**1. MESSAGE BROKERS**

- что лучше amqp или http

- Apache Kafka(transaction log, партиции, гарантии доставки, zookeeper)

- гарантия доставки, как реализуются

- transaction log

- ActiveMQ vs RabbitMQ vs Kafka, основные протоколы, принципы работы, как хранят данные (журнал коммитов, на диске, напр. KahaDB), гарантии доставки

- как положить сообщение в очередь ActiveMQ?

- Data stream processing: Lambda vs Kappa

**2. GIT**

- Git vs other Version Control Systems (Subversion, Mercurial, CVS) differences

- ответы на основные вопросы на собеседовании,

- Что делает git merge.

- merge vs rebase,

- как откатиться до нужного коммита (git reset --hard <идентификатор\_коммита>)

- как удалить коммит (git rebase -i < идентификатор\_коммита > -> drop)

- Как объединить коммиты в один при мерже (git rebase -i < идентификатор\_коммита > -> squash)

- stash, как ансташить (git stash apply "stash@{n}")

- reset(--hard, soft, mixed, keep) vs revert,

- cherypick

**3. METHODOLOGY**

- Agile, Scrum, Canban

- Что такое Agile? Каковы особенности Scrum? Kanban? Waterfall? расскажите про их достоинства и недостатки.

- Расскажите про артефакты Scrum (sprint, retro, stand-up, planning, demo, burn-down diagram)

- Расскажите про артефакты Kanban (backlog, kanban-board, lean)

- agile, чем отличается agile от waterflall, роли в scrum команде (SCRUM Master, Product Owner, Development Team, Stakeholders)

**4. CI/CD**

- принципы, различие

- Continious Integration (Jenkins/hudson, CruiseControl, GitlabCI etc)

- Continious Delivery (Schef, puppet, etc)

**5.** **TEST**

- Виды тестов (unit, integration, functional, smoke, регрессионные, нагрузочные) и используемые для них инструменты

- TDD, BDD

- Отличия Test double: Stub/mock/spy/fake

**6. DOCKER**

- Docker vs. виртуализация

- Что такое гипервизор?

- Какие сети доступны по умолчанию в Docker? устройство сети

- Add vs. Copy (Add старше, может больше: загружать файлы из сети и разархивировать, не реком. к примен.)

- CMD vs. ENTRYPOINT в Dockerfile (ENTRYPOINT используется для определения основной цели контейнера, в то время как CMD обеспечивает поведение по умолчанию, которое может быть переопределено)

- EXPOSE

- Что за команда docker ps, какие параметры у нее есть?

- Dockerfile

- docker-compose

- volume vs bind mounts (v существуют только внутри Docker-a в директории /var/lib/docker/volumes, can be referenced only by their name, используются для backup-ов persistent данных, а bm – любая директория на хосте, используются при разработке, чтобы не нужно было каждый раз пересобирать image, когда меняется код приложения)

- writable layer (этим контейнер и отличается от образа)

- blue/green deploy

- Healthcheck

- swarm vs kubernetes (это как сравнивать paint и photoshop, kubernetes дает автомасшабирование от нагрузки, политики безопасности, мониторинг…)

- основные команды и вопросы на собесах

- что делать, если контейнер падает

- снятие дампа

**7. SQL**

- БД, СУБД, виды NoSQL, SQL(DDL, DML, DCL, TCL)

- Sql vs NoSql, В чем их основное отличие? (ACID vs BASE) для чего лучше подходит SQL, noSQL БД? Был ли опыт работы с NoSQL (какой из них) , виды NoSql?

- Нормализация бд, какие нормальные формы знает

- Что такое первичный, внешний ключ?

- Типы constraints (7 штук)

- Что такое план запроса? (Запрос: парсинг, отпимизация, выполнение) EXPLAIN, хинты, оптимальный(не обязательно это лучший план) найденный план кэшируется

- Что делать если запрос медленно работает и индексы не помогают, и данные часто меняются т.е. кэш не поможет?

- Что такое индекс? Для чего их используют? Кластерный, некластерный

- Транзакция, Какими свойствами должна обладать транзакция? (ACID), проблемы транзакций, уровни изоляции транзакций

- drop vs truncate vs delete

- Какие виды подзапросов вы знаете? (результат подзапроса м.быть скалярный, строковый табличный; коррелированный и некорр.)

- Join vs subquery

- Для чего используется group by? Для чего используется having?

- Для чего нужны операторы UNION, UNION ALL, INTERSECT (общие для таблиц строки), EXCEPT (все строки из первой таблицы, которых нет во второй) можно заменить через EXISTS и NOT EXISTS?

- COALESCE

- Если выполнить «select \* from a, b», что мы получим? (декартово произведение)

- Какие виды join вы знаете?

- full join vs. cross join

- Если вы собираетесь соединить несколько таблиц в запросе (например, n таблиц), сколько условий соединения вам нужно использовать? (n-1)

- «select \* from a, b where a.id=b.a\_id» или «select \* from a inner join b on a.id=b.a\_id» что будет выполнено быстрее? (одинаково если оптимизатор запросов построит одинаковый план)

- Представление, ограничения

- Хранимые процедуры vs пользовательские функции vs триггеры (как в postgres?)

- https://habr.com/ru/articles/255361/

- Есть 2 таблицы, в первой 2 записи, во второй 3, 2 совпадения по FK. Сколько кортежей вернут разные джоины: inner, left, right, full, cross? (2, 2 или 3, 3, 3 или 4, 6)

- есть 4 таблицы: product(id, name, price); order(id, customer\_id); order\_product(id, order\_id, product\_id, product\_count); customer(id, name); 1) вывести количество заказов для каждого покупателя 2) вывести количество заказанных позиций для каждого покупателя 3) вывести товары, которые были заказаны больше 3 раз

- Партиционирование, Репликация, Шардирование

- MongoDB, особенности, когда использовать, транзакции, read/write concern, блокировки replica set, WireTiger

**8. ORM/JPA/HIBERNATE**

- В чем отличия JPA от Hibernate? Какие еще реализации JPA вы можете назвать?

- Какие преимущества от использования Hibernate?

- Какие требования JPA | Hibernate к Entity классам вы можете перечислить?

- Какие типы данных допустимы в атрибутах Entity класса (полях или свойствах)?

- Какие типы связей (relationship) между Entity вы знаете (перечислите восемь типов, либо укажите четыре типа связей, каждую из которых можно разделить ещё на два вида)?

- Какие типы данных можно использовать в атрибутах, входящих в первичный ключ Entity класса (составной или простой), чтобы полученный первичный ключ мог использоваться для любой базы данных? А в случае автогенерируемого первичного ключа (generated primary keys)?

- Какие два типа fetch стратегии в JPA вы знаете?

- Какие четыре статуса жизненного цикла Entity объекта (Entity Instance’s Life Cycle) вы можите перечислить?

- Какой аннотацей можно исключить поля и свойства Entity из маппинга (property or field is not persistent)?

- оптимистическая и пессимистическая блокировки. В чем между ними разница?

- Какие шесть видов блокировок (lock) описаны в спецификации JPA (или какие есть значения у enum LockModeType в JPA)?

- Почему объекты в Hibernate не могут быть final

- В каких статусах могут находится объекты в Hibernate,

- Entitygraph

- FlushModes

- проблема n+1

- fetch size vs. batch size

- hibernate session vs entitymanager

- @DynamicUpdate

- конвертеры, пользовательские типы данных

- Entity Listeners, @PrePersist…

- Уровни кэширования Hibernate

- Как можно настроить работу Hibernate с кэшированием запросов, два сервиса, redis про внутреннюю очередь

- @Cacheable, распределенный и локальный кеш, инвалидация, алгоритмы вытеснения, распр. Лок

- как не меняя респонса добавить новое поле (resultset, кастомный тип данных)

**9. HTTP/SERVLETS/JSP**

- В чем главная особенность протокола HTTP?

- Какой метод сервлета всегда вызывается при обработке запроса?

- Как сервлет контейнер узнает, какой сервлет должен обрабатывать запрос?

- В чем отличия redirect и forward? В каких случаях нужно применять redirect? Почему?

- Предположим мы объявили поле в классе сервлета, в чем будет особенность доступа к этому полю?

- Может ли сервлет/jsp вернуть бинарные данные (что-либо отличное от текста)? Как это реализовать?

- Может ли сервлет обработать PUT/DELETE/... запрос?

- Отличия сервлета и jsp страницы.

- Назовите стандартные переменные, которые можно использовать в сервлете/jsp?

- Что такое http сессия? Для чего она нужна? Каким образом сессия «привязана» к пользователю? Как очистить сессию? Как еще можно сохранить-передавать информацию в пределах нескольких запросов?

**10. SPRING**

- Что такое Spring? Из каких частей состоит Spring Framework? Преимущества использования

- Spring vs EJB

- Объясните суть паттерна DI или IoC.

- Какие еще IoC контейнеры вы знаете?

- как работает injection в Spring, максиамально подробно

- Какие существуют виды DI? Приведите примеры.

- Как добавить поддержку Spring в приложение/web-приложение?

- виды конфигураций

- В чем роль ApplicationContext в Spring?

- Как подымается приложение (beandefinition, ApplicationContext и все остальное)

- реализации beanFactoryPostProcessor и beanPostProcessor

- что такое spring bean? Как объявить бин? Какие способы настроить spring bean?

- Жизненный цикл бина (@PostConstruct, @Predestroy, init method,и все остальное)

- как лучше внедрять бины и почему

- bean scopes

- чем отличается жизненный цикл prototype и singleton

- prototype - в каких случаях используется, как у него вызвать destroy

- приходилось ли работать со скоупами отличными от singleton (рассказать про @RequestScope)

- singleton не потокобезопасен

- Spring обеспечивает потокобезопасность?

- как внедрить prototype в singleton

- как создать бины с одинаковыми типами

- где хранится RequestScope (какая-то мапа)

- Autowired vs Inject vs Resource

- как Spring понимает generic-типы при создании бинов (c Spring 4.0 появился ResolvableType, через @Qualifier)

- self injection

- a зависит от в, в зависит от а, как решить циклическую зависимость?

- AOP, Proxy, CGLib vs Dynamic Proxy

- транзакции в Spring, про аннотацию @Transactional для чего она нужна, когда не будет работать

- какие паттерны проектирования, используются в Spring Framework

- open-session-in-view

- как написать свой starter и зачем, отличия от библиотеки @Conditional

- spring data rest - недостатки, когда не стоит использовать

- Lombok @RequiredArgsConstructor vs @AllArgsConstructor

- Какие есть проблемы у Pageable?

- Shutdown hook

- @ControllerAdvice (Exception Handler) + Custom Annotation for validation

- про JDBC

- @Async

- про webflux

**11. SECURITY**

- Назовите основные Security Risk, которые возникают при разработке web? OWASP Top 10 Vulnerabilities:

1. Broken Access Control, 2. Broken Authentication, 3. Injections , 4. Cross Site Scripting, 5. Cross Site Request Forgery, 6. Using Components with Known Vulnerabilities, 7. Insufficient Logging and Monitoring failures, 8. Insecure Deserialization, 9. Security Misconfiguration, 10. Cryptographic Failures…

Что каждый из них обозначает. Как с ними бороться?

- шифрование симметричное и ассиметричное base64, алгоритмы шифрования sha512…

- JWT

- Единая точка входа SSO

- ADFS, saml

- OAuth2.0 (components, flow, types of grants)

- OpenId

**12. PATTERNS**

- DRY, KISS, YAGNI, SOLID

- Паттерны микросервисов, основные группы:

1. Декомпозиция: **Decompose By Business Capability, Decompose By Subdomain**

## 2. Обнаружение сервисов: [Service registry](https://microservices.io/patterns/service-registry.html), [Client-side discovery](https://microservices.io/patterns/client-side-discovery.html), [Server-side discovery](https://microservices.io/patterns/server-side-discovery.html)

## 3. Управление данными: **Database Per Service, Shared Database, API Composition, Command Query Responsibility Segregation, Event Sourcing, Saga**

## 4. Взаимодействие с внешними клиентами: **API Gateway, Backends for Frontends**

5. Надежность: **Circuit Breaker, Bulkhead**

## 6. Деплой: [Service instance per host](https://microservices.io/patterns/deployment/single-service-per-host.html)(container), [Service deployment platform](https://microservices.io/patterns/deployment/service-deployment-platform.html), **Blue-Green Deployment**

## 7. Мониторинг: **Log Aggregation, Distributed Tracing, Health Check**

8. Общие: [Service Template](https://microservices.io/patterns/service-template.html), [Externalized configuration](https://microservices.io/patterns/externalized-configuration.html)

**13. JAVA**

Общие вопросы:

- принципы ООП

- что такое Java, JDK, JRE, JVM

- Нововведения в Java 8, 11

- позднее связывание

- Сколько аргументов может быть у метода? Как объявить метод с переменным количеством аргументов?

- Какие основные группы типов данных в Java вы знаете? - > В чем их отличия?

- Generics (ко\контр\ин-вариантность, PECS, type erasure, wildcard, пример generic типа)

- модификаторы доступа

- static, final

- Sorting (алгоритмы сортировки), Binary tree, Какой алгоритм сортировки используется для массивов примитивов? Какой для массивов объектов? Для листа объектов?

Exceptions:

- иерархия, checked, unchecked

- exceptions в сигнатуре метода правила переопределения

- Пусть метод может выбросить несколько разных исключений. Имеет ли значение, в каком порядке их обрабатывать, т.е. порядок блоков catch?

- Можно ли использовать конструкцию только try-finally? Для чего она может понадобиться?

- Всегда ли будет выполнен блок finally? А если в конце блока try у нас return?

- Всегда ли мы можем «поймать»/обработать исключение?

Классы и объекты:

- Можно ли объявить final static метод в классе?

- Какие модификаторы доступа может иметь класс?

- Сколько классов можно объявить в одном файле? Какие условия при этом должны быть выполнены?

- Если необходимо использовать 2+ класса с одинаковыми именами (например java.sql.Date/java.util.Date) в одном файле, как это реализовать?

- Что такое абстрактный класс? Может ли абстрактный класс не иметь абстрактных методов?

- В базовом классе есть public static метод. Можно ли в классе наследнике объявить статический метод с такой же сигнатурой? Почему? Какой из методов будет вызван, если мы работаем с объектом наследником через ссылку на родительский класс?

- Можем ли мы использовать this/нестатическое поле класса внутри статического метода?

- Можно ли объявить класс внутри метода?

- Что такое анонимные классы? Для чего их удобно использовать?

- Как обратиться к методу внутреннего класса через ссылку на внешний? А наоборот?

- Можно ли переопределить статический метод? А перегрузить?

- Если нужно ускорить работу со строками, какие классы вы будете использовать?

- Классы обертки. Autoboxing/Unboxing.

- Как создать immutable объект?

- Класс Object. Какие его методы Вы знаете/использовали?

- Класс String. В чем его особенности? Почему нельзя унаследоваться от класса String? Почему его сделали immutable и что это дает?

- Интерфейсы Comparable/Comparator, правила переопределения?

- Класс Optional, разница между orElse и orElseGet в Optional

- Отличия equals() и ==.

Наследование:

- Можно ли изменить модификаторы доступа при наследовании? Каким образом? А почему именно так? Что еще можно изменить в сигнатуре метода при наследовании?

- Как обратиться к объекту родителя, например, вызвать его метод? А если нужно вызвать метод родителя родителя?

- Перегрузка и переопределение. В чем разница?

- Как определить, что объект определенного типа/наследует интерфейс-класс?

Интерфейсы, Enums:

- Что такое интерфейс в java? - Этоконтракт, описывающий некий функционал.

Классы, которые подписываются на этот контракт, должны реализовать этот функционал.

- Что допускается объявить в интерфейсе?

- Какие модификаторы доступа по умолчанию имеют методы интерфейса? А поля? Почему?

- Допускается ли наследование нескольких интерфейсов с методами одинаковой сигнатуры? (+ Что входит в сигнатуру метода?)

- Могут ли интерфейсы наследовать другие интерфейсы? Только один или сколько угодно?

- Сколько классов/интерфейсов допускается наследовать/реализовывать в java?

- Предположим есть 2 интерфейса с одинаковыми методами (имя, параметры), но возвращающими разное значение, можно ли реализовать оба этих интерфейса? Почему? - Нет

- Что такое java Enum? Методы. Пример с abstract-методом; полем, конструктором и геттером

Сериализация:

- Что такое сериализация?

- Каким образом можно сделать объект сериализуемым? Что будет, если сериализовать объект, который не унаследовал Serializable (Externalizable) интерфейс?

- Как исключить поле объекта из сериализации?

- Можно ли переопределить механизм сериализации на свой?

- Пусть родительский класс наследует Serializable. Можем ли мы сериализовать класс наследник? Почему?

- Предположим мы передаем сериализованнй класс на другую ВМ. Какие условия должны быть выполнены для успешной десериализации?

- Любой ли java класс может быть сериализован-десериализован?

JDBC

- Statement и PreparedStatement. Есть одинаковый запрос, с использованием чего он будет выполняться быстрее? Почему? Какие еще преимущества у PreparedStatement’а?

- Пусть из БД в ResultSet вычитано int значение. Как определить, что это значение содержит NULL?

- Каким образом в JDBC реализована поддержка транзакций?

Рефлексия, принцип работы, возможности, популярные методы

- Объекты типа Class. Что за они? Для чего нужны? Что с их помощью можно сделать?

Паттерны проектирования (20 штук, 3 группы)

Classloaders, принцип работы, динамическая(ленивая) загрузка классов, свой загрузчик

Stream API, Functional Interfaces, Lambda:

- что это такое,

- отличие lambda от анонимного класса (state, компиляция, this, expression, arg types, performance, usage)

- основные ф.и: Predicate, Consumer, Supplier, Function, UnaryOperator

- flatMap, peek, reduce, Collectors.toMap, Collectors.groupingBy, sum, average,

- если в лямбде checked-exception бросается мы не имеем возможности пробросить его из метода в котором лямбда описана, почему так происходит

- какие ограничения накладываются на переменные, которые мы используем внутри лямбды (final, effective final)

Устройство памяти & Сборка мусора

- области памяти

- Назвать разные реализации gc, механизм работы, tuning

- как узнать, какой сборщик мусора исп-ся и поменять его на другой

- heap, stack, -Xms –Xmx, happens before

- -Xms128m -Xmx256m -XX:MetaspaceSize=100M -XX:+UseParallelGC

Коллекции:

- Что такое коллекция? (контейнер для объектов, позволяющий производить различные операции над группой объектов)

- Из каких основных частей состоит Collections Framework? (базовые интерфейсы, набор классы-реализаций, алгоритмы)

- Какие основные коллекции есть в Java?

- Методы интерфейсов Collection, Collections

- Каковы основные отличия List от Set?

- Каким образом определяется «одинаковость» объектов при помещении в Set?

- Как правильно переопределить equals & hashCode? Контракт между equals & hashCode

- Как можно синхронизировать доступ например к ArrayList?

- Можно ли вставить null в List/Set/Map (ключ)? Что произойдет при вставке null значения в List/Set/ Map (ключ)?

- Для чего нужен Iterator? Какие его методы вы использовали?

- Как корректно удалить некоторые значения из коллекции в цикле?

- Iterator и ListIterator чем отличаются?

- Какие дополнительные операции позволяет выполнить ListIterator? (add, set, hasPrevious, previous)

- Spliterator vs iterator

- ArrayList и LinkedList? Отличия? Какую реализацию предпочтете использовать при частых модификациях коллекции? Почему? Удалить первый элемент в ArrayList и LinkedList, что быстрее и почему?

- Big O, сложность операций в Коллекциях, внутр. устройство HashMap, HashSet, ArrayList, LinkedList, Queue

- Отличия HashSet и TreeSet.

- Устройство HashMap

- Каким требованиям должен удовлетворять ключ Map?

- Можно ли использовать array (int[]) в качестве ключа? Почему? А если понадобится это сделать, как решить проблему?

- HashMap как потерять ключ

- Что такое коллизия? Что произойдет при вставке в Set/Map в случае коллизии?

Многопоточность:

- процесс vs поток

- deadlock, racecondition

- Как создать поток в java? Почему 2мя способами?

- основные методы работы с потоками (join, yield…)

- квант времени

- Как «защитить» объект, с которым работают несколько потоков?

- Какие способы синхронизации потоков в java вам известны?

- volatile

- Пусть у нас объявлена volatile **int** переменная с начальным значением = 0. Если одновременно запустить 3 потока, каждый из которых делает в цикле инкремент этой переменной 3 раза. Какое значение будет иметь эта переменная, после выполнения всех потоков? А если убрать volatile?

- Пусть в объекте 2 синхронизированных метода (не статич) и один из них вызван в отдельном потоке. Что будет при вызове второго метода? А если он будет несинхронизирован?

- Что является объектом блокировки, при вызове синхронизированного статического метода?

- Как корректно завершить выполнение потока?

- invoke run instead of start in thread

- пакет java.util.concurrent (из чего состоит, почитать про Atomic, CompletableFuture, ExecutorService, ForkJoinPool, conc. collections), CompletableFuture get() vs. join()

Other:

- реализация кэша

- Profilers

- как определить сколько памяти нужно приложению

- OutOfMemoryError - что делать, как разбирать причину

**14. OTHER**

- разница между SOAP и REST

- Назовите библиотеки, которые Вы использовали для работы с REST/SOAP?

- Когда по вашему мнению лучше использовать REST, а когда SOAP?

- Предположим вы создали сервис. Как вы предоставите-опишете “потребителям” этого сервиса ваши методы? В случае REST/SOAP

- Software Development Life Cycle

- EJB основные идеи, виды бинов

- Виды XML-парсеров. Как задать схему для XML? Что такое XSLT?

- Что такое iBatis/MyBatis?

- CAP теорема

- Стек технологий Spring Cloud(ConfigServer, Eureka, Ribbon, Feign, Hystrix, ZuulProxy, sleuth)

- Monolit vs microservices

- что такое Apache Camel

- Rundeck

- Wiremock, serenity-bdd

- AWS, основные сервисы

- Prometheus, Grafana

- Elacticsearch, Logstash, Kibana

- зачем необходима оркестрация? Основы Mezos, Marafone, Kubernetes

- Инернет в Британии (как устроен, монополисты, провайдеры)

- Просмотреть какие-нибудь исходники сервисов ShellEnergy

- основные протоколы: HTTP (S, 2), модель OSI, TCP/IP

- Системы поддержки код-ревью (fisheye, stash etc)

- Системы онлайн-документирования (MediaWiki, TWiki, Redmine-Wiki, Confluence)

- Основные Linux команды (sudo, ifconfig, ssh, scp, top, ps, kill, grep, cat, less, head, tail, bash, iptables, tcpdump, ifconfig, gedit)

- как добавить прав на чтение группе пользователей linux (shmode)

- как узнать чем занят порт (netstat)

- ansible

- saas vs iaas vs paas

## 15. Вопросы по опыту кандидата

- Расскажите о Вашем последнем проекте: краткое описание проекта, стек технологий, какие задачи на проекте выполняли, структура команды, коммуникация внутри команды и с заказчиком, как было организовано тестирование, … + уточняющие вопросы при необходимости.