

**Комплект оценочных материалов по дисциплине**  
**«Современные облачные хранилища данных»**

**Задания закрытого типа**

**Задания закрытого типа на выбор правильного ответа**

*Выберите один правильный ответ*

1. Укажите наиболее популярный вид системы сборки для Spring приложений

- A) Gradle
- Б) Webpack
- В) Maven
- Г) Ant

Правильный ответ: В

Компетенции (индикаторы): ПК-01(ПК-01.1).

*Выберите несколько правильных ответов*

2. Укажите возможные источники хранения описания бинов во фреймворке Spring

- А) Файл application.properties
- Б) Java аннотации
- В) YAML файл
- Г) Файл xml

Правильный ответ: А, Б, Г

Компетенции (индикаторы): ПК-01(ПК-01.2).

3. Укажите корректные способы включения бина в контекст фреймворка Spring

- А) посредством аннотации @Bean;
- Б) посредством директивы #include
- В) посредством стереотипных аннотаций;
- Г) посредством директивы #define
- Д) программно (метод getBean())

Правильный ответ: А, В, Д.

Компетенции (индикаторы): ПК-01(ПК-01.3).

4. Перечислите свойства Java класса, чтобы он мог быть отнесен к JavaBeans

- А) Класс должен реализовать методы рефлексии
- Б) Класс должен иметь конструктор без параметров, с модификатором доступа public.
- В) Свойства класса должны быть доступны через get-еры, set-еры.
- Г) Класс должен быть сериализуем.

Д) Класс должен иметь переопределенные методы equals(), hashCode() и toString().

Правильный ответ: Б, В, Г, Д.

Компетенции (индикаторы): ПК-01(ПК-01.1).

### **Задания закрытого типа на установление соответствие**

*Установите правильное соответствие.*

*Каждому элементу левого столбца соответствует только один элемент правого столбца.*

1. Укажите соответствие описания области видимости Spring бина и ее краткого названия

- |    |   |                |
|----|---|----------------|
| 1) | Возвращает один и тот же экземпляр бина на каждый запрос контейнера Spring IoC.   | A) request     |
| 2) | Возвращает один и тот же экземпляр бина на каждый запрос контейнера Spring IoC.   | Б) session     |
| 3) | Возвращает один и тот же экземпляр бина на каждый запрос контейнера Spring IoC.   | В) singleton   |
| 4) | Создает и возвращает экземпляр бина для каждой HTTP сессии  | Г) prototype   |
| 5) | Бин с данной областью видимости создается один раз для всего сервлет-контекста. Это полезно для данных, которые должны быть общими для всех пользователей и сессий в приложении | Д) Application |

Правильный ответ:

1	2	3	4	5
В	Г	А	Б	Д

Компетенции (индикаторы): ПК-01(ПК-01.2).

2. Установите соответствие описания слоев информационной системы и их названий

- |    |  |                          |
|----|--|--------------------------|
| 1) | Всё, что связано с взаимодействием с пользователем: нажатие кнопок, движение мыши, отрисовка изображения, вывод результатов поиска и т.д | A) Слой бизнес-логики    |
| 2) | Правила, алгоритмы реакции приложения на действия пользователя или на внутренние события, правила обработки данных                       | Б) Слой доступа к данным |

- 3) Хранение, выборка, модификация и удаление данных, связанных с решаемой приложением прикладной задачей

Правильный ответ:

1	2	3
В	А	Б

Компетенции (индикаторы): ПК-01(ПК-01.3).

3. Установите соответствие описания видов архитектур информационных систем и их названий

- |  |                     |
|--|---------------------|
| 1) Распределённая архитектура, когда слои представления и бизнес-логики находятся на клиенте, а также там реализуется часть вычислений.<br>Сервер отвечает только за хранение и управление файлами           | А) Монолитная       |
| 2) Обеспечивает многопользовательскую работу с данными, имеет более высокую надёжность, так как на клиенте находится лишь слой представления и часть слоя бизнес-логики                                      | Б) Файл-серверная   |
| 3) Если все функциональные возможности ИС реализованы в виде одного, а не нескольких серверных компонентов, и поддержаны единой базой данных.  | В) Микросервисная   |
| 4) Если весь набор функциональных возможностей ИС представлен не одним, а несколькими backend'ами, которые взаимодействуют между собой и каждый из них имеет свою ограниченную по контексту область действия | Г) Клиент-серверная |

Правильный ответ:

1	2	3	4
Б	Г	А	В

Компетенции (индикаторы): ПК-01(ПК-01.1)

4. Установите соответствие описания видов протоколов взаимодействия микросервисов и примеров таких протоколов

- |   |         |
|---|---------|
| 1) Асинхронный протокол. Код клиента или отправители сообщения не ждут ответ — сообщения отправляются | А) HTTP |
|---|---------|

аналогично передаче в очередь любого брокера. Такой тип применяет протокол

- 2) Синхронный протокол. При отправке запроса клиент ожидает ответ от службы — задачи выполняются только после того, как сервер пришлет ответ.

Правильный ответ:

1	2
Б	А

Компетенции (индикаторы): ПК-01(ПК-01.2).

**Задания закрытого типа на установление правильной последовательности**

*Установите правильную последовательность. Запишите правильную последовательность букв слева направо.*

1. Расположите этапы жизненного цикла String бина в хронологическом порядке

- А) удаление бина;
- Б) Пре-инициализация ;
- В) Установка свойств из конфигурации бина;
- Г) Инициализация:
- Д) Пост-инициализация5;
- Е) Инстанцирование объекта.

Правильный ответ: Е, В, Б, Г, Д, А

Компетенции (индикаторы): ПК-01(ПК-01.3).

2. Расположите в правильном порядке составные части HTTP запроса.

- А) заголовок запроса (header fields) — характеризуют тело сообщения,
- Б) тело сообщения (body)
- В) строка запроса (request line) — определяет метод, путь к ресурсу и версию протокола

Правильный ответ: В, А, Б.

Компетенции (индикаторы): ПК-01(ПК-01.1).

3. Расположите в правильном порядке составные части HTTP ответа.

- А) тело сообщения (body)
- Б) строка статуса ответа (status line) — код ответа, версия протокола
- В) заголовки ответа (header fields) — характеризуют тело ответа,

Правильный ответ: Б, В, А.

Компетенции (индикаторы): ПК-01(ПК-01.2).

4. Расположите в правильном порядке составные части URL.

- А) host
- Б) порт
- В) параметры
- Г) протокол
- Д) path
- Е) ?

Правильный ответ: Г, А, Б, Д, Е, В.

Компетенции (индикаторы): ПК-01(ПК-01.2).

### **Задания открытого типа на дополнение**

*Напишите пропущенное слово (словосочетание).*

1. Файл описания проекта в Maven называется \_\_\_\_\_

Правильный ответ: pom.xml

Компетенции (индикаторы): ПК-01 (ПК-01.1, ПК-01.2, ПК-01.3).

2. Бины фреймворк Spring создает в \_\_\_\_\_ - специальном пространстве имен для управления бинами.

Правильный ответ: контексте

Компетенции (индикаторы): ПК-01 (ПК-01.3).

3. REST архитектура полностью построена на основе \_\_\_\_\_ протокола

Правильный ответ: HTTP

Компетенции (индикаторы): ПК-01 (ПК-01.3).

4. Сервис \_\_\_\_\_ используется для автоматизации документирования REST API сервисов

Правильный ответ: swagger

Компетенции (индикаторы): ПК-01 (ПК-01.1).

5. Фреймворк Spring MVC для разработки веб-приложений использует архитектурный паттерн \_\_\_\_\_.

Правильный ответ: MVC

Компетенции (индикаторы): ПК-01 (ПК-01.3).

5. Компонент сборки проекта в Maven называется \_\_\_\_\_.

Правильный ответ: плагин.

Компетенции (индикаторы): ПК-01 (ПК-01.3).

### **Задания открытого типа с кратким свободным ответом**

*Напишите пропущенное слово (словосочетание).*

1. Java класс, содержащий описание конфигурации Spring должен быть помечен аннотацией \_\_\_\_\_

Правильный ответ: @configuration / configuration)

Компетенции (индикаторы): ПК-01 (ПК-01.1).

2. Классы, созданием экземпляров которых и установкой в них зависимостей управляет контейнер фреймворка Spring называют \_\_\_\_\_

Правильный ответ: бины / beans

Компетенции (индикаторы): ПК-01 (ПК-01.2).

3. При использовании REST протокола рекомендуется метод \_\_\_\_\_ использовать для получения ресурса, метод \_\_\_\_\_ - чтобы создать новый ресурс, метод \_\_\_\_\_ - для внесения изменений или обновления существующего ресурса, метод \_\_\_\_\_, чтобы избавиться от ресурса.

Правильный ответ: GET, POST, PUT, DELETE

Компетенции (индикаторы): ПК-01 (ПК-01.2).

4. \_\_\_\_\_ это необходимый этап проверки данных, переданных пользователем на серверную сторону приложения при помощи HTTP – запроса.

Правильный ответ: валидация / validation

Компетенции (индикаторы): ПК-01 (ПК-01.3).

### **Задания открытого типа с развернутым ответом**

1. Опишите основные особенности структуры HTTP-запроса

Задачи:

- Проанализировать структуру HTTP-запроса;
- Описать поля строка статуса
- Дать описание строк запроса.

Время выполнения – 20 мин.

Ожидаемый результат (один из возможных вариантов)

HTTP-запрос состоит из нескольких ключевых компонентов, каждый из которых играет важную роль в процессе передачи данных между клиентом и сервером. Вот основные компоненты HTTP-запроса:

### 1. Метод (Method)

Метод указывает, какое действие клиент хочет совершить над ресурсом.

Основные методы включают:

- GET – получение ресурса.
- POST – создание нового ресурса.
- PUT – обновление существующего ресурса.
- DELETE – удаление ресурса.
- HEAD – аналогичен GET, но без тела ответа.
- OPTIONS – запрашивает доступные методы для данного URL.
- PATCH – частичное изменение ресурса.

### ### 2. URL (Uniform Resource Locator)

Это адрес ресурса, к которому отправляется запрос. URL содержит доменное имя сервера, путь до ресурса и параметры запроса (если есть). Например:

`https://example.com/api/users?id=123`

### ### 3. Версия протокола (Protocol Version)

Указывает версию используемого протокола HTTP. Обычно это HTTP/1.1 или HTTP/2.

### ### 4. Заголовки (Headers)

Заголовки содержат дополнительную информацию о запросе и клиентах. Они могут включать такие данные, как:

- Host: Домен, к которому обращается клиент.
- User-Agent: Информация о браузере или приложении, отправившем запрос.

- Content-Type: Тип содержимого в теле запроса (например, `application/json`).
- Authorization: Данные аутентификации (например, токены доступа).
- Accept: Форматы данных, которые клиент готов принять.
- Cookie: Cookies, переданные клиенту ранее.

### ## 5. Тело запроса (Request Body)

Содержит данные, которые отправляются на сервер. В основном используется при методах POST, PUT и PATCH. Может быть пустым для других методов.

### Пример полного HTTP-запроса:

```
POST /api/users HTTP/1.1
Host: example.com
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
{
  "name": "John Doe",
  "email": "johndoe@example.com"
}
```

### Обработка запроса сервером.

1. Метод определяет, какую операцию сервер должен выполнить над ресурсом. Сервер обрабатывает запросы по-разному в зависимости от метода: для получения данных использует GET, а для их изменения – POST, PUT или PATCH.

2. URL помогает серверу определить конкретный ресурс, с которым необходимо работать. Путь и параметры указывают на то, какой именно ресурс требуется обработать.

3. Протокол влияет на то, каким образом осуществляется передача данных. Различные версии HTTP имеют свои особенности обработки запросов и ответов.

4. Заголовки предоставляют дополнительные метаданные, которые помогают серверу правильно интерпретировать запрос. Например, заголовок

Content-Type сообщает серверу формат данных в теле запроса, а заголовок Authorization передает данные для проверки прав доступа.

5. Тело запроса содержит данные, которые должны быть обработаны сервером. Эти данные могут использоваться для создания новых ресурсов (POST), обновления существующих (PUT) или выполнения других операций.

Таким образом, все компоненты HTTP-запроса работают вместе, обеспечивая корректную передачу информации между клиентом и сервером.

Критерии оценивания:

- наличие описания структуры HTTP-запроса
- наличие описания строки статуса;
- описание структуры заголовков и тела запроса.

Компетенции (индикаторы): ПК-01(ПК-01.2).

## 2. Опишите структуру HTTP – ответа

Задачи:

- Проанализировать структуру HTTP-ответа;
- Описать поля строка статуса
- Дать описание строк ответа.

Время выполнения – 20 мин.

Ожидаемый результат (один из возможных вариантов).

HTTP-ответ – это сообщение, которое сервер отправляет клиенту в ответ на HTTP-запрос. Структура HTTP-ответа состоит из нескольких частей, каждая из которых выполняет свою функцию. Рассмотрим основные компоненты HTTP-ответа:

### ### 1. Стока статуса (Status Line)

Первая строка ответа, которая включает три части:

- Версия протокола: Версия HTTP, использованная для ответа (обычно HTTP/1.1 или HTTP/2).

- Код состояния: Числовой код, который указывает результат обработки запроса. Наиболее распространенные коды:

- 200 OK – успешная обработка запроса.
- 404 Not Found – запрошенный ресурс не найден.
- 500 Internal Server Error – внутренняя ошибка сервера.
- Фраза состояния: Описание кода состояния на естественном языке (например, «OK», «Not Found»).

Пример строки статуса:

HTTP/1.1 200 OK

### ### 2. Заголовки (Headers)

Заголовки содержат дополнительную информацию об ответе и сервере.

Некоторые важные заголовки:

- Content-Type: Определяет формат данных в теле ответа (например, text/html, application/json).
- Content-Length: Размер тела ответа в байтах.
- Date: Дата и время отправки ответа.
- Server: Информация о программном обеспечении сервера.

- Set-Cookie: Устанавливает cookie для клиента.
- Cache-Control: Управляет кэшированием ответа.

Пример заголовков:

```
Content-Type: text/html
Date: Mon, 25 Dec 2023 12:34:56 GMT
Server: Apache/2.4.41 (Ubuntu)
Set-Cookie: session_id=abc123; Path=/; HttpOnly
Cache-Control: max-age=3600
```

### ### 3. Пустая строка

После заголовков идет пустая строка, которая отделяет заголовки от тела ответа.

### ### 4. Тело ответа (Response Body)

Содержит данные, которые были запрошены клиентом. Тело ответа может быть пустым (например, для запросов типа HEAD или некоторых ошибок), либо содержать HTML-документ, JSON-данные, изображение и т.п.

Пример тела ответа:

```
{
  "message": "Success!",
  "data": {
    "id": 123,
    "name": "John Doe"
  }
}
```

### ### Обработка HTTP-ответа клиентом

1. Стока статуса предоставляет клиенту основную информацию о результате обработки запроса. Код состояния и фраза состояния позволяют клиенту понять, был ли запрос успешным или произошла ошибка.

2. Заголовки передают дополнительную информацию, необходимую для правильной интерпретации тела ответа. Например, заголовок Content-Type сообщает клиенту, в каком формате представлены данные, а заголовок Set-Cookie устанавливает cookies для последующих запросов.

3. Пустая строка служит разделителем между заголовками и телом ответа, позволяя клиенту легко различать эти две части сообщения.

4. Тело ответа содержит сами данные, которые были запрошены клиентом. Клиент анализирует тело ответа в соответствии с типом данных, указанным в заголовке Content-Type.

Таким образом, структура HTTP-ответа обеспечивает эффективную и стандартизированную передачу данных между сервером и клиентом, позволяя им обмениваться информацией в понятной форме.

Критерии оценивания:

- наличие описания структуры HTTP-ответа
- наличие описания строки статуса;
- описание структуры заголовков и тела ответа.

Компетенции (индикаторы): ПК-01(ПК-01.1).

## **Экспертное заключение**

Представленный комплект оценочных материалов по дисциплине «Современные облачные хранилища данных» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 09.04.02 Информационные системы и технологии.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической комиссии института компьютерных систем и информационных технологий

Ветрова Н. Н.

## Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)
1	В фонд оценочных средств добавлен комплект оценочных материалов	26.02.2025 г., №14	 А.И. Горбунов