

Java Fundamentals

1. JVM-JDK-JRE. Что это такое? Кто кого включает и как взаимодействуют.
2. Как скомпилировать и запустить класс, используя консоль?
3. Что такое classpath. Если в classpath есть две одинаковые библиотеки (или разные версии одной библиотеки), объект класса из какой библиотеки создастся?
4. Какие области памяти использует java для размещения простых типов, объектов, ссылок, констант, методов, пул строк и т.д.
5. Пакеты в java. Зачем применяются? Принцип именования пакетов.
6. Модификаторы доступа.
7. Почему метод main() объявлен как public static void?
8. Что такое package level access. Пример использования.
9. Может ли объект получить доступ к private-переменной класса? Если, да, то каким образом?
10. Классы-оболочки.
11. Autoboxing и unboxing. Принцип действия на примерах.
12. `int a = 1;`
`Integer b = 2;`
`int c = a+b;` - результат?
А каков будет результат если `Integer b = null` ?
13. Что такое var? Достоинства и недостатки.

Java Strings

14. В чем разница между созданием строки как `new String()` и литералом (при помощи двойных кавычек)?
15. Как реализуется класс String, какие поля там есть?
16. Как работает метод `substring()` класса String?
17. Понятие Юникод. UTF-8, описание кодировки. Отличие от UTF-16.
18. String, StringBuilder, StringBuffer. Отличия.

Java Classes

19. Базовый класс в java. Методы класса.
20. OOP abstraction. Принципы ООП.
21. Правила переопределения метода `boolean equals(Object o)`.
22. Зачем переопределять методы `hashCode` и `equals` одновременно?
23. Написать метод `equals` для класса, содержащего одно поле типа String или StringBuilder.
24. Правила переопределения метода `int hashCode()`. Можно ли в качестве результата возвращать константу?
25. Правила переопределения метода `clone()`.
26. Чем отличаются `finally` и `finalize`? Для чего используется ключевое слово `final`?
27. JavaBeans: основные требования к классам Bean-компонентов, соглашения об именах.
28. Как работает Garbage Collector. Какие самые распространенные алгоритмы? Можно ли самому указать сборщику мусора, какой объект удалить из памяти.
29. В каких областях памяти хранятся значения и объекты, массивы?
30. Какие идентификаторы по умолчанию имеют поля интерфейса?
31. Чем отличается абстрактный класс от интерфейса

32. Когда применять интерфейс логичнее, а когда абстрактный класс?
33. Бывают ли интерфейсы без методов. Для чего?
34. Перегрузка и переопределение. Можно ли менять модификатор доступа метода, если да, то каким образом?
35. Перегрузка и переопределение. Можно ли менять возвращаемый тип метода, если да, то как? Можно ли менять тип передаваемых параметров?
36. Каким образом передаются переменные в методы, по значению или по ссылке?
37. Что такое конструктор по умолчанию?
38. Свойства конструктора. Способы его вызова.
39. Mutable и Immutable классы. Привести примеры из java core. Как создать класс, который будет immutable. Класс record.
40. static - что такое? Что будет, если значение атрибута изменить через объект класса? Всегда ли static поле содержит одинаковые значения для всех его объектов?
41. Внутренние классы, какие бывают и для каких целей используются. Области видимости данных при определенных ситуациях.
42. Анонимные классы. Практическое применение.
43. Generics. Что это такое и для чего применяются. Во что превращается во время компиляции и выполнения? Использование wildcards.
44. Что такое enum? Область применения. Какое использование перечисления некорректно? Привести примеры.
45. Отличия в применении интерфейсов Comparator и Comparable?
46. Класс Optional. Как помогает бороться с null?
47. Принципы SOLID, Yagni, Kiss, Dry.
48. Если Interf – это интерфейс, аClazz – это класс, который реализует интерфейс и в нем объявлен метод (которого нет в Interf), например, method(). Корректно ли выражение:
Interf a = new Clazz();
a.method();
То же самое, если Interf – не интерфейс, а абстрактный класс.

Java IO

49. Что такое сериализация, для чего нужна, когда применяется? Ключевое слово transient, для чего нужно? Сериализация static-полей.
50. (сериализация)Возможно ли сохранить объект не в байт-код, а в xml-файл?
51. Что такое ClassLoader? Если изменить static переменную в классе, загруженном одним ClassLoader, что будет видно в том же классе, загруженном другим. Возможно ли синглтоны создавать несколько раз?

Java Exceptions

52. Опишите иерархию исключений.
53. Что такое checked и unchecked Exception? Их отличия.
54. Опишите работу блока try-catch-finally. Может ли работать данный блок без catch.
55. Что такое Error. Перехват Error. Можно ли, есть ли смысл, в каких случаях возникает и что с ним делать.
56. Чем отличается OutOfMemoryError от StackOverflowError. Как их избежать?
57. Оператор throw. Как работает? Какие свойства?
58. try {
 a = 5;
 throw new Exception("1");
} catch (Exception e) {
 a = 10;
 throw new Exception("2");
}

```

    } finally {
        a = 15;
        throw new Exception("3");
    }

```

С каким сообщением будет сгенерировано исключение и какое значение примет а?

Java TestNG

59. Механизм assert. Когда возникает AssertionError? В чем его смысл?

60. JUnit. TestNG. Что это такое? Принципы написания.

Java & XML

61. XML – парсеры.

62. Что лучше использовать в каких случаях, well-formed, valid.

63. В чем отличие dtd и xsd, какая из этих схем написана в формате xml?

Многопоточность

64. Что такое процесс? Что такое поток? Состояния потока.

65. Как создать поток? Какими способами можно создать поток, запустить его, прервать (завершить, убить)?

66. Как выполнить набор команд в отдельном потоке?

67. Как работают методы wait и notify/notifyAll?

68. Чем отличается работа метода wait с параметром и без параметра?

69. Как работает метод yield()? Чем отличаются методы Thread.sleep() и Thread.yield()?

70. Чем отличаются методы Thread.sleep и wait?

71. Как работает метод join()?

72. Как правильно завершить работу потока? (Иногда говорят, убить поток).

73. Что такое синхронизация? Зачем она нужна? Для чего нужно ключевое слово synchronized? Какие методы синхронизации вы знаете? Какими средствами достигается?

74. Отличия работы synchronized от Lock?

75. Есть ли у Lock механизм, аналогичный механизму wait\notify у synchronized?

76. Что такое deadlock? Нарисовать схему, как это происходит.

77. Semaphore, CyclicBarrier, CountDownLatch. Чем похожи на Lock и чем от него отличаются?

78. Написать deadlock, придумать примеры с использованием synchronized, AtomicInteger

79. По каким объектам синхронизируются статические и нестатические методы?

80. Для чего применяется volatile? Пакет java.util.concurrent.atomic.

81. Есть массив из N-ти элементов. Создать N потоков, которые принимают по числу из массива, обрабатывают и возвращают обратно. Собрать все обработанные числа обратно в массив.

Java. Collection

82. Основные интерфейсы коллекций и их иерархия (List, Set, Queue). Какие бывают коллекции? В чём особенности разных видов коллекций? Когда какие стоит применять?

83. Что такое интерфейс Map? Представляет ли он коллекцию?

84. Сравнить ArrayList и LinkedList.

85. Сравнить HashMap и Hashtable.
86. Как устроены HashSet, TreeMap, TreeSet?
87. Принцип работы и реализации HashMap. Изменения HashMap в java8.
88. Чем отличается ArrayList от Vector?
89. Особенности интерфейса Set.
90. Как добавляются объекты в HashSet?
91. Какими способами можно отсортировать коллекцию? (привести три способа)
92. Как правильно удалить элемент из коллекции при итерации в цикле?
93. Как правильно удалить элемент из ArrayList (или другой коллекции) при поиске этого элемента в цикле?
94. Коллекции из пакета concurrent. Их особенности.
95. Что происходит при добавлении в ArrayList нового элемента и как это реализовано.
96. Thread-safe and non-thread safe collections.
97. Метод для преобразования потоко-небезопасной коллекции в потоко-безопасную.
98. Написать метод в котором проверяется HashMap на наличие в нем некоторого значения, и его извлечения, если такого значения нет, надо добавить значение с пустой строкой и её вернуть. Написать код, чтобы он был как можно более эффективным (меньше затратных действий).
99. Какие потокобезопасные коллекции более «быстрые» – legacy(Vector, Hashtable) или из пакета concurrent?
100. Если в коллекцию часто добавлять элементы, и удалять, какую лучше использовать? Почему? Как они устроены?
101. Как быстро получить копию коллекции. Записать код преобразования.

Functional Programming

102. Функциональные интерфейсы. Определение. Default & static методы. Область применения.
103. Лямбда-выражение. Замыкания. Синтаксис. Характеристики.
104. Function, Supplier, Predicate, Consumer. Их применение.
105. Ссылка на метод? Что это такое? Или это все же ссылка на объект?
106. Собственные функциональные интерфейсы.

Java. Streams

107. Чем Stream отличается от коллекции?
108. Прмежуточные и терминальные операции.
109. Методы: map() vs flatMap()
110. Что такое потоковая обработка данных.

JDBC

111. Как создать Connection?
112. Как правильно закрыть Connection?
113. Какие есть типы драйверов для соединения с СУБД?
114. Чем отличается Statement от PreparedStatement? Где сохраняется запрос после первого вызова PreparedStatement? Будет ли тот же самый эффект как и от PreparedStatement, если формировать запрос просто в строке и отправлять его в Statement?
115. Зачем нужен CallableStatement?
116. Отличие executeUpdate от executeQuery

117. Как в объекте ResultSet вернуться в предыдущую строку? Всегда ли можно вернуться в предыдущую строку?
118. Последовательность действий необходимых для выполнения запроса к БД.
119. Как получить сгенерированный СУБД первичный ключ без выполнения дополнительного запроса к БД?

Design Patterns

1. Зачем нужны паттерны? Привести примеры из проекта.
2. Какие паттерны вы знаете и как их применяет Java SE.
3. Factory Method, Builder
4. Singleton (class Runtime)
5. Как сделать чтобы в Singleton не работала двойная блокировка?
6. Как у Singleton создать второй объект? И как воспрепятствовать этому.
7. Prototype
8. Command
9. Composite
10. Chain of responsibility (Filter, closing io stream, closing connection)
11. State (Thread.State)
12. Iterator (Enumeration, Iterator, ListIterator)
13. Proxy
14. Observer (Listener-s)
15. Wrapper
16. Immutable
17. MVC
18. DAO vs. Repository

JEE

1. Что входит в JEE?
2. Что такое сервер приложений? Что такое веб-сервер, в чём его отличие от сервера приложений? Привести примеры веб-сервера и сервера приложений.
3. Что такое контейнер сервлетов? Что такое сервлет?
4. Методы сервлета. Жизненный цикл сервлета.
5. Что такое jsp. Жизненный цикл.
6. Что такое сессия? Жизненный цикл.
7. Что такое request? Из чего состоит? Жизненный цикл.
8. Cookies. Как можно достать Cookies, а если Cookies удалить, можно ли достать сессию?
9. Что нужно написать в браузерной строке, чтобы обратиться к сервлету? Можно ли из браузерной строки напрямую вызвать метод сервлета?
10. Чем отличаются методы POST и GET. Если не указать напрямую, какой из этих методов выполнится по умолчанию?
11. Как сделать redirect незаметно для пользователя?
12. Какие scopes (области видимости) переменных существуют в JSP?
13. В чём различие forward и redirect?
14. Отличия getAttribute() от getParameter() в сервлете.
15. Из чего состоит url?

16. Отличие `jsp:include` от директивы `include`.
17. Применение классов `HttpServletRequestWrapper` и `HttpServletResponseWrapper`.
18. Что делает `RequestDispatcher.include()`?
19. В какой последовательности выполняются сервлет фильтры?
20. Как обрабатывается тег с телом?
21. JSTL.
22. Что нужно написать в строке браузера, чтобы обратиться к хосту, на котором установлен tomcat, развёрнуто приложение, в котором есть несколько сервлетов? Как обратиться к конкретному сервлету? Что такое `www`? Где нужно указывать порт?
23. Можно ли в `web.xml` определить сервлет без указания `url` паттерна и как к нему обратиться?
24. Что такое HTTP? Отличия HTTP 1.0 и HTTP 2.
25. Сохраняет ли `http` протокол своё состояние.
26. Как сервер понимает, что для пользователя создана сессия и не нужно её создавать?
27. Отличие авторизации от аутентификации.
28. Сервер приложений. Веб-сервер. Отличия.
29. Пошагово рассказать, что происходит, когда пользователь нажимает на кнопку. (с формы поля сетаются в `request`, потом вызывается контейнер сервлетов, потом он как-то по `request` понимает, куда нужно идти дальше (в дескриптор развёртывания, а их может быть несколько, нужно как-то понимать в какой)).
30. Какие бывают `WebService`?
31. RESTful сервис. Принципы работы.

SQL

19. Что такое нормализация.
20. Какие есть типы связей в базе данных. Привести пример.
21. Что такое `primary key` (первичный ключ)?
22. Что такое `foreign key` (внешний ключ)?
23. Что такое индексы в базе данных? Для чего их используют? Чем они хороши и чем плохи?
24. Какие есть типы JOIN'ов. Кратко опишите каждый из типов.
25. Для чего используется слово `HAVING`? Отличия от `WHERE`.
26. Зачем нужно View и какие поля там будут.
27. Есть таблицы `Customer` и `Order`. Вывести всех `customer`, у которых суммарный заказ будет > 10000 .
28. Что такое агрегирующая функция, примеры.
29. Результат запроса *Select * from Table1, Table2;*
30. Есть 2 таблицы

IDDepartment	Department	
IDEmployer	Salary	IDDepartment

- Написать запросы на выборку `Department`, `salary` если сумма `salary > 100`;
- Написать запрос на ту же выборку через `join`;

Technology

Spring

1. Рассказать про Spring;
2. Spring inversion of control. Что такое. Как используется и для чего.
3. Spring dependency injection. Что такое. Как используется и для чего.

Hibernate

1. Уровни кеша;
2. Способы запросов;
3. Как из select * from a,b достать только объекты b;
4. Entities состояния;
5. Сессия в хибернейте;
6. sessionFactory.getCurrentSession() vs entityManagerFactory.createEntityManager()
7. У какого объекта (если использовать Hibernate) вызываются методы для работы с БД;
8. Если в хибернейте сделать:
Object o = transaction.get(id, table);
o.setName(name);
o.commit();
Сохранится ли name?

JMS

1. Что такое JMS

Ant & Maven

1. Различия Ant & Maven;
2. Команды в консоли для JVM, Maven, Ant;
3. Какой командой билдится проект?

Web services