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?

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?

**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 (иначе называют кастинг)? 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 инициируются 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?

**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. Могут ли быть приватные конструкторы? Для чего они нужны?
9. Расскажите про классы-загрузчики и про динамическую загрузку классов.
10. Чем отличаются конструкторы по умолчанию, конструктор копирования и конструктор с параметрами?
11. Какие модификаторы доступа есть в Java? Какие применимы к классам?
12. Какие модификаторы доступа применимы к классу верхнего уровня ? Модификаторы для переменных / Модификаторы для методов. Что означает модификатор static?
13. Может ли статический метод быть переопределён или перегружен?
14. Могут ли нестатические методы перегрузить статические?
15. Можно ли сузить уровень доступа/тип возвращаемого значения при переопределении метода?
16. Что можно изменить в сигнатуре метода при переопределении? Можно ли менять модификаторы (throws и тп)?
17. Могут ли классы быть статическими?
18. Что означает модификатор final? К чему он может быть применим?
19. Что такое абстрактные классы? Чем они отличаются от обычных?
20. Может ли быть абстрактный класс без абстрактных методов?
21. Могут ли быть конструкторы у абстрактных классов? Для чего они нужны?
22. Что такое интерфейсы? Какие модификаторы по умолчанию имеют поля и методы интерфейсов?
23. Чем интерфейсы отличаются от абстрактных классов? В каких случаях следует использовать абстрактный класс, а в каких интерфейс?
24. Может ли один интерфейс наследоваться от другого? От двух других?
25. Что такое дефолтные методы интерфейсов? Для чего они нужны?
26. Как решается проблема ромбовидного наследования при наследовании интерфейсов при наличии default методов?
27. Каков порядок вызова конструкторов и блоков инициализации с учетом иерархии классов?
28. Зачем нужны и какие бывают блоки инициализации?
29. Для чего в Java используются статические блоки инициализации?
30. Какое исключение выбрасывается при возникновении ошибки в блоке инициализации класса?
31. Что такое класс Object?
32. Что такое класс Class?
33. Какие методы есть у класса Object (перечислить все)? Что они делают?
34. Расскажите про equals и hashcode
35. Каким образом реализованы методы hashCode() и equals() в классе Object?
36. Зачем нужен equals(). Чем он отличается от операции ==?
37. Правила переопределения equals()
38. Что будет, если переопределить equals() не переопределяя hashCode()? Какие могут возникнуть проблемы?
39. Какой контракт между hashCode() и equals()?
40. Для чего нужен метод hashCode()?
41. Правила переопределения метода hashСode()
42. Есть ли какие-либо рекомендации о том, какие поля следует использовать при подсчете hashCode()?
43. Могут ли у разных объектов быть одинаковые hashCode()?
44. Почему нельзя реализовать hashcode() который будет гарантированно уникальным для каждого объекта?
45. Есть класс Point {int x, y;} Почему хэш-код в виде 31 \* x + y предпочтительнее чем x + y?
46. Чем a.getClass().equals(A.class) отличается от a instanceof A.class

**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?

**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)
15. Что такое сборщик мусора? (Garbage collector)

**Garbage collector**

1. Что такое сборщик мусора? (Garbage collector)