1. Что такое ООП?
2. Какие преимущества у ООП?
3. Какие недостатки у ООП?
4. Назовите основные принципы ООП.
5. Что такое инкапсуляция? (c примером)
6. Что такое наследование? (c примером)
7. Что такое полиморфизм? (с примером)
8. Что такое абстракция (с примерами)?
9. Что такое ассоциация, агрегация и композиция?
10. Расскажите про раннее и позднее связывание.
11. Что такое SOLID?
12. Какие еще принципы можешь назвать?
13. Какая основная идея языка?
14. За счет чего обеспечивается кроссплатформенность?
15. Какие преимущества у JAVA?
16. Какие недостатки у Java?
17. Что такое JDK? Что в него входит?
18. Что такое JRE? что в него входит?
19. Что такое JVM?
20. Что такое byte code?
21. Что такое загрузчик классов (ClassLoader)?
22. Что такое JIT?
23. Что такое сборщик мусора? (Garbage collector)
24. Что такое Heap и Stack память в Java? Чем они отличаются?
25. Какие примитивные типы данных есть в Java?
26. Что такое char?
27. Сколько памяти занимает boolean?
28. Что такое классы-обертки?
29. Что такое автоупаковка и автораспаковка?
30. Что такое явное и неявное приведение типов (иначе называют кастинг)? В каких случаях в java нужно использовать явное приведение?
31. Что такое пул интов?
32. Какие нюансы у строк в Java?
33. Что такое пул строк?
34. Почему не рекомендуется изменять строки в цикле? Что рекомендуется использовать?
35. Почему строки не рекомендуется использовать для хранения паролей?
36. Почему String неизменяемый и финализированный класс?
37. Почему строка является популярным ключом в HashMap в Java?
38. Что делает метод intern() в классе String?
39. Можно ли использовать строки в конструкции switch?
40. Какая основная разница между String, StringBuffer, StringBuilder?
41. Существуют ли в java многомерные массивы?
42. Какими значениями инициируются переменные по умолчанию?
43. Что такое сигнатура метода?
44. Расскажите про метод main
45. Каким образом переменные передаются в методы, по значению или по ссылке?
46. Какие виды классов есть в java?
47. Расскажите про вложенные классы. В каких случаях они применяются?
48. Что такое «локальный класс»? Каковы его особенности?
49. Что такое «анонимные классы»? Где они применяются?
50. Каким образом из вложенного класса получить доступ к полю внешнего класса?
51. Что такое перечисления (enum)?
52. Как проблема ромбовидного наследования решена в java?
53. Могут ли быть приватные конструкторы? Для чего они нужны?
54. Расскажите про классы-загрузчики и про динамическую загрузку классов.
55. Чем отличаются конструкторы по умолчанию, конструктор копирования и конструктор с параметрами?
56. Какие модификаторы доступа есть в Java? Какие применимы к классам?
57. Какие модификаторы доступа применимы к классу верхнего уровня ? Модификаторы для переменных / Модификаторы для методов. Что означает модификатор static?
58. Может ли статический метод быть переопределён или перегружен?
59. Могут ли нестатические методы перегрузить статические?
60. Можно ли сузить уровень доступа/тип возвращаемого значения при переопределении метода?
61. Что можно изменить в сигнатуре метода при переопределении? Можно ли менять модификаторы (throws и тп)?
62. Могут ли классы быть статическими?
63. Что означает модификатор final? К чему он может быть применим?
64. Что такое абстрактные классы? Чем они отличаются от обычных?
65. Может ли быть абстрактный класс без абстрактных методов?
66. Могут ли быть конструкторы у абстрактных классов? Для чего они нужны?
67. Что такое интерфейсы? Какие модификаторы по умолчанию имеют поля и методы интерфейсов?
68. Чем интерфейсы отличаются от абстрактных классов? В каких случаях следует использовать абстрактный класс, а в каких интерфейс?
69. Может ли один интерфейс наследоваться от другого? От двух других?
70. Что такое дефолтные методы интерфейсов? Для чего они нужны?
71. Как решается проблема ромбовидного наследования при наследовании интерфейсов при наличии default методов?
72. Каков порядок вызова конструкторов и блоков инициализации с учетом иерархии классов?
73. Зачем нужны и какие бывают блоки инициализации?
74. Для чего в Java используются статические блоки инициализации?
75. Какое исключение выбрасывается при возникновении ошибки в блоке инициализации класса?
76. Что такое класс Object?
77. Что такое класс Class?
78. Какие методы есть у класса Object (перечислить все)? Что они делают?
79. Расскажите про equals и hashcode
80. Каким образом реализованы методы hashCode() и equals() в классе Object?
81. Зачем нужен equals(). Чем он отличается от операции ==?
82. Правила переопределения equals()
83. Что будет, если переопределить equals() не переопределяя hashCode()? Какие могут возникнуть проблемы?
84. Какой контракт между hashCode() и equals()?
85. Для чего нужен метод hashCode()?
86. Правила переопределения метода hashСode()
87. Есть ли какие-либо рекомендации о том, какие поля следует использовать при подсчете hashCode()?
88. Могут ли у разных объектов быть одинаковые hashCode()?
89. Почему нельзя реализовать hashcode() который будет гарантированно уникальным для каждого объекта?
90. Есть класс Point {int x, y;} Почему хэш-код в виде 31 \* x + y предпочтительнее чем x + y?
91. Чем a.getClass().equals(A.class) отличается от a instanceof A.class
92. Что такое исключения?
93. Опишите иерархию исключений.
94. Расскажите про обрабатываемые и необрабатываемые исключения
95. Можно ли обработать необрабатываемые исключения?
96. Какой оператор позволяет принудительно выбросить исключение?
97. О чем говорит ключевое слово throws?
98. Как создать собственное («пользовательское») исключение?
99. Расскажите про механизм обработки исключений в java (Try-catch-finally)
100. Возможно ли использование блока try-finally (без catch)?
101. Может ли один блок catch отлавливать сразу несколько исключений?
102. Всегда ли выполняется блок finally? Существуют ли ситуации, когда блок finally не будет выполнен?
103. Может ли метод main() выбросить исключение во вне и если да, то где будет происходить обработка данного исключения?
104. В каком порядке следует обрабатывать исключения в catch блоках?
105. Что такое механизм try-with-resources?
106. Что произойдет если исключение будет выброшено из блока catch после чего другое исключение будет выброшено из блока finally?
107. Что произойдет если исключение будет выброшено из блока catch после чего другое исключение будет выброшено из метода close() при использовании try-with-resources?
108. Что такое сериализация и как она реализована в Java?
109. Для чего нужна сериализация?
110. Опишите процесс сериализации/десериализации с использованием Serializable.
111. Как изменить стандартное поведение сериализации/десериализации?
112. Какие поля не будут сериализованы при сериализации? Будет ли сериализовано final поле?
113. Как создать собственный протокол сериализации?
114. Какая роль поля serialVersionUID в сериализации?
115. Когда стоит изменять значение поля serialVersionUID?
116. В чем проблема сериализации Singleton?
117. Расскажите про клонирование объектов.
118. В чем отличие между поверхностным и глубоким клонированием?
119. Какой способ клонирования предпочтительней?
120. Почему метод clone() объявлен в классе Object, а не в интерфейсе Cloneable?
121. Как создать глубокую копию объекта? (2 способа)
122. Что такое сборщик мусора? (Garbage collector)
123. Что такое сборщик мусора? (Garbage collector)
124. ТЫ ВЫИГРАЛ АВТОМОБИЛЬ
125. Чем процесс отличается от потока?
126. Чем Thread отличается от Runnable? Когда нужно использовать Thread, а когда Runnable?
127. Как выполнить две задачи параллельно?
128. Что такое монитор? Как монитор реализован в java?
129. Что такое синхронизация? Какие способы синхронизации существуют в java?
130. Как работают методы Object wait(), notify() и notifyAll()?
131. В каких состояниях может находиться поток?
132. Что такое семафор? Как он реализован в Java?
133. Что обозначает ключевое слово volatile? Почему операции над volatile переменными не атомарны?
134. Расскажи про правила «happens-before» и Java Memory Model (JMM).
135. Для чего нужны Atomic типы данных? Чем отличаются от volatile?
136. Что такое потоки демоны? Для чего они нужны? Как создать поток-демон?
137. Что такое приоритет потока? На что он влияет? Какой приоритет у потоков по умолчанию?
138. Как работает Thread.join()? Для чего он нужен?
139. Чем отличаются методы yield() и sleep()?
140. Как правильно остановить поток? Для чего нужны методы .stop(), .interrupt(), .interrupted(), .isInterrupted().
141. Чем Runnable отличается от Callable?
142. Что такое FutureTask?
143. Что такое deadlock?
144. Что такое livelock?
145. Что такое race condition?
146. Что такое Фреймворк fork/join? Для чего он нужен?
147. Что означает ключевое слово synchronized? Где и для чего может использоваться?
148. Что является монитором у статического синхронизированного класса?
149. Что является монитором у не статического синхронизированного класса?
150. Способы управления потоками java.util.сoncurrent
151. Расскажи про устройство и алгоритм работы ConcurrentHashMap
152. Что знаешь про CopyOnWriteArrayList?
153. Stream API & ForkJoinPool Как связаны, что это такое.
154. Зачем используются thread local переменные?
155. Как реализовать двусторонний обмен данными между потоками?
156. Какими коллекциями пользоваться в многопоточной среде?

ИДИ ПОСПИ

1. Что такое данные, информация, база данных? Что «под капотом» БД?
2. Что такое SQL?
3. Что такое Query?
4. Какие бывают связи таблиц SQL?
5. Что такое DDL? Какие операции в него входят? Рассказать про них.
6. Что такое DML? Какие операции в него входят? Рассказать про них.
7. Что такое TCL? Какие операции в него входят? Рассказать про них.
8. Что такое DCL? Какие операции в него входят? Рассказать про них.
9. Нюансы работы с NULL в SQL. Как проверить поле на NULL?
10. Виды Join’ов (виды связывания таблиц).
11. Что лучше использовать join или подзапросы? Почему?
12. Что делает UNION?
13. Чем WHERE отличается от HAVING?
14. Что такое GROUP BY?
15. Что такое ORDER BY?
16. Что такое DISTINCT? Что такое LIMIT?
17. Что такое EXISTS?
18. Расскажите про операторы IN, BETWEEN, LIKE.
19. Что делает оператор MERGE? Какие у него есть ограничения?
20. Какие агрегатные/агрегирующие функции вы знаете?
21. Что такое ограничения (constraints)? Какие вы знаете?
22. Что такое суррогатные ключи?
23. Теория множеств
24. Какой вид структуры данных (какое дерево) используется при работе с базами данных?
25. Как оценить сложность выполняемого запроса?
26. Как узнать, сколько места занимает таблица в памяти?
27. Методы сканирования таблиц
28. Что такое индексы? Какие они бывают?
29. Как и зачем создавать индекс?
30. Что такое селективность?
31. Анализ выполнения запросов (если есть проблемы с производительностью)
32. Чем TRUNCATE отличается от DELETE?
33. Что такое хранимые процедуры? Для чего они нужны?
34. Что такое представления (VIEW)? Для чего они нужны?
35. Что такое временные таблицы? Для чего они нужны?
36. Что такое транзакции? Расскажите про принципы ACID.
37. Расскажите про уровни изолированности транзакций.
38. Что такое нормализация и денормализация? Расскажите про 3 нормальные
39. формы?
40. Что такое TIMESTAMP?
41. Расскажи про шардирование баз данных
42. Как сделать запрос из двух баз?
43. Что такое триггер?
44. Что такое sql-injection (SQL инъекции)?
45. Как выбрать между Statement, PreparedStatement и CallableStatement?
46. Какие классы вовлечены в соединение с базой данных?
47. Что можно делать с классом Connection?
48. Какая разница между @⁠ElementCollection, @⁠OneToMany и @⁠ManyToMany?
49. SQL live coding на собеседовании
50. Что такое ORM?
51. Что такое JPA?
52. Что появилось раньше JPA или Hibernate?
53. Что такое Hibernate?
54. Какие ключевые интерфейсы использует Hibernate?
55. PROXY: что это такое, для чего и как применяется в Hibernate.
56. Расскажи про Hibernate кэширование (уровни кэша и что под капотом)
57. Как работать с кэшем второго уровня?
58. Что такое Persistence Context?
59. Что такое EntityManager?
60. Методы интерфейса EntityManager
61. Какие функции он выполняет?
62. Каким условиям должен удовлетворять класс чтобы являться Entity?
63. Расскажи про жизненный цикл сущности Entity, перечисли четыре статуса ЖЦ Entity объекта (Entity Instance’s Life Cycle).
64. Как влияют операции persist, remove, merge, refresh, detach на Entity объекты каждого из четырех статусов?
65. Может ли абстрактный класс быть Entity?
66. Может ли Entity класс наследоваться от не Entity классов (non-entity classes)?
67. Может ли Entity класс наследоваться от других Entity классов?
68. Может ли не Entity класс наследоваться от Entity класса?
69. Что такое встраиваемый (Embeddable) класс?
70. Какие требования JPA устанавливает к встраиваемым (Embeddable) классам?
71. Для чего нужны аннотации @Embedded и @Embeddable?
72. Что такое Mapped Superclass?
73. Какие стратегии маппинга иерархии наследования (Inheritance Mapping Strategies) описаны в JPA?
74. Как мапятся Enum -ы?
75. Как мапятся даты (до Java 8 и старше)?
76. Как Hibernate работает с разными типами?
77. Как сохранять в базе данных коллекции базовых типов?
78. Какие есть виды связей (маппинг ассоциаций)?
79. Что такое владелец связи?
80. Что такое каскады?
81. Какие два типа fetch стратегии в JPA вы знаете?
82. Для чего нужна аннотация Basic?
83. Для чего нужна аннотация Column?
84. Для чего нужна аннотация Access?
85. Для чего нужна аннотация Cacheable?
86. Каких провайдеров кэша второго уровня знаешь?
87. Как смапить составной ключ?
88. Для чего нужна аннотация ID?
89. Какие @GeneratedValue вы знаете?
90. Расскажите про аннотации @JoinColumn и @JoinTab[le](https://vk.com/joincolumn)? Где и для чего они используются?
91. Для чего нужны аннотации @OrderBy и @OrderColumn, чем они отличаются?
92. Для чего нужна аннотация Transient?
93. Какие шесть видов блокировок (lock) описаны в спецификации JPA (или какие есть значения у enum LockModeType в JPA)?
94. Какие два вида кэшей (cache) вы знаете в JPA и для чего они нужны?
95. Как работать с кэшем 2 уровня?
96. Что такое JPQL/HQL и чем он отличается от SQL?
97. Что означает полиморфизм (polymorphism) в запросах JPQL (Java Persistence query language) и как его «выключить»?
98. Что такое Criteria API и для чего он используется?
99. В чем разница в требованиях к Entity в Hibernate, от требований к Entity, указанных в спецификации JPA?
100. Расскажите про проблему N+1 Select и путях ее решения.
101. Что такое EntityGraph? Как и для чего их использовать?
102. Можно ли в JDBC реализовать кэш?
103. Какие интерфейсы, классы есть в JDBC?
104. Если установить стратегию генерации ключа AUTO, то какую из типов стратеги Hibernate САМ никогда не выберет и почему?
105. Из каких основных сущностей состоит Spring-приложение?
106. Что такое инверсия контроля (IoC) и внедрение зависимостей (DI)?
107. Как эти принципы реализованы в Spring?
108. Что такое IoC контейнер?
109. Что такое Bean в спринге?
110. Расскажите про аннотацию @Bean?
111. Расскажите про аннотацию @Component?
112. Чем отличаются аннотации @Bean и @Component?
113. Расскажите про аннотации @Service и @Repository. Чем они отличаются?
114. Расскажите про аннотацию @Autowired
115. Расскажите про аннотацию @Resource
116. Расскажите про аннотацию @Inject
117. Расскажите про аннотацию @Lookup
118. \*Как работает инъекция прототипа в синглтон?
119. Можно ли вставить бин в статическое поле? Почему?
120. Расскажите про аннотации @Primary и @Qualifier
121. Как заинжектить примитив?
122. Как заинжектить коллекцию?
123. Расскажите про аннотацию @Conditional
124. Расскажите про аннотацию @Profile
125. Расскажите про ApplicationContext и BeanFactory, чем отличаются? В каких случаях что стоит использовать?
126. Расскажите про жизненный цикл бина, аннотации @PostConstruct и @PreDestroy
127. Расскажите про скоупы бинов? Какой скоуп используется по умолчанию?
128. Что изменилось в пятом спринге?
129. Расскажите про аннотацию @ComponentScan
130. Как спринг работает с транзакциями?
131. Расскажите про аннотацию @Transactional
132. Расскажите про аннотации @Controller и @RestController. Чем они отличаются?
133. Как вернуть ответ со своим статусом (например, 213)?
134. Что такое ViewResolver?
135. Чем отличаются Model, ModelMap и ModelAndView?
136. Расскажите про паттерн MVC, как он реализован в Spring?
137. Расскажите про паттерн Front Controller, как он реализован в Spring?
138. Что такое AOP? Как реализовано в спринге?
139. PROXY что это, для чего и как применяется в Spring
140. В чем разница между Filters, Listeners and Interceptors?
141. Можно ли передать в запросе один и тот же параметр несколько раз? Как?
142. Как работает Spring Security? Как сконфигурировать? Какие интерфейсы используются?
143. Что такое SpringBoot? Какие у него преимущества? Как конфигурируется? Подробно.
144. Какие задачи решает Spring Data?
145. Какова иерархия интерфейсов/классов репозитория в Spring Data JPA?
146. \*Как преодолеть проблему блокирующих вызовов?
147. Как вызвать транзакционный метод из того же класса?
148. \*Когда отрабатывает @Autowired а когда @Transactional? почему?
149. Почему нельзя отработать @Transactional в другом методе?
150. Структура HTTP запроса, ответа
151. HTTP методы
152. Ключевое отличие методов POST и GET
153. REST стиль
154. Идемпотентность: что это такое, методы
155. Как метод пост сделать идемпотентным?
156. Что такое «шаблон проектирования»?
157. Приведите примеры основных шаблонов проектирования.
158. Назовите основные характеристики шаблонов.
159. Назовите три основные группы паттернов (классификация)
160. Расскажите про паттерн Одиночка (Singleton).
161. Почему синглтон называют анти-паттерном?
162. Расскажите про паттерн Строитель (Builder).
163. Расскажите про паттерн Фабричный метод (Factory Method).
164. Расскажите про паттерн Абстрактная фабрика (Abstract Factory).
165. Расскажите про паттерн Прототип (Prototype).
166. Расскажите про паттерн Адаптер (Adapter).
167. Расскажите про паттерн Декоратор (Decorator).
168. Расскажите про паттерн Заместитель (Proxy).
169. Расскажите про паттерн Итератор (Iterator).
170. Расскажите про паттерн Шаблонный метод (Template Method).
171. Расскажите про паттерн Цепочка обязанностей (Chain of Responsibility).
172. Какие паттерны используются в Spring Framework?
173. Какие паттерны используются в Hibernate?
174. Что такое Dependency Injection?
175. Что такое «антипаттерн»? Какие антипаттерны знаете?
176. Еще что использовали на практике?
177. Стратегии обработки ошибок: Circuit Breaker pattern (автоматический выключатель, предохранитель)
178. Что такое Big O?
179. Как происходит оценка асимптотической сложности алгоритмов?
180. Что такое рекурсия?
181. Сравните преимущества и недостатки итеративных и рекурсивных алгоритмов. С примерами.
182. Что такое жадные алгоритмы? Приведите пример.
183. Расскажите про пузырьковую сортировку.
184. Расскажите про сортировку слиянием
185. Расскажите про быструю сортировку.
186. Расскажите про бинарное дерево.
187. Расскажите про красно-чёрное дерево.
188. Расскажите про линейный и бинарный поиск.
189. Расскажите про очередь и стек.
190. Сравните сложность вставки, удаления, поиска и доступа по индексу в ArrayList и LinkedList.