1. Базовые типы данных в Java (byte, word, int, long, float, double, boolean, char). Их характеристика, диапазоны значений.
2. На какие основные группы можно поделить типы данных в Java?
3. Какие примитивные типы в Java вы знаете?
4. Что вы знаете о преобразовании примитивных типов данных, есть ли потеря данных, можно ли преобразовать логический тип?
5. Какими значениями инициализируются переменные по умолчанию?
6. Как передается значение переменной (по ссылке/значению)?
7. Что вы знаете о методе main, какие обязательные условия его определения?
8. Какие логические операции и операторы вы знаете?
9. Что такое тернарный оператор выбора?
10. Какие унарные и бинарные арифметические операции вы знаете?
11. Какие побитовые операции вы знаете?
12. Какова роль и правила написания оператора выбора (switch)?
13. Какие циклы вы знаете, в чем их отличия?
14. Какие параметры имеет цикл for, можно ли их не задать?
15. Какой оператор используется для немедленной остановки цикла?
16. Какой оператор используется для перехода к следующей итерации цикла?
17. Что такое массив?
18. Что вы знаете о классах оболочках (Wrappers)?
19. Что такое автоупаковка (boxing/unboxing)?
20. Как правильно организовать доступ к полям класса?
21. Дайте определение понятию “конструктор”.
22. Чем отличаются конструкторы по-умолчанию конструктор с параметрами?
23. Какие модификации уровня доступа к членам класса вы знаете, расскажите про каждый из них.
24. О чем говорят ключевые слова “this”, “super”, где и как их можно использовать?
25. Дайте определение понятию “метод”.
26. Что такое сигнатура метода?
27. Какие методы называются перегруженными?
28. Могут ли нестатические методы перегрузить статические?
29. Может ли метод принимать разное количество параметров (аргументы переменной длины)?
30. Как получить доступ к переопределенным методам родительского класса?
31. Где можно инициализировать статические/нестатические поля?
32. Зачем нужен оператор instanceof?
33. Зачем нужны и какие бывают блоки инициализации?
34. Каков порядок вызова конструкторов и блоков инициализации двух классов: потомка и его предка?
35. Где и для чего используется модификатор abstract?
36. Можно ли объявить метод абстрактным и статическим одновременно?
37. Что означает ключевое слово static?
38. К каким конструкциям Java применим модификатор static?
39. Что будет, если в static блоке кода возникнет исключительная ситуация?
40. Что такое статический класс, какие особенности его использования?
41. Какие особенности инициализации final static переменных?
42. Как влияет модификатор static на класс/метод/поле?
43. О чем говорит ключевое слово final?
44. Дайте определение понятию “интерфейс”.
45. Какие модификаторы по умолчанию имеют поля и методы интерфейсов?
46. Почему нельзя объявить метод интерфейса с модификатором final или static?
47. Какие типы классов бывают в java (вложенные… и.т.д.)
48. Какие особенности создания вложенных классов: простых и статических.
49. Что вы знаете о вложенных классах, зачем они используются? Классификация, варианты использования, о нарушении инкапсуляции.
50. В чем разница вложенных и внутренних классов?
51. Какие классы называются анонимными?
52. Каким образом из вложенного класса получить доступ к полю внешнего класса?
53. Каким образом можно обратиться к локальной переменной метода из анонимного класса, объявленного в теле этого метода? Есть ли какие-нибудь ограничения для такой переменной?
54. Как связан любой пользовательский класс с классом Object?
55. Расскажите про каждый из методов класса Object.
56. Что такое метод equals(). Чем он отличается от операции ==.
57. Если вы хотите переопределить equals(), какие условия должны удовлетворяться для переопределенного метода?
58. Если equals() переопределен, есть ли какие-либо другие методы, которые следует переопределить?
59. В чем особенность работы методов hashCode и equals? Каким образом реализованы методы hashCode и equals в классе Object? Какие правила и соглашения существуют для реализации этих методов? Когда они применяются?
60. Какой метод возвращает строковое представление объекта?
61. Что будет, если переопределить equals не переопределяя hashCode? Какие могут возникнуть проблемы?
62. Есть ли какие-либо рекомендации о том, какие поля следует использовать при подсчете hashCode?
63. Как вы думаете, будут ли какие-то проблемы, если у объекта, который используется в качестве ключа в hashMap изменится поле, которое участвует в определении hashCode?
64. Чем отличается абстрактный класс от интерфейса, в каких случаях что вы будете использовать?
65. Можно ли получить доступ к private переменным класса и если да, то каким образом?
66. Что такое volatile и transient?
67. Расширение модификаторов при наследовании, переопределении и сокрытии методов. Если у класса-родителя есть метод, объявленный как private, может ли наследник расширить его видимость? А если protected? А сузить видимость?
68. Имеет ли смысл объявлять метод private final?
69. Какие особенности инициализации final переменных?
70. Что будет, если единственный конструктор класса объявлен как final?
71. Что такое finalize? Зачем он нужен? Что Вы можете рассказать о сборщике мусора и алгоритмах его работы.
72. Почему метод clone объявлен как protected? Что необходимо для реализации клонирования?
73. Синтаксис описания класса в языке Java. Модификаторы оператора class.
74. Описание полей в классе языка Java. Модификаторы описания поля.
75. Описание методов в классе языка Java. Модификаторы описания методов.
76. Особенности наследования классов в языке Java.
77. Интерфейсы в языке Java. Назначение интерфейсов. Синтаксис описания интерфейсов.
78. Реализация интерфейсов классами привести пример.
79. Интерфейс Comparable. Его назначение и характеристики. Привести пример использования.
80. Класс Object. Его назначение, характеристики. Основные методы.
81. Метод equals класса Object. Привести пример его переопределения в наследнике класса Object.
82. Метод toString класса Object. Привести пример его переопределения в наследнике класса Object.
83. Класс Class. Его назначение и характеристики.
84. Дайте определение понятию “исключение”
85. Какова иерархия исключений.
86. Можно/нужно ли обрабатывать ошибки jvm?
87. Какие существуют способы обработки исключений?
88. О чем говорит ключевое слово throws?
89. В чем особенность блока finally? Всегда ли он исполняется?
90. Может ли не быть ни одного блока catch при отлавливании исключений?
91. Могли бы вы придумать ситуацию, когда блок finally не будет выполнен?
92. Может ли один блок catch отлавливать несколько исключений (с одной и разных веток наследований)?
93. Что вы знаете об обрабатываемых и не обрабатываемых (checked/unchecked) исключениях?
94. В чем особенность RuntimeException?
95. Как написать собственное (“пользовательское”) исключение? Какими мотивами вы будете руководствоваться при выборе типа исключения: checked/unchecked?
96. Какой оператор позволяет принудительно выбросить исключение?
97. Есть ли дополнительные условия к методу, который потенциально может выбросить исключение?
98. Может ли метод main выбросить исключение во вне и если да, то где будет происходить обработка данного исключения?
99. Если оператор return содержится и в блоке catch и в finally, какой из них “главнее”?
100. Какая конструкция используется в Java для обработки исключений?
101. Предположим, есть блок try-finally. В блоке try возникло исключение и выполнение переместилось в блок finally. В блоке finally тоже возникло исключение. Какое из двух исключений “выпадет” из блока try-finally? Что случится со вторым исключением?
102. Предположим, есть метод, который может выбросить IOException и FileNotFoundException в какой последовательности должны идти блоки catch? Сколько блоков catch будет выполнено?
103. Дайте определение понятию “коллекция”.
104. Назовите преимущества использования коллекций.
105. Какие данные могут хранить коллекции?
106. Какова иерархия коллекций?
107. Что вы знаете о коллекциях типа List?
108. Что вы знаете о коллекциях типа Set?
109. Что вы знаете о коллекциях типа Queue?
110. Что вы знаете о коллекциях типа Map, в чем их принципиальное отличие?
111. Назовите основные реализации List, Set, Map.
112. Какие реализации SortedSet вы знаете и в чем их особенность?
113. В чем отличия/сходства List и Set?
114. Что разного/общего у классов ArrayList и LinkedList, когда лучше использовать ArrayList, а когда LinkedList?
115. В каких случаях разумно использовать массив, а не ArrayList?
116. Чем отличается ArrayList от Vector?
117. Что вы знаете о реализации классов HashSet и TreeSet?
118. Чем отличаются HashMap и TreeMap? Как они устроены и работают? Что со временем доступа к объектам, какие зависимости?
119. Что такое Hashtable, чем она отличается от HashMap? На сегодняшний день она deprecated, как все-таки использовать нужную функциональность?
120. Что будет, если в Map положить два значения с одинаковым ключом?
121. Как задается порядок следования объектов в коллекции, как отсортировать коллекцию?
122. Дайте определение понятию “итератор”.
123. Какую функциональность представляет класс Collections?
124. Как получить не модифицируемую коллекцию?
125. Какие коллекции синхронизированы?
126. Почему Map не наследуется от Collection?
127. Что такое стек и очередь, расскажите в чем их отличия?
128. В чем разница между интерфейсами Comparable и Comparator?
129. Какие существуют виды потоков ввода/вывода?
130. Назовите основные предки потоков ввода/вывода.
131. Что общего и чем отличаются следующие потоки: InputStream, OutputStream, Reader, Writer?
132. Что вы знаете о RandomAccessFile?
133. Какие есть режимы доступа к файлу?
134. В каких пакетах лежат классы-потоки?
135. Какой класс-надстройка позволяет читать данные из входного байтового потока в формате примитивных типов данных?
136. Какой класс-надстройка позволяет ускорить чтение/запись за счет использования буфера?
137. Какие классы позволяют преобразовать байтовые потоки в символьные и обратно?
138. Какой класс предназначен для работы с элементами файловой системы (ЭФС)?
139. Какой символ является разделителем при указании пути к ЭФС?
140. Что такое сериализация?
141. Какие условия “благополучной” сериализации объекта?
142. Какие классы позволяют архивировать объекты?
143.  Что такое Аннотации в Java?
144. Какие функции выполняет Аннотации?
145. Какие встроенные аннотации в Java вы знаете?
146. Что делают аннотации @Retention, @Documented, @Target и @Inherited?
147. Что делают аннотации @Override, @Deprecated, @SafeVarargs и @SuppressWarnings?
148. Какой жизненный цикл аннотации можно указать с помощью @Retention?
149. К каким элементам можно применять аннотацию, как это указать?
150. Как создать свою Аннотацию?
151. Что такое пакет в Java? Правила работы с пакетами. Операторы import. Привести пример приложения, состоящего из нескольких пакетов.
152. Что такое параметр общего типа?
153. Каковы некоторые преимущества использования универсальных типов?
154. Что такое параметр ограниченного типа?
155. Что такое верхний ограниченный шаблон?
156. Что такое подстановочный знак снизу?