

CORE - 1

1. ООП

1. Что такое ООП?
2. Какие преимущества у ООП?
3. Какие недостатки у ООП?
4. Назовите основные принципы ООП.
5. Что такое инкапсуляция? (С примером)
6. Что такое наследование?(С примером)
7. Что такое полиморфизм? (С примером)
8. Что такое ассоциация, агрегация и композиция?
9. Расскажите про раннее и позднее связывание.
10. SOLID

2. Java

1. Какая основная идея языка?
2. За счет чего обеспечивается кроссплатформенность?
3. Какие преимущества у java?
4. Какие недостатки у Java?
5. Что такое JDK? Что в него входит?
6. Что такое JRE? что в него входит?
7. Что такое JVM?
8. Что такое byte code?
9. Что такое загрузчик классов (classloader)?
10. Что такое JIT?
11. Что такое сборщик мусора? (Garbage collector)
12. Что такое Heap и Stack память в Java? Чем они отличаются?

3. Процедурная Java

1. Какие примитивные типы данных есть в Java?
2. Что такое char?
3. Сколько памяти занимает boolean?
4. Что такое классы-обертки?
5. Что такое автоупаковка и автораспаковка?
6. Что такое явное и неявное приведение типов? В каких случаях в java нужно использовать явное приведение?
7. Что такое пул интов?
8. Какие нюансы у строк в Java?

9. Что такое пул строк?
10. Почему не рекомендуется изменять строки в цикле? Что рекомендуется использовать?
11. - Почему строки не рекомендуется использовать для хранения паролей? -
12. Почему String неизменяемый и финализированный класс?
13. - Почему строка является популярным ключом в HashMap в Java? -
14. Что делает метод intern() в классе String?
15. Можно ли использовать строки в конструкции switch?
16. Какая основная разница между String, StringBuffer, StringBuilder?
17. Существуют ли в java многомерные массивы?
18. Какими значениями иницируются переменные по умолчанию?
19. Что такое сигнатура метода?
20. Расскажите про метод main
21. Каким образом переменные передаются в методы, по значению или по ссылке?

4. ООП в Java

1. Какие виды классов есть в java?
2. Расскажите про вложенные классы. В каких случаях они применяются?
3. - Что такое «локальный класс»? Каковы его особенности?
4. Что такое «анонимные классы»? Где они применяются?
5. - Каким образом из вложенного класса получить доступ к полю внешнего класса?
6. Что такое перечисления (enum)?
7. Как проблема ромбовидного наследования решена в java?
8. Что такое конструктор по умолчанию?
9. Могут ли быть приватные конструкторы? Для чего они нужны?
10. Расскажите про классы-загрузчики и про динамическую загрузку классов.
11. Чем отличаются конструкторы по-умолчанию, копирования и конструктор с параметрами?
12. Какие модификаторы доступа есть в Java? Какие применимы к классам?
13. Что означает модификатор static?
14. Может ли статический метод быть переопределён или перегружен?
15. Могут ли нестатические методы перегрузить статические?
16. Можно ли сузить уровень доступа/тип возвращаемого значения при переопределении метода?
17. Что можно изменить в сигнатуре метода при переопределении? Можно ли менять модификаторы (throws и тп)?
18. Могут ли классы быть статическими?
19. Что означает модификатор final? К чему он может быть применим?
20. Что такое абстрактные классы? Чем они отличаются от обычных?
21. Может ли быть абстрактный класс без абстрактных методов?
22. Могут ли быть конструкторы у абстрактных классов? Для чего они нужны?
23. Что такое интерфейсы? Какие модификаторы по умолчанию имеют поля и методы интерфейсов?

24. Чем интерфейсы отличаются от абстрактных классов? В каких случаях следует использовать абстрактный класс, а в каких интерфейс?
25. Может ли один интерфейс наследоваться от другого? От двух других?
26. Что такое дефолтные методы интерфейсов? Для чего они нужны?
27. Как решается проблема ромбовидного наследования при наследовании интерфейсов при наличии default методов?
28. Каков порядок вызова конструкторов и блоков инициализации с учётом иерархии классов?
29. **Пример с кодом**
30. Зачем нужны и какие бывают блоки инициализации?
31. Для чего в Java используются статические блоки инициализации?
32. - Что произойдёт, если в блоке инициализации возникнет исключительная ситуация?
33. - Какое исключение выбрасывается при возникновении ошибки в блоке инициализации класса?
34. Что такое класс Object?
35. Какие методы есть у класса Object (перечислить все)? Что они делают?
36. Расскажите про equals и hashCode
37. Каким образом реализованы методы hashCode() и equals() в классе Object?
38. Зачем нужен equals(). Чем он отличается от операции ==?
39. Правила переопределения equals()
40. Что будет, если переопределить equals() не переопределяя hashCode()? Какие могут возникнуть проблемы?
41. Какой контракт между hashCode() и equals()?
42. Для чего нужен метод hashCode()?
43. - Правила переопределения метода hashCode().
44. - Есть ли какие-либо рекомендации о том, какие поля следует использовать при подсчете hashCode()?
45. Могут ли у разных объектов быть одинаковые hashCode()?
46. Почему нельзя реализовать hashCode() который будет гарантированно уникальным для каждого объекта?
47. Есть класс Point{int x, y;}. Почему хэш-код в виде $31 * x + y$ предпочтительнее чем $x + y$?
48. Чем `a.getClass().equals(A.class)` отличается от `a instanceof A.class`

5. Исключения

1. Что такое исключения?
2. Опишите иерархию исключений.
3. Расскажите про обрабатываемые и необрабатываемые исключения
4. Можно ли обработать необрабатываемые исключения?
5. Какой оператор позволяет принудительно выбросить исключение?
6. О чем говорит ключевое слово throws?
7. Как создать собственное («пользовательское») исключение?
8. Расскажите про механизм обработки исключений в java (Try-catch-finally)

9. Возможно ли использование блока try-finally (без catch)?
10. Может ли один блок catch отлавливать сразу несколько исключений?
11. Всегда ли выполняется блок finally? Существуют ли ситуации, когда блок finally не будет выполнен?
12. Может ли метод main() выбросить исключение во вне и если да, то где будет происходить обработка данного исключения?
13. В каком порядке следует обрабатывать исключения в catch блоках?
14. Что такое механизм try-with-resources?
15. Что произойдет если исключение будет выброшено из блока catch после чего другое исключение будет выброшено из блока finally?
16. Что произойдет если исключение будет выброшено из блока catch после чего другое исключение будет выброшено из метода close() при использовании try-with-resources?

6. Сериализация и копирование

1. Что такое сериализация и как она реализована в Java?
2. Для чего нужна сериализация?
3. Опишите процесс сериализации/десериализации с использованием Serializable.
4. Как изменить стандартное поведение сериализации/десериализации?
5. Какие поля не будут сериализованы при сериализации? Будет ли сериализовано final поле?
6. Как создать собственный протокол сериализации?
7. Какая роль поля serialVersionUID в сериализации?
8. Когда стоит изменять значение поля serialVersionUID?
9. В чем проблема сериализации Singleton?
10. Расскажите про клонирование объектов.
11. В чем отличие между поверхностным и глубоким клонированием?
12. Какой способ клонирования предпочтительней?
13. Почему метод clone() объявлен в классе Object, а не в интерфейсе Cloneable?
14. Как создать глубокую копию объекта? (2 способа)

CORE-2

7. Дженерики

1. Что такое дженерики?
2. Для чего нужны дженерики?
3. Что такое сырые типы (raw type)?
4. Что такое вайлдкарды?
5. - Расскажите про принцип PECS

8. Коллекции

1. Что такое «коллекция»?
2. Расскажите про иерархию коллекций
3. Почему Map — это не Collection, в то время как List и Set являются Collection?
4. В чем разница между классами `java.util.Collection` и `java.util.Collections`?
5. Какая разница между итераторами с fail-fast и fail-safe поведением? (С примерами)
6. Чем различаются Enumeration и Iterator?
7. Как между собой связаны Iterable, Iterator и «for-each»?
8. Можно ли итерируясь по ArrayList удалить элемент? Какое вылетит исключение?
9. Как поведёт себя коллекция, если вызвать `iterator.remove()`?
10. Чем Set отличается от List?
11. Расскажите про интерфейс Set.
12. Расскажите про реализации интерфейса Set
13. В чем отличия TreeSet и HashSet?
14. Чем LinkedHashSet отличается от HashSet?
15. Что будет, если добавлять элементы в TreeSet по возрастанию?
16. Как устроен HashSet, сложность основных операций.
17. Как устроен LinkedHashSet, сложность основных операций.
18. Как устроен TreeSet, сложность основных операций.
19. Расскажите про интерфейс List
20. Как устроен ArrayList, сложность основных операций.
21. Как устроен LinkedList, сложность основных операций.
22. Почему LinkedList реализует и List, и Deque?
23. Чем отличаются ArrayList и LinkedList?
24. Что такое Queue?
25. Что такое Dequeue? Чем отличается от Queue?
26. Приведите пример реализации Dequeue.
27. Какая коллекция реализует FIFO?
28. Какая коллекция реализует LIFO?
29. Оцените количество памяти на хранение одного примитива типа `byte` в LinkedList?
30. Оцените количество памяти на хранение одного примитива типа `byte` в ArrayList?
31. Какие существуют реализации Map?
32. Как устроена HashMap, сложность основных операций? (Расскажите про принцип корзины)
33. Что такое LinkedHashMap?
34. Как устроена TreeMap, сложность основных операций?
35. Что такое WeakHashMap?
36. Как работает HashMap при попытке сохранить в него два элемента по ключам с одинаковым `hashCode()`, но для которых `equals() == false`?

37. Что будет, если мы кладем в HashMap ключ, у которого equals и hashCode определены некорректно?
38. Возможна ли ситуация, когда HashMap вырождается в список даже с ключами имеющими разные hashCode()?
39. Почему нельзя использовать byte[] в качестве ключа в HashMap?
40. Будет ли работать HashMap, если все добавляемые ключи будут иметь одинаковый hashCode()?
41. Какое худшее время работы метода get(key) для ключа, которого нет в HashMap?
42. Какое худшее время работы метода get(key) для ключа, который есть в HashMap?

9. Функциональные интерфейсы

1. Что такое функциональный интерфейс?
2. Для чего нужна аннотация @FunctionalInterface?
3. Какие встроенные функциональные интерфейсы вы знаете?
4. Что такое ссылка на метод?
5. Что такое лямбда-выражение? Чем его можно заменить?

10. Stream API

1. Что такое Stream API? Для чего нужны стримы?
2. Почему Stream называют ленивым?
3. Какие существуют способы создания стрима?
4. Как из коллекции создать стрим?
5. Какие промежуточные методы в стримах вы знаете?
6. Расскажите про метод peak().
7. Расскажите про метод map().
8. Расскажите про метод flatMap().
9. Чем отличаются методы map() и flatMap().
10. Расскажите про метод filter().
11. Расскажите про метод limit().
12. Расскажите про метод skip().
13. Расскажите про метод sorted().
14. Расскажите про метод distinct().
15. Какие терминальные методы в стримах вы знаете?
16. Расскажите про метод collect().
17. Расскажите про метод reduce().
18. Расскажите про класс Collectors и его методы.
19. Расскажите о параллельной обработке в Java 8.
20. Что такое IntStream и DoubleStream?

11. Java 8

1. Какие нововведения появились в java 8?
2. Какие новые классы для работы с датами появились в java 8?
3. Расскажите про класс Optional
4. Что такое Nashorn?
5. Что такое jjs?
6. Какой класс появился в Java 8 для кодирования/декодирования данных?
7. Как создать Base64 кодировщик и декодировщик?
8. Какие дополнительные методы для работы с ассоциативными массивами (maps) появились в Java 8?
9. Что такое LocalDateTime?
10. Что такое ZonedDateTime?

Многопоточность

1. Чем процесс отличается от потока?
2. Чем Thread отличается от Runnable? Когда нужно использовать Thread, а когда Runnable?(Ответ что тред - это класс, а ранбл интерфейс - считается не полным, нужно рассказать подробно)
3. Что такое монитор? Как монитор реализован в java?
4. Что такое синхронизация? Какие способы синхронизации существуют в java?
5. Как работают методы wait(), notify() и notifyAll()?
6. В каких состояниях может находиться поток?
7. Что такое семафор? Как он реализован в Java?
8. Что обозначает ключевое слово volatile? Почему операции над volatile переменными не атомарны?
9. Для чего нужны Atomic типы данных? Чем отличаются от volatile?
10. Что такое потоки демоны? Для чего они нужны? Как создать поток-демон?
11. Что такое приоритет потока? На что он влияет? Какой приоритет у потоков по умолчанию?
12. Как работает Thread.join()? Для чего он нужен?
13. Чем отличаются методы yield() и sleep()?
14. Как правильно остановить поток? Для чего нужны методы .stop(), .interrupt(), .interrupted(), .isInterrupted().
15. Чем Runnable отличается от Callable?
16. Что такое FutureTask?
17. Что такое deadlock?
18. Что такое livelock?
19. Что такое race condition?
20. Что такое Фреймворк for/join? Для чего он нужен?
21. Что означает ключевое слово synchronized? Где и для чего может использоваться?

22. Что является монитором у статического синхронизированного класса?
23. Что является монитором у нестатического синхронизированного класса?

SQL и базы данных

1. Что такое DDL? Какие операции в него входят? Рассказать про них.
2. Что такое DML? Какие операции в него входят? Рассказать про них.
3. Что такое TCL? Какие операции в него входят? Рассказать про них.
4. Что такое DCL? Какие операции в него входят? Рассказать про них.
5. Нюансы работы с NULL в SQL. Как проверить поле на NULL?
6. Виды Join'ов?
7. Что лучше использовать join или подзапросы? Почему?
8. Что делает UNION?
9. Чем WHERE отличается от HAVING (ответа про то что используются в разных частях запроса - недостаточно)
10. Что такое ORDER BY?
11. Что такое GROUP BY?
12. Что такое DISTINCT?
13. Что такое LIMIT?
14. Что такое EXISTS?
15. Расскажите про операторы IN, BETWEEN, LIKE.
16. Что делает оператор MERGE? Какие у него есть ограничения?
17. Какие агрегатные функции вы знаете?
18. Что такое ограничения (constraints)? Какие вы знаете?
19. (?) Что такое суррогатные ключи?
20. Что такое индексы? Какие они бывают?
21. Чем TRUNCATE отличается от DELETE?
22. Что такое хранимые процедуры? Для чего они нужны?
23. Что такое представления (VIEW)? Для чего они нужны?
24. Что такое временные таблицы? Для чего они нужны?
25. Что такое транзакции? Расскажите про принципы ACID.
26. Расскажите про уровни изолированности транзакций.
27. Что такое нормализация и денормализация? Расскажите про 3 нормальные формы?
28. Что такое TIMESTAMP?

JPA и Hibernate

1. Что такое ORM? Что такое JPA? Что такое Hibernate?
2. Что такое EntityManager? Какие функции он выполняет?
3. Каким условиям должен удовлетворять класс чтобы являться Entity?
4. Может ли абстрактный класс быть Entity?
5. Может ли Entity класс наследоваться от не Entity классов (non-entity classes)?

6. Может ли Entity класс наследоваться от других Entity классов?
7. Может ли не Entity класс наследоваться от Entity класса?
8. Что такое встраиваемый (Embeddable) класс? Какие требования JPA устанавливает к встраиваемым (Embeddable) классам?
9. Что такое Mapped Superclass?
10. Какие три типа стратегии наследования мапинга (Inheritance Mapping Strategies) описаны в JPA?
11. Как маются Enumy?
12. Как маются даты (до java 8 и после)?
13. Как смापить коллекцию примитивов?
14. Какие есть виды связей?
15. Что такое владелец связи?
16. Что такое каскады?
17. Какие два типа fetch стратегии в JPA вы знаете?
18. Какие четыре статуса жизненного цикла Entity объекта (Entity Instance's Life Cycle) вы можете перечислить?
19. Как влияет операция persist на Entity объекты каждого из четырех статусов?
20. Как влияет операция remove на Entity объекты каждого из четырех статусов?
21. Как влияет операция merge на Entity объекты каждого из четырех статусов?
22. Как влияет операция refresh на Entity объекты каждого из четырех статусов?
23. Как влияет операция detach на Entity объекты каждого из четырех статусов?
24. Для чего нужна аннотация Basic?
25. Для чего нужна аннотация Column?
26. Для чего нужна аннотация Access?
27. Для чего нужна аннотация Cacheable?
28. Для чего нужны аннотации @Embedded и @Embeddable?
29. Как смापить составной ключ?
30. Для чего нужна аннотация ID? Какие @GeneratedValue вы знаете?
31. Расскажите про аннотации @JoinColumn и @JoinTable? Где и для чего они используются?
32. Для чего нужны аннотации @OrderBy и @OrderColumn, чем они отличаются?
33. Для чего нужна аннотация Transient?
34. Какие шесть видов блокировок (lock) описаны в спецификации JPA (или какие есть значения у enum LockModeType в JPA)?
35. Какие два вида кэшей (cache) вы знаете в JPA и для чего они нужны?
36. Как работать с кешем 2 уровня?
37. Что такое JPQL/HQL и чем он отличается от SQL?
38. Что такое Criteria API и для чего он используется?
39. Расскажите про проблему N+1 Select и путях ее решения.
40. Что такое EntityGrpah? Как и для чего их использовать?
41. Какие шесть видов блокировок (lock) описаны в спецификации JPA (или какие есть значения у enum LockModeType в JPA)?

Spring

1. Что такое инверсия контроля (IoC) и внедрение зависимостей (DI)? Как эти принципы реализованы в Spring?
2. Что такое IoC контейнер?
3. Что такое Bean в спринге?
4. Расскажите про аннотацию @Bean?
5. Расскажите про аннотацию @Component?
6. Чем отличаются аннотации @Bean и @Component?
7. Расскажите про аннотации @Service и @Repository. Чем они отличаются?
8. Расскажите про аннотацию @Autowired
9. Расскажите про аннотацию @Resource
10. Расскажите про аннотацию @Inject
11. Расскажите про аннотацию @Lookup
12. Можно ли вставить бин в статическое поле? Почему?
13. Расскажите про аннотации @Primary и @Qualifier
14. Как заинжектировать примитив?
15. Как заинжектировать коллекцию?
16. Расскажите про аннотацию @Conditional
17. Расскажите про аннотацию @ComponentScan
18. Расскажите про аннотацию @Profile
19. Расскажите про ApplicationContext и BeanFactory, чем отличаются? В каких случаях что стоит использовать?
20. Расскажите про жизненный цикл бина, аннотации @PostConstruct и @PreDestroy()
21. Расскажите про скоупы бинов? Какой скоуп используется по умолчанию? Что изменилось в пятом спринге?
22. Расскажите про аннотацию @ComponentScan
23. Как спринг работает с транзакциями? Расскажите про аннотацию @Transactional.
24. Расскажите про аннотации @Controller и @RestController. Чем они отличаются? Как вернуть ответ со своим статусом (например 213)?
25. Что такое ViewResolver?
26. Чем отличаются Model, ModelMap и ModelAndView?
27. Расскажите про паттерн MVC, как он реализован в Spring?
28. Расскажите про паттерн Front Controller, как он реализован в Spring?
29. Что такое АОП? Как реализовано в спринге?
30. В чем разница между Filters, Listeners and Interceptors?
31. Можно ли передать в запросе один и тот же параметр несколько раз? Как?
32. Как работает Spring Security? Как сконфигурировать? Какие интерфейсы используются?
33. Что такое SpringBoot? Какие у него преимущества? Как конфигурируется? Подробно.

34. Расскажите про нововведения Spring 5.

Паттерны

1. Что такое «шаблон проектирования»?
2. Назовите основные характеристики шаблонов.
3. Назовите три основные группы паттернов.
4. Расскажите про паттерн Одиночка (Singleton).
5. Расскажите про паттерн Строитель (Builder).
6. Расскажите про паттерн Фабричный метод (Factory Method).
7. Расскажите про паттерн Абстрактная фабрика (Abstract Factory).
8. Расскажите про паттерн Прототип (Prototype).
9. Расскажите про паттерн Адаптер (Adapter).
10. Расскажите про паттерн Декоратор (Decorator).
11. Расскажите про паттерн Заместитель (Proxy).
12. Расскажите про паттерн Итератор (Iterator).
13. Расскажите про паттерн Шаблонный метод (Template Method).
14. Расскажите про паттерн Цепочка обязанностей (Chain of Responsibility).
15. Какие паттерны используются в Spring Framework?
16. Какие паттерны используются в Hibernate?

Алгоритмы и структуры данных

1. Что такое Big O? Как происходит оценка асимптотической сложности алгоритмов?
2. Что такое рекурсия? Сравните преимущества и недостатки итеративных и рекурсивных алгоритмов. С примерами.
3. Что такое жадные алгоритмы? Приведите пример.
4. Расскажите про пузырьковую сортировку.
5. Расскажите про быструю сортировку.
6. Расскажите про сортировку слиянием.
7. Что такое жадные алгоритмы? Приведите пример.
8. Расскажите про бинарное дерево.
9. Расскажите про красно-черное дерево.
10. Расскажите про линейный и бинарный поиск.
11. Расскажите про очередь и стек.
12. Сравните сложность вставки, удаления, поиска и доступа по индексу в ArrayList и LinkedList.