаниями), с отклонениями от требований, в работе обучающийся не демонстрирует прочные знания предметной области, не до конца разобрался в сути решаемых задач, не обосновывает проектные решения;

- обучающийся участвовал в мероприятиях текущего контроля и получал по ним, в основном, удовлетворительные отметки.

Все вышеперечисленное говорит о сформированности запланированных компетенций на низком уровне.

Оценка «**неудовлетворительно**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано незнание значительного объема материала, «фрагментарность имеющихся знаний, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, неумение выделить главное, сделать выводы и обобщения, неспособность отвечать на дополнительные вопросы;

- индивидуальное проектное задание выполнено на оценку «удовлетворительно» с грубыми ошибками и множеством замечаний, в работе обучающийся не демонстрирует знаний предметной области, не обосновывает проектные решения;

- обучающийся участвовал в мероприятиях текущего контроля, как правило, не своевременно, и получал по ним в основном удовлетворительные отметки.

Все вышеперечисленное говорит о несформированности запланированных компетенций.

5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Компетенция ПК-2 - Способен проектировать цифровой программный продукт

Обучающийся знает: понятие архитектуры программного обеспечения, традиционные и современные архитектуры информационных систем, архитектуры web-приложения; понятие базы данных, распределенной базы данных облачных баз данных; методологию разработки архитектуры web-приложения и необходимые для этого программно-аппаратные средства.

Обучающийся умеет: проектировать web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющихся цифровых и отраслевых решений; проектировать базы данных для web-приложения с учетом современных тенденций развития информационных технологий, потребностей и возможностей заказчика, имеющихся цифровых и отраслевых решений; осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации web-архитектуры.

Обучающийся владеет: навыками моделирования и работы с современными средствами построения и визуализации моделей web-приложения, в том числе удаленного доступа (MS Visio, BP Win, MIRO); навыками анализа и оценки цифровых платформ, облачных решений и программно-аппаратных средств.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА (10 вопросов)

Задание 1. Что такое архитектура программного обеспечения?

- 1) инструкция пользователя;
- 2) структура программного комплекса, включающая компоненты, их взаимосвязи и свойства;
- 3) код программы;
- 4) графический интерфейс программы.

Ответ: 2) структура программного комплекса, включающая компоненты, их взаимосвязи и свойства.

Задание 2. Какая архитектура предполагает разделение приложения на клиентскую и серверную части?

- 1) монолитная;
- 2) микросервисная;
- 3) клиент-сервер;
- 4) P2P.

Ответ: 3) клиент-сервер.

Задание 3. Что из перечисленного является примером распределенной базы данных?

- 1) MySQL;
- 2) Oracle RAC;
- 3) Microsoft Access;
- 4) SQLite.

Ответ: 2) Oracle RAC.

Задание 4. Какой методологии разработки архитектуры web-приложения не существует?

- 1) Agile;
- 2) Waterfall;
- 3) Spiral;
- 4) Randomize.

Ответ: 4) Randomize.

Задание 5. Какое утверждение о микросервисной архитектуре верно?

- 1) уменьшает сложность разработки и поддержки приложения;
- 2) подразумевает использование единственной базы данных;
- 3) ограничивает масштабирование приложения;
- 4) обеспечивает высокую степень модульности.

Ответ: 4) обеспечивает высокую степень модульности.

Задание 6. Что такое облачная база данных?

- 1) база данных, хранящаяся на локальном сервере предприятия;
- 2) база данных, доступ к которой осуществляется через интернет из облака;
- 3) база данных, использующаяся исключительно для хранения облачных конфигураций;
- 4) база данных, которая автоматически обновляется через облако.

Ответ: 2) база данных, доступ к которой осуществляется через интернет из облака.

Задание 7. Какой принцип не относится к проектированию web-приложений?

- 1) адаптивный дизайн;
- 2) минимализм в интерфейсе;
- 3) использование исключительно серверной логики;
- 4) оптимизация под мобильные устройства.

Ответ: 3) использование исключительно серверной логики.

Задание 8. Что из перечисленного является характеристикой традиционных архитектур информационных систем?

- 1) высокая гибкость и масштабируемость;
- 2) централизованное управление данными и процессами;
- 3) распределенные вычисления;
- 4) автоматическое масштабирование.

Ответ: 2) централизованное управление данными и процессами.

Задание 9. Какие сервисы являются облачной платформой для развертывания web-приложений? (множественный выбор)

- 1) AWS (Amazon Web Services);
- 2) Google Cloud Platform;
- 3) Microsoft Excel Online;
- 4) Microsoft Azure.

Ответ: 1, 2 и 4.

Вопрос 10. Для чего используются специализированные CASE-инструменты при проектировании ИС?

- 1) для создания мультимедийных объектов;
- 2) для автоматизации процесса разработки, улучшения качества и упрощения документирования;
- 3) только для создания диаграмм классов;
- 4) для общения с клиентами.

Ответ: 2) Для автоматизации процесса разработки, улучшения качества и упрощения документирования.

ЗАДАНИЯ ОТКРЫТОГО ТИПА (15 заданий)

Категория: уметь.

Задание 11. Какие информационные сети используются в корпоративных информационных сетях?

Ответ: В КИС используются различные типы информационных сетей для обеспечения коммуникации и передачи данных между различными подразделениями и структурными единицами организации. К таким сетям относятся локальные сети (LAN), которые обеспечивают связь в пределах одного здания или кампуса; глобальные сети (WAN), соединяющие расположенные на большом расстоянии друг от друга объекты; корпоративные виртуальные частные сети (VPN), позволяющие безопасно подключаться к корпоративным ресурсам из любой точки мира через интернет; а также сети передачи данных на основе технологий MPLS, обеспечивающие высокую пропускную способность и качество связи между удаленными объектами.

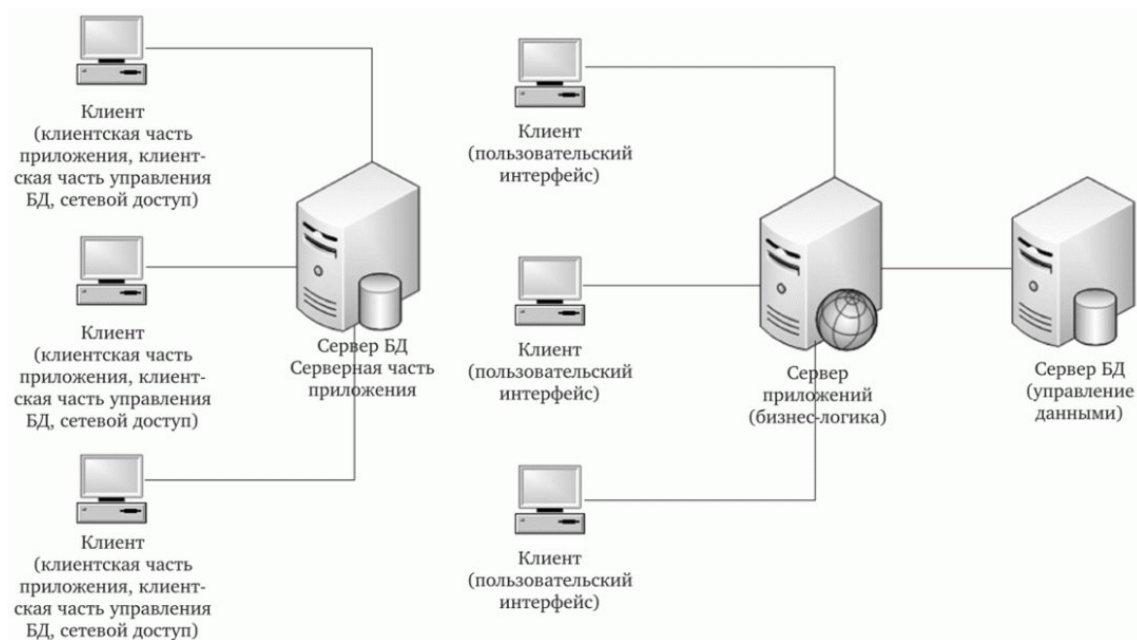
Задание 12. Какой информационной системе соответствует следующее определение: программно-аппаратный комплекс, способный объединять в одно целое предприятия с различной функциональной направленностью (производственные, торговые, кредитные и др. организации)? Пояснить.

Ответ: Описание соответствует ERP-системам (Enterprise Resource Planning) — программно-аппаратным комплексам, предназначенным для интеграции и управления основными бизнес-процессами предприятия в реальном времени. ERP-системы объединяют в одном приложении процессы планирования, закупок, продаж, бухгалтерского учета, управления персоналом, складом и доставкой. Это позволяет предприятиям различной функциональной направленности эффективно управлять своими ресурсами, повышать производительность и оптимизировать бизнес-процессы.

Задание 13. Укажите характеристики информационной системы, которые можно использовать для ее оценки и выбора.

Ответ: При оценке и выборе информационной системы учитываются следующие характеристики: 1) функциональность: способность системы выполнять необходимые задачи и процессы; 2) масштабируемость: возможность системы адаптироваться к росту объемов данных и числа пользователей; 3) надежность: стабильность работы системы и минимальное количество ошибок; 4) производительность: способность системы обрабатывать большие объемы данных с приемлемой скоростью; 5) безопасность: защита данных от неавторизованного доступа и утечек; 6) интеграция: способность системы взаимодействовать с другими приложениями и системами; 7) удобство использования: интуитивно понятный интерфейс и простота эксплуатации; 8) стоимость владения: общая стоимость приобретения, внедрения и поддержки системы.

Задание 14. Приведены две схемы архитектуры ИС. Охарактеризовать каждую из схем, пояснив различие между ними.



Ответ: На схеме приведены двухуровневая и трехуровневая архитектуры «клиент-сервер».

Двухуровневая архитектура "клиент-сервер" представляет собой простую модель, в которой клиент напрямую взаимодействует с сервером. В этой модели сервер выполняет большинство задач, включая обработку данных, хранение и управление ресурсами. Клиент же отвечает за предоставление пользовательского интерфейса и инициирование запросов к серверу.

Трехуровневая архитектура "клиент-сервер", в свою очередь, включает дополнительный уровень между клиентом и сервером, часто называемый уровнем приложений или бизнес-логики. Этот промежуточный слой отвечает за обработку запросов, выполнение бизнес-логики и обеспечение безопасности. Такое разделение позволяет лучше масштабировать систему, повышает её надежность и облегчает разработку и поддержку, поскольку изменения в одном из слоёв минимально затрагивают другие.

Главное различие между двухуровневой и трехуровневой архитектурой заключается в наличии промежуточного уровня бизнес-логики в трехуровневой модели, что делает её более гибкой, масштабируемой и безопасной по сравнению с двухуровневой моделью. Если двухуровневая модель чаще всего используется в простых или устаревших приложениях, где

нет строгих требований к масштабируемости или безопасности, то трехуровневая архитектура широко применяется в современных корпоративных приложениях и Интернет-сервисах.

Задание 15. Необходимо создать ИС для университета, которая будет обрабатывать большое количество разнородных данных: от административной информации до научных исследований. Каковы критерии выбора СУБД?

Ответ: Критерии выбора СУБД: 1) тип данных: необходимо учитывать, будут ли данные структурированными, неструктурированными или обоими; 2) производительность: способность СУБД эффективно обрабатывать большие объемы запросов в реальном времени; 3) масштабируемость: возможность расширения СУБД для поддержки растущего объема данных и пользователей; 4) безопасность: функции защиты данных от неавторизованного доступа и утечки информации; 5) стоимость: бюджет на приобретение и обслуживание СУБД, включая лицензии и техподдержку.

Задание 16. Представьте, что вы разрабатываете ИС для крупной розничной сети, которая планирует расширение на международный рынок. На этапе выбора архитектуры ИС вам необходимо учесть следующие факторы ...

Ответ: Учитываемые факторы: 1) масштабируемость: система должна поддерживать увеличение объема данных и количества пользователей без снижения производительности; 2) гибкость: возможность быстро адаптироваться к изменениям в бизнес-процессах и законодательстве разных стран; 3) надежность и доступность: обеспечение стабильной работы системы 24/7 с минимальными простоями; 4) безопасность: защита данных клиентов и компании от несанкционированного доступа; 5) интеграция: способность системы взаимодействовать с другими внутренними и внешними системами.

Задание 17. Какие инструменты для моделирования бизнес-процессов стартапа с распределенной командой можно использовать?

Ответ: Представим стартап в области EdTech, команда которого работает из разных городов и даже стран. Для моделирования бизнес-процессов подойдут: 1) MIRO: позволяет создавать доски для совместной работы над диаграммами и моделями в реальном времени; 2) Lucidchart: инструмент для создания сложных схем и диаграмм с возможностью коллаборации; 3) Microsoft Teams или Slack: для обеспечения эффективного коммуникационного канала между членами команды.

Категория: **владеть**.

Задание 18. Приведена схема архитектуры ИС. К сожалению, на схеме четко видны только комментарии. Изучив информационную часть схемы, описать, какой вид архитектуры был приведен на схеме первоначально? Аргументированно обосновать.

Группы функций программной системы:

- Функции отображения и ввода данных
Уровень представления, Presentation Layer, PrezLayer
- Прикладные функции – логика предметной области
Уровень бизнес- логики, Business Layer, BizLayer
- Функции доступа к базам данных
Уровень данных, Database Layer, DataLayer

Ответ: Исходя из предоставленной информации, можно сделать вывод, что на схеме была изображена трехуровневая архитектура ИС. Этот вывод основывается на описании трех различных групп функций программной системы, которые четко соответствуют трем основным слоям, характерным для трехуровневой архитектуры: уровень представления (Presentation Layer), уровень бизнес-логики (Business Layer) и уровень данных (Database Layer).

Задание 19. Необходимо разработать сайт предприятия. Какие действия должна выполнить команда разработчиков на первоначальном этапе, чтобы схематично представить модель предприятия? Имеются только материалы предприятия (документация).

Ответ: Необходимо выполнить следующие действия: 1) ознакомиться с информацией об объекте автоматизации; 2) создать модели основных бизнес-процессов, обеспечивающих и управленческих процессов (использовать программные средства Visio, BP Win, MIRO и др.); 3) провести декомпозицию процессов; 4) определить цели и задачи автоматизации процессов; 5) в соответствии с целями и задачами автоматизации определить основные компоненты (модули, составляющие) интернет-портала организации (не менее 4-х). Результатом разработки модели бизнес-процессов является структурная схема модели объекта.

Задание 20. Вам поручено разработать интернет-портал для вашей организации. Для этого необходимо исследовать предоставленные кейс-материалы о предприятии и использовать результаты предыдущих анализов. Ваша задача - создать четкий план разработки, учитывая структуру и потребности организации.

Ответ: В результате анализа структуры вашей организации и её потребностей, вы сформируете комплексный план по созданию интернет-портала. Этот план будет включать в себя четкое определение бизнес-, системных и функциональных требований к portalу, а также документацию по модели требований и матрицу зависимостей между ними: 1) изучить подробности об организации, чтобы понять, как она устроена и функционирует; 2) определить, какие цели и задачи должен решать интернет-портал, чтобы он служил интересам организации; 3) выявить ключевых лиц, которые будут вовлечены в проект, и организовать их в соответствующие группы для работы с порталом; 4) исходя из целей создания портала, определить, какие бизнес-требования он должен удовлетворять; 5) проанализировать, какие функциональные возможности должен предоставлять портал, чтобы соответствовать поставленным бизнес-требованиям; 6) учитывая структуру организации, определить технические требования к интернет-порталу; 7) создать документацию, отражающую все требования к portalу; 8) подготовить матрицу, которая покажет, как требования связаны между собой.

Задание 21. Разработать концепцию распределенной ИС для предметной области «Онлайн-школа».

Ответ: Создание концепции распределенной информационной системы (ИС) для онлайн-школы представляет собой задачу, требующую глубокого анализа и планирования.

1. Для разработки концепции необходимо определить ключевые функции, которые должна выполнять ИС онлайн-школы. К таким функциям относятся: регистрация и авторизация пользователей, управление профилями учеников и преподавателей, создание и редактирование курсов, проведение онлайн-уроков, организация тестирования и контроля знаний, обеспечение обратной связи и коммуникации между участниками учебного процесса.

2. Разработка архитектуры распределенной ИС. Это включает в себя выбор подходящей модели данных, технологий для фронтенда и бэкенда, а также решений для обеспечения безопасности данных и транзакций. Важным аспектом является выбор стратегии хранения данных (например, использование облачных сервисов) и подходов к масштабированию системы для обеспечения ее стабильной работы при увеличении числа пользователей.

3. Разработка пользовательского интерфейса, который должен быть интуитивно понятным и дружелюбным к пользователю всех возрастных групп. Дизайн должен

способствовать легкой навигации по курсам и материалам, а также эффективному взаимодействию между участниками процесса.

4. Обеспечение качества обучения. Для этого система должна предусматривать инструменты для создания интерактивных учебных материалов, видеоуроков и тестов. Важным является внедрение системы аналитики для отслеживания успеваемости учеников и эффективности курсов.

5. Обеспечение безопасности ИС. Необходимо реализовать механизмы защиты персональных данных пользователей и финансовых транзакций, а также предусмотреть систему резервного копирования и восстановления данных.

6. Разработка распределенной ИС для онлайн-школы требует комплексного подхода к проектированию и реализации всех компонентов системы. Это включает в себя не только технические аспекты, но и учет особенностей учебного процесса, потребностей и предпочтений пользователей. Важно предусмотреть возможность гибкой настройки и масштабирования системы для адаптации к изменяющимся условиям и требованиям образовательного процесса.

Задание 22. Обеспечение масштабируемости ИС для быстрорастущего интернет-магазина осуществляется за счет определенных решений. Какие решения можно предложить?

Ответ: Для достижения высокой масштабируемости и гибкости системы можно предложить несколько стратегических решений:

1. Применение микросервисной архитектуры. Этот подход предполагает разделение всей системы на небольшие, независимо разрабатываемые и развертываемые сервисы. Каждый микросервис отвечает за определенную функциональность и может быть масштабирован независимо от остальных. Это позволяет не только упростить разработку и поддержку сложных систем, но и обеспечить более гибкое масштабирование в ответ на изменяющиеся нагрузки, что особенно важно для быстрорастущего интернет-магазина.

2. Использование облачных сервисов, таких как AWS (Amazon Web Services), Google Cloud Platform или Microsoft Azure. Облачные платформы предоставляют широкий спектр инфраструктурных решений и сервисов, которые можно адаптировать под конкретные нужды бизнеса. Благодаря возможности гибкого управления ресурсами и оптимизации затрат, облачные сервисы позволяют легко масштабировать ИС в соответствии с текущими требованиями, а также автоматически адаптироваться к пиковым нагрузкам, обеспечивая высокую доступность и производительность системы.

3. Контейнеризация при помощи таких инструментов, как Docker и Kubernetes. Контейнеризация позволяет упаковывать приложение вместе со всем необходимым окружением, что упрощает процесс развертывания и гарантирует его консистентность на любой платформе. Kubernetes предоставляет мощные средства для оркестрации контейнеров, автоматического масштабирования и управления жизненным циклом приложений. Это обеспечивает высокую отказоустойчивость, упрощает процесс развертывания новых версий приложений и позволяет эффективно масштабировать систему в зависимости от текущих потребностей.

Необходимо учитывать специфику бизнес-процессов интернет-магазина, ожидаемые объемы трафика и данные о поведении пользователей. Также крайне важно обеспечить безопасность данных и транзакций в условиях распределенной системы.

Задание 23. Какими принципами и механизмами должна руководствоваться крупная финансовая организация при интеграции современной CRM-системы с устаревшей базой данных клиентов?

Ответ: Для успешной реализации такого проекта важно обратить внимание на следующие аспекты:

1. Принцип SOA (Service-Oriented Architecture, или архитектура, ориентированная на сервисы) для создания адаптеров или прокси-сервисов. Эти компоненты служат посредниками, обеспечивающими взаимодействие между новой CRM-системой и устаревшей базой данных через API (Application Programming Interface). Применение SOA позволяет обеспечить гибкость

интеграции, поскольку адаптеры и прокси-сервисы могут быть разработаны таким образом, чтобы скрыть сложность и различия в структурах данных устаревшей системы от новой CRM. Это облегчает процесс интеграции и повышает его надежность.

2. Механизм ETL (Extract, Transform, Load), который используется для миграции и трансформации данных между системами. Этап "Extract" включает извлечение данных из устаревшей системы. На этапе "Transform" происходит преобразование данных в формат, совместимый с новой CRM-системой. Это может включать очистку данных, их нормализацию, агрегацию и другие виды трансформации. Наконец, этап "Load" подразумевает загрузку преобразованных данных в целевую систему. Применение ETL позволяет не только перенести данные из одной системы в другую, но и значительно улучшить их качество, что положительно сказывается на эффективности работы новой CRM.

Эти подходы обеспечивают гибкость, безопасность и эффективность процесса интеграции, что крайне важно для крупных финансовых организаций.

Задание 24. Компания, специализирующаяся на разработке мобильных приложений, сталкивается с необходимостью обработки больших объемов неструктурированных данных, включая логи пользовательских сессий, сообщения об ошибках и данные о взаимодействии с интерфейсом. Традиционные реляционные базы данных уже не справляются с нагрузкой из-за высоких требований к масштабируемости и гибкости структуры данных. В связи с этим компания рассматривает возможность перехода на NoSQL-решение. Какие преимущества и недостатки использования NoSQL баз данных в данной ситуации?

Ответ: В ситуации, когда компания сталкивается с необходимостью обработки больших объемов неструктурированных данных и традиционные реляционные базы данных не справляются с задачей, использование NoSQL может предложить ряд значительных преимуществ. NoSQL базы данных спроектированы для работы с большими объемами данных, предлагая высокую масштабируемость и гибкость в управлении структурой данных, что делает их идеальными для сценариев с неструктурированными или полуструктурированными данными.

Преимущества использования NoSQL баз данных включают:

1. Масштабируемость: NoSQL базы данных легко масштабируются горизонтально, что позволяет добавлять дополнительные сервера для обработки возрастающих объемов данных без значительных изменений в архитектуре.

2. Гибкость структуры: Они предлагают гибкую схему данных, что позволяет легко добавлять и изменять структуру данных без необходимости заранее определять все поля и типы данных.

3. Высокая производительность: Благодаря оптимизации под конкретные типы запросов и хранения данных, NoSQL базы данных часто обеспечивают более высокую производительность для определенных операций, особенно при работе с большими объемами неструктурированных данных.

4. Надежность и доступность: Многие NoSQL системы предлагают продвинутые возможности по обеспечению высокой доступности и устойчивости к отказам через репликацию и распределение данных.

Использование NoSQL баз данных представляет собой мощный инструмент для компаний, сталкивающихся с вызовами обработки больших объемов неструктурированных данных.

Задание 25. Перед руководителем команды поставлена задача моделирования бизнес-процессов предприятия. Какую нотацию должен использовать разработчик, чтобы представить руководителю бизнес-процессы предприятия? Аргументированно описать, почему все остальные методологии не подходят.

- 1) IDEF0;
- 2) BPMN;
- 3) UML;

4) ERD.

Ответ: Для моделирования бизнес-процессов предприятия и представления их руководителю наиболее подходящей нотацией является BPMN (Business Process Model and Notation). BPMN специально разработана для описания и моделирования бизнес-процессов. Эта нотация позволяет в графическом виде представить шаги процесса, участников, события и информационные потоки. Одно из ключевых преимуществ BPMN заключается в её понятности и наглядности для людей, не обладающих специальными техническими знаниями, что делает её идеальным выбором для демонстрации бизнес-процессов руководителю.

IDEF0 - это методология, используемая для моделирования функций системы и их взаимосвязей. Хотя IDEF0 может быть полезна для анализа процессов на высоком уровне, она менее эффективна для детального моделирования бизнес-процессов, особенно когда требуется демонстрация последовательности действий и взаимодействия между различными элементами процесса. IDEF0 больше сосредоточена на функциональном аспекте, чем на процессном, что делает её менее подходящей для задачи представления бизнес-процессов руководителю.

UML (Unified Modeling Language) - это мощный инструмент для моделирования программного обеспечения, который также может использоваться для описания бизнес-процессов через диаграммы деятельности. Однако UML является более технической и комплексной нотацией, что может затруднить понимание для неподготовленных лиц, таких как руководители, не имеющие специализированных знаний в области информационных технологий. UML предназначена для широкого спектра задач моделирования и не специализируется исключительно на бизнес-процессах, что делает её менее предпочтительным выбором для данной задачи.

ERD (Entity-Relationship Diagram) - это тип диаграммы, используемой для моделирования структур данных и отношений между ними в базах данных. ERD сосредоточена на представлении сущностей и связей между ними и не предназначена для моделирования бизнес-процессов. Использование ERD для демонстрации бизнес-процессов руководителю было бы неэффективным, поскольку эта нотация не способна адекватно отразить последовательность операций, условия выполнения процессов и взаимодействие между различными участниками процесса.

Таким образом, BPMN является наиболее подходящей нотацией для моделирования бизнес-процессов предприятия и их представления руководителю благодаря своей специализации, наглядности и понятности. В то время как другие методологии, такие как IDEF0, UML и ERD, могут быть полезны в определенных контекстах, они не обладают такой же специализацией и простотой восприятия для целей демонстрации бизнес-процессов.

Компетенция ПК-3 - Способен разрабатывать цифровой программный продукт

Обучающийся знает: особенности web-архитектуры и основные способы реализации современных web-архитектурных решений, их достоинства и недостатки, интеграционные сложности и риски.

Обучающийся умеет: осуществлять подбор средств разработки web-приложения с учетом их интеграционных возможностей, потенциальных отказов и безопасности работы.

Обучающийся владеет: навыками анализа и оценки потенциальных угроз в разработке web-приложений.

ЗАДАНИЯ ОТКРЫТОГО ТИПА (10 вопросов)

Задание 1. Что такое клиент-серверная архитектура?

- 1) архитектура, где обработка данных происходит только на стороне клиента;
- 2) архитектура, где обработка данных происходит только на стороне сервера;
- 3) архитектура, в которой задачи разделены между поставщиками ресурсов (серверами) и запросившими их клиентами;
- 4) архитектура, не предполагающая обмен данных между компонентами.

Ответ: 3) архитектура, в которой задачи разделены между поставщиками ресурсов (серверами) и запросившими их клиентами.

Задание 2. Какая модель архитектуры предполагает разделение приложения на клиентский интерфейс, сервер приложений и базу данных?

- 1) одноуровневая архитектура;
- 2) двухуровневая архитектура;
- 3) трехуровневая архитектура;
- 4) многоуровневая архитектура.

Ответ: 3) трехуровневая архитектура.

Задание 3. Какое из следующих утверждений верно относительно микросервисной архитектуры?

- 1) микросервисы увеличивают зависимость между компонентами системы;
- 2) микросервисы облегчают масштабирование и обновление отдельных частей системы;
- 3) микросервисы снижают гибкость развертывания отдельных частей системы;
- 4) микросервисы упрощают управление состоянием приложения.

Ответ: 2) микросервисы облегчают масштабирование и обновление отдельных частей системы.

Задание 4. Какой из недостатков характерен для монолитной архитектуры?

- 1) высокая гибкость в развертывании новых функций;
- 2) простота в добавлении новых технологий;
- 3) упрощенное управление состоянием;
- 4) сложности с масштабированием отдельных компонентов приложения.

Ответ: 4) сложности с масштабированием отдельных компонентов приложения.

Задание 5. Какой метод HTTP используется для создания нового ресурса?

- 1) GET;
- 2) POST;
- 3) PUT;
- 4) DELETE.

Ответ: 2) POST.

Задание 6. Какой паттерн позволяет различным системам обмениваться данными в реальном времени?

- 1) пакетная обработка;
- 2) запрос-ответ;
- 3) публикация-подписка;
- 4) транзакционная интеграция.

Ответ: 3) публикация-подписка.

Задание 7. Что является основным преимуществом использования WebSockets?

- 1) поддержка старых браузеров;
- 2) двусторонняя связь в реальном времени;
- 3) автоматическое шифрование данных;
- 4) уменьшение нагрузки на сервер.

Ответ: 1) двусторонняя связь в реальном времени.

Задание 8. Как CDN (сеть доставки контента) ускоряет загрузку веб-страниц?

- 1) распределяя контент по серверам, расположенным ближе к пользователю;
- 2) уменьшая размер файлов на сервере;
- 3) увеличивая пропускную способность интернета пользователя;

4) автоматически сжимая контент перед отправкой.

Ответ: 1) распределяя контент по серверам, расположенным ближе к пользователю.

Задание 9. Какая мера является эффективной для предотвращения XSS (Cross-Site Scripting)?

- 1) использование HTTPS;
- 2) экранирование пользовательского ввода;
- 3) регулярное обновление серверного программного обеспечения;
- 4) использование сложных паролей.

Ответ: 2) экранирование пользовательского ввода.

Задание 10. Для чего в Web-приложениях используются куки?

- 1) для хранения больших объемов данных на стороне клиента;
- 2) для управления сессиями пользователя и хранения небольших объемов данных;
- 3) для полного замещения баз данных;
- 4) исключительно для отслеживания поведения пользователя на сайте.

Ответ: 2) для управления сессиями пользователя и хранения небольших объемов данных.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА (15 вопросов)

Задание 11. Сопоставьте протоколы с их основным использованием в разработке web-приложений.

- | | |
|--------------|--|
| 1) HTTPS | А) Шифрование передаваемых данных |
| 2) WebSocket | Б) Построение API |
| 3) REST | В) Поддержка двусторонней связи в реальном времени |

Ответ:

- 1 - А) Шифрование передаваемых данных.
- 2 - В) Поддержка двусторонней связи в реальном времени.
- 3 - Б) Построение API.

Задание 12. Распределите следующие типы баз данных по их наилучшему применению.

- | | |
|-------------------------------|---|
| 1) Документно-ориентированная | А) Хранение сложных иерархических данных |
| 2) Реляционная | Б) Управление большими объемами неструктурированных данных |
| 3) Графовая | В) Обработка транзакций с жесткими требованиями к согласованности |

Ответ:

- 1 - Б) Управление большими объемами неструктурированных данных.
- 2 - В) Обработка транзакций с жесткими требованиями к согласованности.
- 3 - А) Хранение сложных иерархических данных.

Задание 13. Как использование CDN способствует повышению производительности web-приложений?

Ответ: CDN (Content Delivery Network) ускоряет загрузку статического контента (изображений, CSS и JavaScript файлов) путем кэширования его на серверах, расположенных ближе к пользователю. Это снижает время отклика, уменьшает нагрузку на основной сервер приложения и повышает общую производительность и скорость работы web-приложений. Также CDN может помочь защитить приложения от DDoS-атак и обеспечить более высокую доступность контента.

Задание 14. Какие методы аутентификации пользователей наиболее распространены в web-приложениях?

Ответ: Наиболее распространенными методами аутентификации в web-приложениях являются: Basic Authentication, где имя пользователя и пароль передаются через HTTP-заголовок; форма входа с сохранением сессии пользователя; JWT (JSON Web Tokens), предоставляющий токены для безопасного обмена информацией между клиентом и сервером; OAuth для аутентификации через сторонние сервисы (например, социальные сети). Выбор метода зависит от требований к безопасности и удобству пользователя.

Задание 15. Какие критерии следует учитывать при выборе фреймворка для разработки web-приложения?

Ответ: При выборе фреймворка для разработки web-приложения важно учитывать следующие критерии: 1) поддержку языка программирования, который вы хотите использовать; 2) интеграционные возможности с другими инструментами и сервисами; 3) наличие встроенных механизмов для обеспечения безопасности приложения; 4) способность фреймворка обрабатывать потенциальные отказы и нагрузку; 5) активность и размер сообщества, которое может предоставить поддержку и документацию. Также важно оценить, насколько легко фреймворк позволяет масштабировать приложение в будущем.

Задание 16. Какие методы защиты от SQL-инъекций являются наиболее эффективными?

Ответ: Для защиты от SQL-инъекций наиболее эффективными методами являются 1) использование параметризованных запросов, которые не позволяют исполнение произвольного SQL-кода; 2) очистка и проверка входящих данных (input validation) для удаления или замены потенциально опасных символов; 3) использование ORM (Object-Relational Mapping) библиотек, которые автоматически применяют эти практики. Эти методы помогают минимизировать риск выполнения нежелательных SQL-операций, которые могут привести к утечке или изменению данных.

Задание 17. К какому понятию относятся сразу несколько понятий: «защита», «шифрование», «многоуровневость» и «аутентификация»? Построить определение данного понятия и пояснить.

Ответ: Эти понятия относятся к концепции *многослойной защиты* в контексте информационной безопасности. Вот как данные понятия вписываются в эту концепцию:

1. Защита. Это общее понятие, которое означает применение мер и технологий для обеспечения безопасности информации и систем от различных угроз, таких как вирусы, хакерские атаки, фишинг и другие.

2. Шифрование. Это метод превращения информации в форму, которая не может быть прочитана без соответствующего ключа. Шифрование является одним из слоёв защиты в стратегии обороны на глубине, поскольку даже если данные будут перехвачены, без ключа они останутся недоступными для злоумышленников.

3. Многоуровневость. Это принцип организации или структурирования чего-либо, который предполагает разделение на несколько уровней, слоёв или этапов, каждый из которых имеет свои задачи, функции или характеристики.

4. Аутентификация. Это процесс проверки подлинности пользователя или устройства, пытающегося получить доступ к ресурсам системы. Аутентификация является ещё одним важным слоем защиты, поскольку она помогает гарантировать, что доступ к информации и ресурсам предоставляется только авторизованным пользователям.

Таким образом, *многослойная защита* – использование *многоуровневой защиты*, включая фильтрацию входных данных, *шифрование*, *аутентификацию* и авторизацию.

Категория: **владеть**.

Задание 18. Как можно обеспечить безопасность управления сессиями в вашем веб-приложении?

Ответ: Для безопасного управления сессиями необходимо генерировать уникальные идентификаторы сессий с использованием надежных криптографических методов. Важно также ограничивать время жизни сессии, автоматически завершая её по истечении определённого времени неактивности. Использование флагов безопасности для кук, таких как HttpOnly и Secure, помогает предотвратить доступ к идентификаторам сессий через скрипты и при передаче данных по незащищённым каналам соответственно. Регулярное обновление механизмов аутентификации и контроль за актуальностью используемых библиотек и фреймворков повышают безопасность управления сессиями.

Задание 19. Вы хотите убедиться, что ваше веб-приложение защищено от CSRF (Cross-Site Request Forgery). Какие шаги вы предпримете?

Ответ: Для защиты от CSRF важно требовать наличие токена безопасности при каждом выполнении действий, изменяющих состояние на сервере. Этот токен должен быть уникальным для каждой сессии пользователя и проверяться сервером при каждом запросе. Использование политики SameSite для кук может также помочь предотвратить CSRF, поскольку оно ограничивает отправку кук при запросах с других сайтов. Включение CAPTCHA для критически важных действий может быть ещё одним слоем защиты, хотя это может повлиять на удобство использования. Регулярное обновление и тестирование безопасности вашего веб-приложения также являются ключевыми элементами защиты от CSRF и других угроз.

Задание 20. Какие меры вы примете для защиты API вашего веб-приложения?

Ответ: Меры: 1) Для защиты API ключевым является использование аутентификации и авторизации запросов, например, через OAuth 2.0 или JWT (JSON Web Tokens), чтобы контролировать доступ к API. 2) Важно также ограничивать количество запросов от одного пользователя или IP-адреса для предотвращения DDoS-атак и злоупотреблений. 3) Использование HTTPS для шифрования данных передаваемых через API предотвращает перехват информации. 4) Разработка системы логирования и мониторинга поможет своевременно выявлять подозрительную активность и реагировать на инциденты безопасности. 5) Регулярное проведение аудитов безопасности API и обновление зависимостей повышает уровень защиты от известных уязвимостей.

Задание 21. Проектирование ИС для сферы услуг. Какие требования к мобильности можно сформулировать при проектировании ИС?

Ответ: При проектировании информационной системы (ИС) для сферы услуг, особое внимание следует уделить требованиям к мобильности. Это обусловлено тем, что пользователи ожидают доступности и удобства использования сервиса в любое время и в любом месте. Рассмотрим ключевые требования к мобильности, которые необходимо учесть при проектировании такой системы.

1. Адаптивный дизайн: ИС должна корректно отображаться на различных устройствах, включая смартфоны, планшеты и ноутбуки.

2. Офлайн-режим: Важно предусмотреть возможность работы приложения в офлайн-режиме с последующей синхронизацией данных при восстановлении интернет-соединения.

3. Оптимизация производительности: приложение должно быстро загружаться и обрабатывать большое количество информации, минимизируя время ожидания пользователем.

4. Безопасность: обеспечение безопасности передаваемых данных является критически важным.

5. Геолокационные сервисы: интеграция геолокационных сервисов позволяет предоставлять пользователям персонализированные предложения, основанные на их текущем местоположении, а также направлять их к ближайшим точкам обслуживания.

6. Уведомления: система должна поддерживать отправку push-уведомлений для информирования пользователей о статусе их заказов, специальных предложениях и других важных событиях.

7. Мультиязычность: поддержка нескольких языков делает ИС доступной для широкого круга пользователей из разных стран и культур.

8. Интеграция с социальными сетями: возможность входа через социальные сети и делиться информацией об услугах в социальных сетях может значительно увеличить охват потенциальной аудитории.

9. Доступность: ИС должна быть доступна для пользователей с ограниченными возможностями, следуя принципам создания доступного контента.

10. Обратная связь: предоставление простого и интуитивно понятного способа для отправки отзывов и предложений помогает улучшить качество сервиса и повысить удовлетворенность пользователей.

Разработка мобильной версии ИС для сферы услуг требует комплексного подхода к удовлетворению потребностей пользователей. Учет вышеупомянутых требований позволит создать удобное, функциональное и безопасное приложение, способное привлечь и удержать целевую аудиторию.

Задание 22. При разработке веб-приложения вы столкнулись с необходимостью защитить его от XSS (Cross-Site Scripting) атак. Какой метод или технологию следует выбрать для эффективной защиты? Также аргументировать неправильные ответы.

Варианты:

- 1) использование Content Security Policy (CSP);
- 2) хранение данных в открытом виде;
- 3) использование GET-запросов для передачи чувствительной информации.

Ответ: 1) использование Content Security Policy (CSP).

Аргументация правильного ответа: Content Security Policy (CSP) является мощным инструментом для защиты веб-приложений от XSS-атак. CSP позволяет веб-разработчикам контролировать ресурсы, которые могут быть загружены и выполнены на странице, тем самым предотвращая выполнение злонамеренного кода. С помощью CSP можно определить, откуда разрешено загружать скрипты, стили, картинки и другие ресурсы, что значительно снижает риск XSS-атак.

Аргументация неправильных ответов:

1. Хранение данных в открытом виде (2) не является мерой защиты от XSS-атак. На самом деле, это может увеличить риск утечки данных, если злоумышленник сможет внедрить и выполнить скрипт на странице. XSS-атака не связана напрямую с тем, как хранятся данные, а с тем, как данные выводятся на страницу и могут ли они быть интерпретированы как часть исполняемого кода.

2. Использование GET-запросов для передачи чувствительной информации (3) также не защищает от XSS-атак. Метод передачи данных (GET или POST) не влияет на уязвимость к XSS. Проблема XSS заключается во внедрении злоумышленником кода в веб-страницу, который будет выполнен в браузере пользователя. Метод передачи данных не предотвращает возможность такого внедрения; важно правильно обрабатывать и экранировать пользовательский ввод, чтобы предотвратить выполнение произвольного JavaScript-кода.

Задание 23. Какие методы тестирования веб-приложений существуют?

Ответ: Существует несколько основных методов тестирования:

1. Функциональное тестирование - проверка соответствия функционала веб-приложения его техническому заданию и требованиям. Этот метод включает в себя тестирование отдельных функций, модулей, а также интеграцию компонентов приложения.

2. Тестирование производительности - определение скорости работы веб-приложения, его способности обрабатывать определенное количество запросов за единицу времени, а также его масштабируемость.

3. Тестирование безопасности - выявление уязвимостей веб-приложения перед атаками, такими как SQL инъекции, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery) и другие. Это направлено на защиту данных пользователей и системы в целом.

4. Тестирование совместимости - проверка корректности отображения и функционирования веб-приложения в различных браузерах, операционных системах и на различных устройствах.

5. Тестирование юзабилити (удобства использования) - оценка удобства интерфейса для конечного пользователя, логичности и интуитивности навигации, доступности ключевых функций.

6. Регрессионное тестирование - проводится после изменений в коде приложения для проверки, не привели ли эти изменения к появлению новых ошибок или к возникновению проблем в уже протестированных частях программы.

В зависимости от специфики проекта и стадии разработки выбираются наиболее подходящие методы тестирования.

Задание 24. Для защиты данных, передаваемых между клиентом и сервером вашего веб-приложения, необходимо выбрать подходящий метод шифрования. Какой метод вы бы использовали? Также аргументировать неправильные ответы.

Варианты:

- 1) шифрование данных алгоритмом MD5;
- 2) использование протокола HTTPS с TLS;
- 3) передача данных через незащищённый HTTP;
- 4) шифрование каждого сообщения уникальным ключом.

Ответ: 2) Использование протокола HTTPS с TLS

Аргументация правильного ответа: Использование протокола HTTPS с TLS - правильный ответ, так как HTTPS (Hypertext Transfer Protocol Secure) использует TLS (Transport Layer Security) для шифрования данных, передаваемых между клиентом и сервером. Это обеспечивает конфиденциальность и целостность данных, предотвращая перехват и модификацию информации злоумышленниками.

Аргументация неправильных ответов:

1. Шифрование данных алгоритмом MD5 - неправильный ответ, потому что MD5 является алгоритмом хеширования, а не шифрования. Хеширование - это односторонний процесс, который генерирует фиксированный размер хеша из данных любого размера. Хеш не может быть обратно преобразован в исходные данные, что делает MD5 не подходящим для шифрования данных, которые должны быть расшифрованы на другой стороне.

2. Передача данных через незащищённый HTTP - неправильный ответ, поскольку HTTP без добавления защиты TLS является незащищённым протоколом, который не обеспечивает шифрование данных. Это означает, что все передаваемые данные могут быть легко перехвачены или изменены третьими лицами.

3. Шифрование каждого сообщения уникальным ключом - хотя этот метод теоретически может обеспечить высокий уровень безопасности, он не даёт конкретного решения или стандарта для реализации. Кроме того, управление уникальными ключами для каждого сообщения может быть сложным и непрактичным в большинстве случаев. Использование HTTPS с TLS является более стандартизированным и проверенным временем способом защиты данных.

Таким образом, использование протокола HTTPS с TLS является наилучшим выбором для защиты данных, передаваемых между клиентом и сервером веб-приложения, поскольку он обеспечивает эффективное шифрование данных и защиту от многих видов атак.

Задание 25. Существуют ли какие-то особые меры защиты для веб-приложений, работающих с облачными сервисами?

Ответ: Использование комплексного подхода, сочетающего технологические, организационные и образовательные меры:

1. Использование HTTPS и TLS: Обеспечивает шифрование данных, передаваемых между клиентом и сервером, защищая конфиденциальность и целостность данных.
 2. Аутентификация и авторизация: Применение сложных политик доступа через сервисы управления идентификацией (IAM) для контроля доступа к ресурсам.
 3. Обновление ПО и патчи безопасности: Регулярное обновление программного обеспечения для защиты от эксплуатации известных уязвимостей.
 4. Использование VPN и сетевых экранов (firewalls): Контроль трафика между веб-приложением и облачными сервисами для предотвращения неавторизованного доступа.
 5. Шифрование данных на сервере: Защита хранимой информации, включая базы данных и файлы, с помощью различных опций шифрования.
 6. Аудит безопасности и мониторинг: Регулярное проведение аудита и мониторинг системы на предмет подозрительной активности с использованием инструментов логирования.
 7. Защита от DDoS-атак: Применение специализированных сервисов для фильтрации трафика и защиты приложения от перегрузки.
 8. Резервное копирование и стратегия восстановления: Обеспечение сохранности данных и разработка плана действий на случай атаки или сбоя.
 9. Обучение персонала: Повышение осведомленности и компетенции сотрудников в области кибербезопасности для минимизации рисков, связанных с человеческим фактором.
- Эти меры помогут обеспечить комплексную защиту веб-приложений, использующих облачные сервисы, от различных угроз безопасности.

6. ЛИСТ СОГЛАСОВАНИЯ

Составил:

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий выпускающей кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)