**Java Core**

Основы java

**Java:**

**Вопросы по теме Особенности Java**

Попробуйте ответить на следующие вопросы:

1. В чем особенность языка Java? Почему он такой распространенный?
2. Какие преимущества есть у языка Java перед другими языками программирования?
3. Что необходимо для запуска Вашего java-приложения на устройстве пользователя?
4. За что отвечает JVM?
5. Что такое байткод?

**Особенности Java**

Типизация – это особенность работы языка программирования с различными типами переменных.

Строгая (сильная) типизация – не позволяет смешивать несовместимые типы данных. Например, в JavaScript можно слаживать int + boolean или String + int, в Java такого делать нельзя. Вы можете сказать, как так? Мы же можем спокойно смешивать int + char, в то время как int и char не совместимы. Дело в том, что в char хранится код символа в виде числового значения и с этим кодом можно производить арифметические операции.

Статическая типизация – проверка типов данных производится на стадии компиляции.

int a = 1;

a = 1.1; // ошибка

Явная типизация – нужно явно указывать, какого типа данных будет создаваемая переменная.

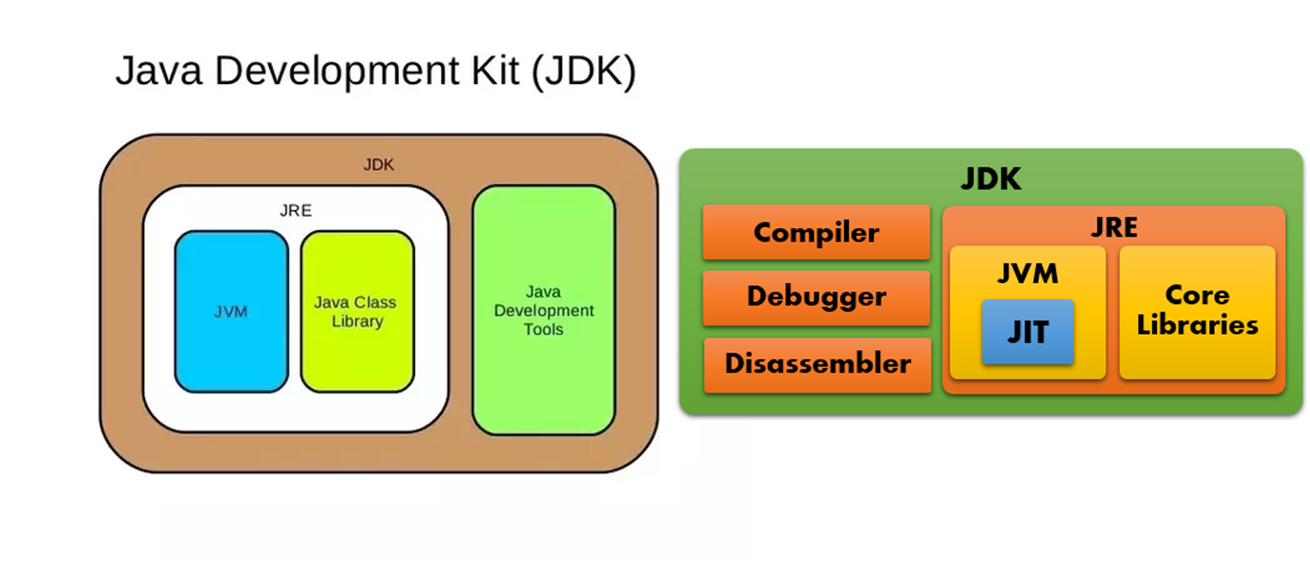
int a = 1;

ООП – Java использует объектно-ориентированную методологию программирования.

Байткод - Java код транслируется в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, для которой существует реализация виртуальной Java-машины (JVM). Достоинством подобного способа выполнения программ является полная независимость байт-кода от [операционной системы](https://ru.wikipedia.org/wiki/Операционная_система) и [оборудования](https://ru.wikipedia.org/wiki/Аппаратная_платформа), что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина.

Система безопасности - гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

**Что такое JRE, JIT, JDK, JVM**



JDK - комплект разработчика приложений, включающий в себя компилятор Java, стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE).

JRE - минимальная (без компилятора и других средств разработки) реализация виртуальной машины, необходимая для исполнения скомпилированного Java-приложения. Состоит из виртуальной машины (JVM) и библиотеку Java классов.

JVM - виртуальная машина, которая запускает скомпилированный Java байткод. JVM не распознает исходный код Java. Поэтому сначала компилятор компилирует файлы .java в файлы с байткодом .class.

Загрузчики классов - специальные классы, которые являются частью JVM. Они загружают классы в память и делают их доступными для выполнения. Класс будет загружен только тогда, когда понадобится приложению. Существует 3 загрузчика классов:

1. Bootstrap classloader - написан на С++. Содержит все основные библиотеки.
2. Extension classloader - загружает классы расширений при запуске из папки jre/lib/ext.
3. System classloader - загружает классы, которые определены в переменной среды окружения CLASSPATH.

JIT-компиляция – технология, при которой компиляция программы происходит во время ее выполнения, а также сохранение и переиспользование повторяющихся участков кода. Таким образом увеличивается производительность за счет увеличения потребления памяти и затрат времени на компиляцию.

Байткод – промежуточный код между кодом на языке Java и машинным кодом.

(показать как выглядит байткод в ИДЕА (ссылка) и машинный код (скачать картинку))

<https://habr.com/ru/post/568402/>

7 ООП:

Принципы SOLID

Что такое ООП

Основные принципы ООП

Основные понятия ООП

Преимущества ООП

Недостатки ООП

Что подразумевается под «является» и «имеет» в плане ООП

8 Класс Object:

Перечисли классы, которые не наследуются от класса Object

Почему все классы наследуются от класса Object

Назовите методы класса Object

Можно ли создать экземпляр класса Object

Зачем кому-то может понадобится создавать экземпляр класса Object

Что делает метод finalize()

Что будет, если finalize() подвиснет или выбросит исключение

Чем отличаются final, finally, finalize()

<https://www.youtube.com/watch?v=gmS0aaqgOXQ>

9 Equals() и hashCode():

Отличие equals() от ==

Условия для переопределения equals()

Правила переопределения equals()

Для чего нужен метод hashCode()

Как реализованы hashCode() и equals() в классе Object

Что будет, если переопределить equals, но не переопределить hashCode?

Как пользоваться интерфейсом Comparable?

Как пользоваться интерфейсом Comparator?

Для чего предназначен интерфейс Optional

10 Типы данны:

Перечислите типы данных в Java

В чем разница между ссылочными и примитивными типами данных

В чем разница между стеком (stack) и кучей (heap)

Типы ссылок в Java

Что такое приведение типов

Что такое автоупаковка / автораспаковка

Примитивные типы данных: значения по умолчанию и диапазон значений

В чем отличие i++ и ++i

Чем отличается использование && и & для типа boolean?

Что такое литералы

13 Метод и переменные:

Что такое local variable и instance variable

Что такое сигнатура метода

Что такое перегрузка метода

Как передаются переменные в метод, по ссылке или по значению

Какие бывают модификаторы доступа

Как получить доступ к private члену класса

Что такое рефлексия

Может ли статический метод быть переопределен или перегружен

Могут ли нестатические методы перегрузить статические

О чем говорят ключевые слова “this”, “super”

Как получить доступ к переопределенному методу родительского класса

Можно ли объявить метод абстрактным и статическим

В чем разница между членом экземпляра класса и статическим членом класса

5 Инициализация данных:

Разница между ранним (статическим) и поздним (динамическим) связыванием

Порядок вызова конструктора и блоков инициализации

Для чего используются статические блоки инициализации

Если в блоке инициализации ошибка, что будет

Где разрешена инициализация статических / нестатических полей

8 Интерфейс:

Что такое интерфейс

Что такое default-метод интерфейса

Как вызвать default-метод интерфейса в реализующем классе

Что такое static метод интерфейса

Как вызвать static метод интерфейса

Как разрешить проблему с двумя одинаковыми методами в интерфейсе

Почему в некоторых интерфейсах не определены методы

Почему нельзя объявить метод интерфейса final

12 Абстрактный класс:

Что такое абстрактные методы и классы

Отличие абстрактного класса от интерфейса

Может ли у абстрактного класса быть конструктор

Могут ли абстрактные классы реализовывать интерфейсы. Должны ли они реализовывать все методы

Может ли абстрактный класс быть final

Может ли у абстрактного класса быть статический метод

Можно ли создать экземпляр абстрактного класса

Обязательно ли в абстрактном классе должны быть абстрактные методы

Когда имеет смысл предпочесть абстрактный класс интерфейсу и наоборот

Что такое абстрактный метод в языке Java

Может ли абстрактный класс в Java содержать метод main

Иерархия по уровню абстракции

<https://javarush.ru/groups/posts/431-10-voprosov-po-abstraktnihm-klassam-i-interfeysam-s-sobesedovaniy-po-jazihku-java>

8 Enum:

Может ли Enum реализовывать (implement) интерфейс в Java

Может ли Enum наследовать (extends) класс?

Как создать Enum без экземпляров объектов? Возможно ли это без ошибки компиляции?

Можем ли мы переопределить метод toString() для Enum? Что будет, если не будем переопределять?

Можем ли мы указать конструктор внутри Enum?

Какая разница сравнивать Enum при помощи == или метода equals()?

Можно использовать Enum с TreeSet или TreeMap в Java?

Какая выгода использовать Enum вместо паттернов перечисления int и перечисления String?

9 Типы классов:

Типы классов

Что такое вложенный класс

Что такое локальные класс

Что такое анонимный класс

Во что компилируется анонимный внутренний класс

Зачем использовать ключевое слово final при создании анонимных классов

Как внутренние классы решают проблему множественного наследования в Java

Чем отличаются анонимные классы, созданные на основе интерфейса и на основе класса

Можно ли создать анонимный статический вложенный класс

11 Класс String:

Что такое immutable объекты

Как создать immutable класс

Какая разница между String, StringBuilder, StringBuffer, StringJoiner

// конкатенация в Java — синтаксический сахар над созданием StringBuilder, несколькими вызовами append и затем toString.

Что такое пул стрингов (строк)

Как отсортировать список строк в алфавитном порядке?

В какой кодировке хранятся строки в Java?

Почему char[] лучше для хранение пароля чем String

Почему String хороший ключ для HashMap

Что делает метод intern() в классе String

Можно ли использовать строки в switch

Почему String immutable и final

13 Клонирование и сериализация:

Расскажите про клонирование объектов

Какой способ клонирования предпочтительней

Что такое сериализация

Как сделать Java-класс сериализуемым?

В чем разница между интерфейсом Serializable и Externalizable в Java?

Что такое serialVersionUID? Что произойдет, если вы его не определите?

При сериализации вы хотите, чтобы некоторые члены не сериализовались? Как этого добиться?

Что произойдет, если один из членов класса не реализует интерфейс Serializable?

Если класс является сериализуемым, а его суперкласс - нет, каково будет состояние переменных экземпляра, унаследованных от суперкласса после десериализации?

Можете ли вы настроить процесс сериализации или переопределить процесс сериализации по умолчанию в Java?

Предположим, суперкласс нового класса реализует интерфейс Serializable, как можно избежать сериализации нового класса?

Какие методы используются в процессе сериализации и десериализации в Java?

Предположим, у вас есть класс, который вы сериализовали и сохранили, а затем изменили этот класс, чтобы добавить новое поле. Что произойдет, если вы десериализуете уже сериализованный объект?

<https://javarush.ru/groups/posts/3098-top-13-voprosov-pro-serializaciju-na-sobesedovanijakh>

6 Generics:

Что такое дженерики

Какие преимущества дает использование дженериков

Что такое стирание типов

Расскажите про extends и super в Generic’ах

Что такое wildcard и как его использовать

Можете ли вы передать List <String> методу, который принимает List <Object>

// добавить еще вопросов, заданий (взять по аналогии с JavaRush)

<https://coderlessons.com/articles/java/java-generics-interviu-voprosy>

Java 8

5 Лямбда-выражения и ссылка на метод:

Что такое лямбда-выражения

К каким переменным есть доступ у лямбда выражений

Отсортировать список с помощью лямбда выражений (блок практич. заданий)

Что такое «ссылка на метод» (method reference)

Виды ссылок на метод

<https://javarush.ru/groups/posts/lambda-vihrazhenija-v-java-chast-1>

2 Функциональный интерфейс:

Что такое функциональный интерфейс

Распространенные функциональные интерфейсы

9 Stream:

Что такое Stream

Способы создания Stream

Промежуточные операции

Финальные операции

В чем разница между Collection и Stream

Расскажите о параллельной обработке

Критерии, которые могут влиять на производительность в параллельных Stream

Промежуточные методы работы со Sream

Конечные методы работы со Stream

// практические задания по stream

<https://annimon.com/article/2778>

Исключения

4 Описание и иерархия исключений:

Что такое исключение

Иерархия исключений

Что вы знаете об обрабатываемых и не обрабатываемых (checked/unchecked) исключениях?

В чем особенность RuntimeException?

15 Обработка исключений:

Можно/нужно ли обрабатывать ошибки jvm?

Какие существуют способы обработки исключений?

Принудительное выбрасывание исключений

Что за ключевое слово throw

В чем особенность блока finally? Всегда ли он исполняется?

Может ли не быть ни одного блока catch при отлавливании исключений?

Могли бы вы придумать ситуацию, когда блок finally не будет выполнен?

Может ли один блок catch отлавливать несколько исключений (с одной и разных веток наследований)?

Есть ли дополнительные условия к методу, который потенциально может выбросить исключение?

Try-catch-finally

Try-with-resources

Как лучше всего освобождать ресурсы

Как написать собственное исключение

Может ли main() выбросить исключение, где оно будет обрабатываться?

Best practices обработки исключений

3 Нюансы по исключениям

Предположим, есть метод, который может выбросить IOException и FileNotFoundException в какой последовательности должны идти блоки catch? Сколько блоков catch будет выполнено?

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?

Если оператор return содержится и в блоке catch и в finally, какой из них “главнее”?

<https://javastudy.ru/interview/exceptions/>

**Java Collection**

12 Java Collection Framework:

Расскажите про Java Collection Framework

Иерархия Collection

В чем разница между java.util.Collection и java.util. Collections

Fail-fast и fail-safe поведение

В чем отличие Enumeration и Iteration

Связь между Iterable и Iterator

Что такое ListIterator

Дисциплины облуживания FIFO и FILO

Что такое sorted и ordered в коллекциях

Какие методы есть у класса Collections?

Какие методы есть у класса Arrays?

Collections.sort() и Arrays.sort()

4 ArrayList:

Что такое ArrayList

Какая временная сложность операций у ArrayList

Почему в ArrayList доступ к объектам О(1)

Как происходит удаление элементов из ArrayList. Как меняется в этом случае размер ArrayList

2 LinkedList:

Что такое LinkedList

Какая временная сложность операций у LinkedList

9 HashMap:

Что такое HashMap

Какая временная сложность операций у HashMap

Может ли HashMap выродится в список даже с разными hashCode()

Как можно потерять элемент в HashMap

Почему нельзя использовать byte[] в качестве ключа в HashMap

Какое максимальное число значений hashCode()

Время работы get(key) в HashMap

Сколько создается объектов при добавлении в HashMap

Перебор HashMap

1 LinkedHashMap:

Что такое LinkedHashMap

2 TreeMap:

Что такое TreeMap

Что такое красно-черное дерево

1 HashSet:

Что такое HashSet

1 LinkedHashSet:

Что такое LinkedHashSet

2 TreeSet:

Что такое TreeSet

Что будет, если добавлять элементы в TreeSet по возрастанию (возможн переместить вопрос на после красно-черного дерева)

2 Queue:

Что такое Queue, Deque, PriorityQueue, ArrayDeque

Что дает LinkedList’у наследование от Deque

5 Сравнение коллекций:

Отличие ArrayList от Vector

Сравните вставку в середину в ArrayList или LinkedList

Что лучше ArrayList или LinkedList

В чем отличия TreeSet и HashSet

Необходимо добавить 1млн. элемент, какую структуру вы используете?

<https://habr.com/ru/post/162017/>

Операции с коллекциями (сделать практические задания, самим найти ответ)

Как быстро сделать копирование System.arraycopy() (практич зад)

**Потоки ввода / вывода**

11 Потоки ввода / вывода

Разница между NIO / IO

Что такое канал

Основные классы потоков ввода / вывода

В каких пакетах расположены классы потоков ввода / вывода (практич. зад.)

Можно ли перенаправить потоки стандартного ввода / вывода

Что такое «абсолютный путь» и «относительный путь» (практич. зад.)

Что такое символьная ссылка

Какой класс предназначен для работы с файловой системой

Что вы знаете об интерфейсе FileFilter

Что такое DynamicProxy

Что такое Wrapper Classes (Классы обертки)

4 InputStream / OutputStream

Класс InputStream и его подклассы

Класс OutputStream и его подклассы

Разница между BufferedOutputStream и ByteArrayOutputStream

Что вы знаете о классе RandomAccessFile

3 Reader / Writer

Класс Reader и его подклассы

Класс Writer и его подклассы

Разница между PrintWrite и PrintStream

**Многопоточность**

12 Основные понятия:

Что такое модель памяти Java (JMM)

Что такое потокобезопасность

В чем разница между конкуренцией и параллелизмом

Какой тип многопоточности использует Java

Что такое процесс

Что такое поток

Разница между потоком и процессом

Что такое daemon thread

Можно ли сделать основной поток программы daemon

Что такое sequential consistency (as-if-serial semantics)

Что такое happens-before

Beast practices при разработке многопоточных программ

4 Создание многопоточности:

Как создать новый поток

Сколько способов создать нить вы знаете (Thread, Runnable, Callable<T>)

Разница между Thread и Runnable

Чем отличаются интерфейсы Runnable и Callable

18 Управление многопоточностью:

Разница между start() и run()

Что будет, если метод start() вызвать дважды

Как принудительно запустить поток

Состояния в жизненном цикле потока

Можно ли запустить поток дважды

Как работают методы wait(), notify(), notifyAll()

В чем разница между notify() и notifyAll()

Почему wait() и notify() вызываются только в synchronized блоке

Отличие работы wait() с параметрами и без параметров

Чем отличаются методы wait(1000) и sleep(1000)

Что такое Thread.yield() и Thread.sleep()

Как вывести поток из режима сна (sleep()) раньше времени

Снимется ли блокировка, если вызвать метод sleep() во время выполнения синхронизированного кода

Что такое Thread.join()

Как остановить поток

Почему не рекомендуется использовать Thread.stop()

Разница между interrupted() и isInterrupted()

Что такое busy spin

10 Синхронизация:

Что такое синхронизация

Что такое блокирующий метод

Что такое монитор, семафор, мьютекс

Зачем может быть нужен private мьютекс

Как проверить, удерживает ли поток монитор определенного ресурса

Чем полезны неизменяемые объекты с точки зрения многопоточности

Ключевые слова transient и native

Ключевые слова volatile и synchronized

Можно ли создавать новые экземпляры класса пока выполняется static synchronized method

На каком объекте будет синхронизация при вызове static synchronized method

6 Ошибки при использовании многопоточности:

Что происходит, когда в потоке выбрасывается исключение

Что такое shutdownhook

Что такое состояние гонки (race condition)

Способы решения состояния гонки

Что такое deadlock

Что такое livelock

13 ThreadPool:

Что такое пул потоков (ThreadPool)

Что такое приоритет потока

Зачем нужен класс ThreadGroup

Какого размера должен быть пул потоков

Что будет, если очередь пула потоков заполнена, но подается новая задача

Что такое интерфейс Future

Что такое FutureTask

Что такое интерфейс Executor

Зачем нужен класс ThreadPoolExecutor

Разница между submit() и execute() у пула потоков

В чем разница между CyclicBarrier и CountDownLatch

Как поделится данными между двумя потоками

Что такое ThreadLocal переменная

8 Библиотека Concurrent:

Concurrent сollection

Назовите различия между Collections.synchronizedMap(new HashMap()) и ConcurrentHashMap

В чем разница между volatile и atomic переменными

Что такое класс Lock

Что такое ReentrantLock

Разница между synchronized и ReentrantLock

Что такое ReadWriteLock

Что такое StampedLock

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/concurrent/package-summary.html>

**Практические задачи**

Написать сортировки

Найти задачки

**Собеседование (аудио)**

Рекомендации по прохождению аудиособеседований (представить, что это реальное собеседование, отвечать в слух)

10 аудиособеседований

**Резюме**

Правила создания резюме

<https://habr.com/ru/post/162017/>

Кстати, с ArrayList ещё интересна постановка вопроса о вставке в середину или удалении из середины. Да, с математической точки зрения это O(N-k), где k — позиция вставки/удаления. Но в действительности это делается за одну операцию System.arraycopy, которая реализована нативно и сводится к ассемблерным инструкциям копирования блока памяти, которые даже для незакэшированного массива отработают довольно быстро. У меня (CoreDuo E7300 @2.66GHz) вставка в начало ArrayList из миллиона элементов занимает около одной миллисекунды, то есть по сути дела сдвиг одного элемента в массиве занимает наносекунду. Поэтому здесь линейная сложность не так уж страшна для большинства задач.

Вступление.

Здравствуйте дорогие студенты моего курса. Меня зовут Виталий. Я работаю программистом c 2018 года. Этот курс предназначен подготовки Вас к предстоящему собеседованию. Также этот курс будет полезен для систематизации и дополнения уже имеющихся у Вас знаний.

Я прекрасно помню свои первые собеседования на позицию Junior Java Developer. Это было ужасно. От волнения я забывал ответы на вопросы, которые прекрасно знал. Также собеседования показали, что мои знания в языке Java весьма поверхностны. Я не понимал многих вещей, которые должен знать программист на Java. Например, в каком порядке инициализируются данные, как устроены коллекции в Java Framework Collection, различные нюансы при работе с многопоточностью.

После осознания скудности моих теоретических знаний (при том что на тот момент у меня в портфолио было полноценное приложение на Spring с авторизацией и правами доступа на основе ролей), я конечно же начал искать вопросы в интернете, читать книги. Откровенно говоря, это не самый веселый процесс.

Поэтому я и создал этот курс, чтобы облегчить Вам поиск информации. Теперь Вам не нужно тратить кучу времени на поиск вопросов, задач, а также внятных ответов на них. В конце курса Вам предстоит самостоятельно пройти несколько аудио собеседований, которые (при должном участии с Вашей стороны) должны помочь Вам справится с волнением и придать уверенности в своих силах.

Успехов Вам!

И помните: «Дорогу осилит идущий». Не останавливайтесь на пути к своей цели.

Меня зовут Виталий. Я работаю программистом c 2018 года. Этот курс предназначен подготовки Вас к предстоящему собеседованию. Также этот курс будет полезен для систематизации и дополнения уже имеющихся у Вас знаний.

Я прекрасно помню свои первые собеседования на позицию Junior Java Developer. Это было ужасно. От волнения я забывал ответы на вопросы, которые прекрасно знал. Также собеседования показали, что мои знания в языке Java весьма поверхностны. Я не понимал многих вещей, которые должен знать программист на Java (при том что на тот момент у меня в портфолио было полноценное приложение на Spring с авторизацией и правами доступа на основе ролей). Поэтому я и создал курс для подготовки к собеседованию на позицию Java Developer.

Почему Вам нужно пройти этот курс:

без воды, все четко и по делу

конспект в конце каждого модуля для быстрого поворения изученного материала