1. Назовите принципы ООП и расскажите о каждом.
2. Дайте определение понятию «класс».
3. Что такое поле/атрибут класса?
4. Как правильно организовать доступ к полям класса?
5. Дайте определение понятию «конструктор».
6. Чем отличаются конструкторы по-умолчанию, копирования и конструктор с параметрами?
7. Какие модификации уровня доступа вы знаете, расскажите про каждый из них.
8. Расскажите об особенностях класса с единственным закрытым (private) конструктором.
9. О чем говорят ключевые слова «this», «super», где и как их можно использовать?
10. Дайте определение понятию «метод».
11. Что такое сигнатура метода?
12. Какие методы называются перегруженными?
13. Могут ли нестатические методы перегрузить статические?
14. Расскажите о переопределение методов.
15. Может ли метод принимать разное количество параметров (аргументы переменной длины)?
16. Можно ли сузить уровень доступа/тип возвращаемого значения при переопределении метода?
17. Как получить доступ к переопределенным методам родительского класса?
18. Какие преобразования называются нисходящими и восходящими?
19. Чем отличается переопределение от перегрузки?
20. Где можно инициилизировать статические/нестатические поля?
21. Зачем нужен оператор instanceof?
22. Зачем нужны и какие бывают блоки инициализации?
23. Каков порядок вызова конструкторов и блоков инициализации двух классов: потомка и его предка?
24. Где и для чего используется модификатор abstract?
25. Можно ли объявить метод абстрактым и статическим одновременно?
26. Что означает ключевое поле static?
27. К каким конструкциям Java применим модификатор static?
28. Что будет, если в static блоке кода возникнет исключительная ситуация?
29. Можно ли перегрузить static метод?
30. Что такое статический класс, какие особенности его использования?
31. Какие особенности инициализации final static переменных?
32. Как влияет модификатор static на класс/метод/поле?
33. О чем говорит ключевое слово final?
34. Дайте определение понятию «интерфейс».
35. Какие модификаторы по умолчанию имеют поля и методы интерфейсов?
36. Почему нельзя объявить метод интерфейса с модификатором final или static?
37. Какие типы классов бывают в java (вложенные… и.т.д.)
38. Что вы знаете о вложенных классах, зачем они используются? Классификация, варианты использования, о нарушении инкапсуляции.
39. Каким образом из вложенного класса получить досуп к полю внешнего класса?
40. Какие особенности создания вложенных классов: простых и статических.
41. В чем разница вложенных и внутренних классов?
42. Какие классы называются анонимными?
43. Каким образом можно обратиться к локальной переменной метода из анонимного класса, объявленного в теле этого метода? Есть ли каке-нибудь ограничения для такой переменной?
44. Как связан любой пользовательский класс с классом Object?
45. Расскажите про каждый из методов класса Object.
46. Что такое метод equals(). Чем он отличается от операции ==.
47. Если вы хотите переопределить equals(), какие условия должны удовлетворяться для переопределенного метода?
48. Если equals() переопределен, есть ли какие-либо другие методы, которые следует переопределить?
49. 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?
50. Какой метод возвращает строковое представление объекта?
51. Что будет, если переопределить equals не переопределяя hashCode? Какие могут возникнуть проблемы?
52. Есть ли какие-либо рекомендации о том, какие поля следует использовать при подсчете hashCode?
53. 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?
54. Чем отличается абстрактый класс о интерфейса, в каких случаях что вы будете использовать?
55. Можно ли получить доступ к private переменным классам и если да, то каким образом?
56. Модификаторы. Назначение и варианты использования.
57. Что такое volatile и transient? Для чего и в каких случаях можно было бы использовать default?
58. Контексты использования модификаторов (класс/поле/метод)
59. Какой из модификаторов более строгий: protected или package-private?
60. Расширение модификаторов при наследовании, переопределение и сокрытие методов. Если у класса-родителя есть метод, объявленный как private, может ли наследник расширить его видимость? А если protected? А сузить видимость?
61. Модификатор abstract и final для классов/методов
62. Имеет ли смысл объявлять метод private final?
63. Какие особенности инициализации final переменных?
64. Что будет, если единственный конструктор класса объявлен как final?
65. Расширение/сужение типов в аргументах переопределяемых методов в случае, если бы в Java список формальных параметров не входил бы в сигнатуру метода.
66. Что такое finalize? Зачем он нужен? Что Вы можете рассказать о сборщике мусора и алгоритмах его работы.
67. Почему метод clone объявлен как protected? Что необходимо для реализации клонирования?
68. Знакомы ли Вам какие-либо паттерны проектирования?
69. Напишите Singleton… А с ленивой загрузкой. А если он должен быть потоко-безопасным? А в каких случаях ленивая загрузка хуже?
70. Что можете сказать про MVC? Нарисуйте диаграмму и объясните, как MVC работает.
71. Напишите функцию вычисления факториала.
72. Дана функция вычисления чисел Фибоначчи, известно, что она работает. Найдите логическую ошибку. Оцените сложность получившегося алгоритма.