

# Java Basics Overview Lesson 08

OLGA SMOLYAKOVA

## План конспекта

1. Пройденные темы

2. Перечень вопросов по Java Fundamentals

**1.1 Java Basics. Java Fundamentals.**

1. Установка JDK.
2. Настройка переменных среды окружения (ОС Windows).
3. Пишем первую программу.
4. Целочисленное деление или типы с плавающей точкой спешат на помощь.
5. Литералы.
6. Класс Math.

**1.2 Java Basics. Java Fundamentals.**

1. Литералы.
2. Блок-схемы.
3. Приведение примитивных типов.
4. Приоритет операции.
5. Оператор if.
6. Составные логические выражения.
7. Практическая работа.

**1.3 Java Basics. Java Fundamentals.**

1. Чтение из консоли.
2. Циклы.
3. Оператор switch.

**1.4 Java Basics. Data Structures. Arrays.**

1. Что такое структуры данных.
2. Одномерные массивы примитивных типов данных.
3. Сортировки.
4. Массивы массивов.

**1.5 Java Basics. Procedural Decomposition.**

1. Декомпозиция кода с помощью статических методов.
2. Основы работы с типом String.
3. Области данных времени выполнения (виды памяти в Java).

**1.6 Java Basics. Classes and Objects.**

1. Введение классов и объектов как простейших структур данных.
2. Использование классов и объектов как элементов ООП.
3. Простейшее решение задач с помощью классов и объектов.

## 1.7 Java Basics. Classes and Objects. Tasks.

1. Решение простейших задач на проектирование классов.
2. Подход к решению ООП задач, основанный на отделении логики от сущности.

## 2

## Перечень вопросов по Java Fundamentals

### 2.1 Java Fundamentals (Core)

- Q Объясните, что имеется в виду, когда говорится: Java-язык программирования и Java-платформа.
- Q Поясните, как связаны имя java-файла и классы, которые в этом файле объявляются.
- Q Расшифруйте аббревиатуры JVM, JDK и JRE; покажите, где “они находятся” и что собой представляют.
- Q Объясните, как скомпилировать и запустить приложение из командной строки, а также зачем в переменных среды окружения прописывать пути к установленному jdk.
- Q Перечислите атрибуты доступа, объясните их действие.
- Q Что такое пакеты в java-программе, что представляют собой пакеты на диске? Каково соглашение по именованию пакетов? Как создать пакет?
- Q Объясните, какие классы, интерфейсы, перечисления необходимо импортировать в вашу программу, как это сделать. Влияет ли импорт пакета на импорт классов и др., лежащего в подпакетах? Какой пакет в Java импортируется по умолчанию?
- Q Объясните различия между терминами “объект” и “ссылка на объект”.
- Q Какие примитивные типы Java вы знаете, как создать переменные примитивных типов? Объясните процедуру, по которой переменные примитивных типов передаются в методы как параметры.
- Q Каков размер примитивных типов, как размер примитивных типов зависит от разрядности платформы, что такое преобразование (приведение) типов и зачем оно необходимо? Какие примитивные типы не приводятся ни к какому другому типу.
- Q Объясните, что такое явное и неявное приведение типов, приведите примеры, когда такое преобразование имеет место.
- Q Что такое литералы в Java-программе, какую классификацию литералов вы знаете, как записываются литералы различных видов и типов в Java-программе?

- Q Как осуществляется работа с типами при вычислении арифметически выражений в Java?
- Q Что такое классы-оболочки, для чего они предназначены? Объясните, что значит: объект класса оболочки – константный объект.
- Q Объясните разницу между примитивными и ссылочными типами данных. Поясните существующие различия, при передаче параметров примитивных и ссылочных типов в методы. Объясните, как константные объекты ведут себя при передаче в метод.
- Q Поясните, что такое автоупаковка и автораспаковка.
- Q Перечислите известные вам арифметические, логические и битовые операторы, определите случаи их употребления. Что такое приоритет оператора, как определить, в какой последовательности будут выполняться операции в выражении, если несколько из них имеют одинаковый приоритет.
- Q Укажите правила выполнения операций с плавающей точкой в Java (согласно стандарту IEEE754). Как определить, что результатом вычисления стала бесконечность или нечисло?
- Q Что такое статический импорт, какие элементы можно импортировать при статическом импорте.
- Q Объясните работу операторов if, switch, while, do-while, for, for-each. Напишите корректные примеры работы этих операторов.
- Q Объясните работу оператора instanceof. Что будет результатом работы оператора, если слева от него будет стоять ссылка, равная null?
- Q Дайте определение массиву. Как осуществляется индексация элементов массива. Как необходимо обращаться к i-му элементу массива?
- Q Приведите способы объявления и инициализации одномерных и двумерных массивов примитивных и ссылочных типов. Укажите разницу, между массивами примитивных и ссылочных типов.
- Q Объясните, что представляет собой двумерный массив в Java, что такое “рваный массив”. Как узнать количество строк и количество элементов в каждой строке для “рваного” массива?
- Q Объясните ситуации, когда в java-коде могут возникнуть следующие исключительные ситуации `java.lang.ArrayIndexOutOfBoundsException` и `java.lang.ArrayStoreException`.
- Q Дайте определение таким понятиям как “класс” и “объект”. Приведите примеры объявления класса и создания объекта класса. Какие спецификаторы можно использовать при объявлении класса.

- Q Как вы определяете, какие поля и методы необходимо определить в классе, приведите пример. Какие спецификаторы можно использовать с полями, а какие с методами (и что они значат)?
- Q Что такое конструктор? Как вы отличите конструктор от любого другого метода? Сколько конструкторов может быть в классе? Что такое конструктор по умолчанию, может ли в классе совсем не быть конструкторов? Объясните, что делает оператор `this()` в конструкторе?
- Q Приведите правила, которым должен следовать компонент `java-bean`.

## 2.2 OOP, Classes and Objects, Inheritance, Class Object, Interfaces

- Q Дайте развернутое объяснение трем концепциям ООП.
- Q Опишите процедуру инициализации полей класса и полей экземпляра класса. Когда инициализируются поля класса, а когда – поля экземпляров класса. Какие значения присваиваются полям по умолчанию? Где еще в классе полям могут быть присвоены начальные значения?
- Q Приведите правила, которым должен следовать компонент `java-bean`.
- Q Дайте определение перегрузке методов. Как вы думаете, чем удобна перегрузка методов? Укажите, какие методы могут перегружаться, и какими методами они могут быть перегружены? Можно ли перегрузить методы в базовом и производном классах? Можно ли `private` метод базового класса перегрузить `public` методов производного? Можно ли перегрузить конструкторы, и можно ли при перегрузке конструкторов менять атрибуты доступа у конструкторов?
- Q Объясните, что такое раннее и позднее связывание? Перегрузка – это раннее или позднее связывание? Объясните правила, которым следует компилятор при разрешении перегрузки; в том числе, если методы перегружаются примитивными типами, между которыми возможно неявное приведение или ссылочными типами, состоящими в иерархической связи.
- Q Объясните, как вы понимаете, что такое неявная ссылка `this`? В каких методах эта ссылка присутствует, а в каких – нет, и почему?
- Q Что такое финальные поля, какие поля можно объявить со спецификатором `final`? Где можно инициализировать финальные поля?
- Q Что такое статические поля, статические финальные поля и статические методы. К чему имеют доступ статические методы? Можно ли перегрузить и переопределить статические методы? Наследуются ли статические методы?

- Q Что такое логические и статические блоки инициализации? Сколько их может быть в классе, в каком порядке они могут быть размещены и в каком порядке вызываются?
- Q Что представляют собой методы с переменным числом параметров, как передаются параметры в такие методы и что представляет собой такой параметр в методе? Как осуществляется выбор подходящего метода, при использовании перегрузки для методов с переменным числом параметров?
- Q Чем является класс Object? Перечислите известные вам методы класса Object, укажите их назначение.
- Q Что такое хэш-значение? Объясните, почему два разных объекта могут сгенерировать одинаковые хэш-коды?
- Q Как вы думаете, для чего используется наследование классов в java-программе? Приведите пример наследования. Как вы думаете, поля и методы, помеченными модификатором доступа private, наследуются?
- Q Укажите, как вызываются конструкторы при создании объекта производного класса? Что в конструкторе класса делает оператор super()? Возможно ли в одном конструкторе использовать операторы super() и this()?
- Q Объясните, как вы понимаете утверждения: “ссылка базового класса может ссылаться на объекты своих производных типов” и “объект производного класса может быть использован везде, где ожидается объект его базового типа”. Верно ли обратное и почему?
- Q Что такое переопределение методов? Как вы думаете, зачем они нужны? Можно ли менять возвращаемый тип при переопределении методов? Можно ли менять атрибуты доступа при переопределении методов? Можно ли переопределить методы в рамках одного класса?
- Q Определите правило вызова переопределенных методов. Можно ли статические методы переопределить нестатическими и наоборот?
- Q Какие свойства имеют финальные методы и финальные классы? Как вы думаете, зачем их использовать?
- Q Укажите правила приведения типов при наследовании. Напишите примеры явного и неявного преобразования ссылочных типов. Объясните, какие ошибки могут возникать при явном преобразовании ссылочных типов.
- Q Что такое объект класса Class? Чем использование метода getClass() и последующего сравнения возвращенного значения с Type.class отличается от использования оператора instanceof?

- Q Укажите правила переопределения методов equals(), hashCode() и toString().
- Q Что такое абстрактные классы и методы? Зачем они нужны? Бывают ли случаи, когда абстрактные методы содержат тело? Можно ли в абстрактных классах определять конструкторы? Могут ли абстрактные классы содержать неабстрактные методы? Можно ли от абстрактных классов создавать объекты и почему?
- Q Что такое интерфейсы? Как определить и реализовать интерфейс в java-программе? Укажите спецификаторы, которые приобретают методы и поля, определенные в интерфейсе. Можно ли описывать в интерфейсе конструкторы и создавать объекты? Можно ли создавать интерфейсные ссылки и если да, то на какие объекты они могут ссылаться?
- Q Для чего служит интерфейс Cloneable? Как правильно переопределить метод clone() класса Object, для того, что объект мог создавать свои адекватные копии?
- Q Для чего служат интерфейсы Comparable и Comparator? В каких случаях предпочтительнее использовать первый, а когда – второй? Как их реализовать и использовать?

### 2.3 Generic classes and Interfaces, Enums, Classes into classes.

- Q Что такое перечисления в Java. Как объявить перечисление? Чем являются элементы перечислений? Кто и когда создает экземпляры перечислений? Могут ли перечисления реализовывать интерфейсы или содержать абстрактные методы? Могут ли перечисления содержать статические методы?
- Q Можно ли самостоятельно создать экземпляр перечисления? А ссылку типа перечисления? Как сравнить, что в двух переменных содержится один и тот же элемент перечисления и почему именно так?
- Q Что такое внутренние, вложенные и анонимные классы? Как определить классы такого вида? Как создать объекты классов такого вида.
- Q Перечислите возможности доступа к членам внешнего класса, которым наделены вложенные, внутренние и анонимные классы? Могут ли классы внутри классом быть базовыми, производными или реализующими интерфейсы? Можно ли анонимный класс создать от интерфейса?
- Q Что такое параметризованные классы? Для чего они необходимы? Приведите пример параметризованного класса и пример создания объекта параметризованного класса? Объясните, ссылки какого типа



могут ссылаться на объекты параметризованных классов? Можно ли создать объект, параметризовав его примитивным типом данных?

- Q Какие ограничения на вызов методов существуют у параметризованных полей? Как эти ограничения снимает использование при параметризации ключевого слова `extends`?
- Q Как параметризуются статические методы, как определяется конкретный тип параметризованного метода? Можно ли методы экземпляра класса параметризовать отдельно от параметра класса, и если “да”, то как тогда определять тип параметра?
- Q Что такое wildcard? Приведите пример его использования?
- Q Для чего используется параметризация `<? extends Type>`, `<? super Type>`?

## 2.4 Strings

- Q Как создать объект класса `String`, какие конструкторы класса `String` вы знаете? Что такое строковый литерал? Объясните, что значит “упрощенное создание объекта `String`”?
- Q Можно ли изменить состояние объекта типа `String`? Что происходит при попытке изменения состояния объекта типа `String`? Можно ли наследоваться от класса `String`? Как вы думаете, почему строковые объекты `immutable`?
- Q Объясните, что такое кодировка? Какие кодировки вы знаете? Как создать строки в различной кодировке?
- Q Что такое пул литералов? Как строки заносятся в пул литералов? Как занести строку в пул литералов и как получить ссылку на строку, хранящуюся в пуле литералов? Где хранится (в каком типе памяти) пул литералов в Java 1.6 и Java 1.7?
- Q В чем отличие объектов классов `StringBuilder` и `StringBuffer` от объектов класса `String`? Какой из этих классов потокобезопасный? Как необходимо сравнивать на равенство объекты классов `StringBuilder` и `StringBuffer` и почему?
- Q Что такое Unicode? Что такое code point?
- Q Как кодируется символ согласно кодировке `utf-8`, `utf-16` и `utf-32`?
- Q Какие методы класса `String` используются для работы с кодовыми точками? Как вы думаете, когда следует их использовать?

## 2.5 Regular Expressions

- Q Расскажите, что представляет собой регулярное выражение? Что такое метасимволы регулярного выражения? Какие вы знаете классы символов регулярных выражений? Что такое квантификаторы? Какие логические операторы регулярных выражений вы знаете? Что значит “якорь” для регулярного выражения?
- Q Какие java-классы работают с регулярными выражениями? В каком пакете они расположены? Приведите пример анализа текста с помощью регулярного выражения и поясните код примера.
- Q Что такое группы в регулярных выражениях? Как нумеруются группы? Что представляет собой группа номер 0(ноль)? Приведите пример с использованием групп регулярного выражения.

## 2.6 IO Streams

- Q Что такое поток данных? Какие потоки данных существуют в Java? Приведите иерархию потоков ввода-вывода в Java.
- Q Какие классы байтовых потоков ввода-вывода вам известны? Какие классы символьных потоков ввода-вывода вам известны?
- Q Прокомментируйте поведение методов read() и write() базовых классов иерархии символьных и байтовых потоков. Сравните их.
- Q Для чего используются классы BufferedInputStream, BufferedOutputStream, BufferedReader, BufferedWriter?
- Q Для чего используются классы FilterInputStream, FilterOutputStream, FilterReader, FilterWriter?
- Q Для чего применяются классы InputStreamReader и OutputStreamWriter?
- Q Что такое упаковка (wrapping) потоков?
- Q Какие предопределенные потоки ввода-вывода в Java вы знаете? Кто эти потоки создает и, кто их закрывает?
- Q Что такое сериализация и десериализация? Приведите правила сериализации и десериализации объектов? Будет ли повторно сериализоваться уже сериализованный объект? Что получится при десериализации, если при сериализации создаваемые объекты имели ссылки на одни и те же объекты? Что такое граф ссылок объекта? Как происходит десериализация, вызывается ли конструктор при десериализации? Как десериализуются объекты, созданные от классов, у которых базовые классы несериализуемые?
- Q Как сериализовать и десериализовать объект? Какие классы и интерфейсы для этого необходимо использовать? Какое статическое поле сериализуется?

Q Что происходит при сериализации/десериализации объекта-синглтона. Как правильно сериализовать синглтон?

## 2.7 Localization and Internationalization

- Q Дайте разъяснения таким понятиям как интернационализация и локализация.
- Q Что представляет (и что определяет) собой локаль в java-программе, назначение объектов класса Locale? Как получить локаль? Как узнать, какие локали доступны?
- Q Какую информацию можно локализовать автоматически, применяя объект класса Locale? Как работают классы NumberFormat и DateFormat? Что делает класс DateSimpleFormat?
- Q Как можно локализовать приложение, используя класс ResourceBundle? Для каких еще целей, кроме локализации, можно применять объекты этого класса?
- Q Как сериализовать и десериализовать объект? Какие классы и интерфейсы для этого необходимо использовать? Какое статическое поле сериализуется?

## 2.8 Exceptions and Errors

- Q Что для программы является исключительной ситуацией? Какие способы обработки ошибок в программах вы знаете?
- Q Что такое исключение для Java-программы? Что значит “программа выбросила исключение”? Опишите ситуации, когда исключения выбрасываются виртуальной машиной(автоматически), и когда необходимо их выбрасывать вручную?
- Q Приведите иерархию классов-исключений, делящую исключения на проверяемые и непроверяемые. В чем особенности проверяемых и непроверяемых исключений?
- Q Объясните работу оператора try-catch-finally. Когда данный оператор следует использовать? Сколько блоков catch может соответствовать одному блоку try? Можно ли вкладывать блоки try друг в друга, можно ли вложить блок try в catch или finally? Как происходит обработка исключений, выброшенных внутренним блоком try, если среди его блоков catch нет подходящего? Что называют стеком операторов try? Как работает блок try с ресурсами.

- Q Укажите правило расположения блоков catch в зависимости от типов перехватываемых исключений. Может ли перехваченное исключение быть сгенерировано снова, и, если да, то как и кто в этом случае будет обрабатывать повторно сгенерированное исключение? Может ли блок catch выбрасывать иные исключения, и если да, то опишите ситуацию, когда это может быть необходимо.
- Q Когда происходит вызов блока finally? Существуют ли ситуации, когда блок finally не будет вызван? Может ли блок finally выбрасывать исключений? Может ли блок finally выполниться дважды?
- Q Как выбросить исключение вручную? Какие объекты (объекты каких классов) могут быть выброшены в качестве исключений? Можно ли выбросить две исключения одновременно? Объясните, как работают операторы throw и throws, в чем их отличия? Объясните правила реализации секции throws при переопределении метода и при описании конструкторов производного класса. Опишите поведения блока throws при работе с проверяемыми и непроверяемыми исключениями. Каков будет результат создания объекта, если конструктор при работе выбросит исключительную ситуацию?
- Q Можно ли (и нужно ли) в Java выбрасывать исключения, входящие в пакет JDK? Как в Java создать собственные классы исключений?
- Q Перечислите известные вам методы класса Throwable и их назначение?

## 2.9 Networks Fundamental and Socket Fundamentals

- Q Опишите семь уровней модели ISO/OSI. Для чего была разработана эта модель? Опишите назначения каждого из уровней и основные протоколы, работающие на этих уровнях.
- Q Что такое MAC-адрес? Кто назначает MAC-адреса? Опишите известные вам топологии сетей и методы передачи данных в этих сетях (в общем виде, не углубляясь в детали).
- Q Расскажите о принципах адресации на сетевом уровне? Что такое IP-адрес, какие классы IP-адресов вы знаете? Что такое маска подсети и для чего она служит? Какие зарезервированные IP-адреса вы знаете и для чего они служат? Как вы думаете, почему для адресации в сети недостаточно MAC-адреса сетевого устройства? Для чего предназначены ARP и RARP протоколы? Как происходит выбор маршрута для передачи данных?
- Q В чем отличие протокола TCP от UDP. Как вы думаете, зачем два протокола (TCP и IP), работающие на различных уровнях, объединяют в один стек

TCP/IP? Каково назначение протоколов DHCP? Что такое и зачем нужен DNS? Что представляет собой механизм NAT и в чем его преимущество?

- Q Что такое хост и порт? Дайте определение сокету. Объясните, как происходит взаимодействие двух объектов типа Socket и ServerSocket. Сокеты в Java взаимодействуют по протоколу TCP или UDP? Для чего предназначен класс DatagramSocket? Объясните, как происходит взаимодействие между объектами DatagramSocket?

## 2.10 Multithreading and JMM Fundamentals

- Q Дайте определение процессу и потоку. В чем отличие потоков от процессов?
- Q Укажите два способа создания потокового объекта в Java. Как запустить поток?
- Q Что такое “главный поток”, как получить ссылку на главный поток выполнения программы?
- Q Опишите жизненный цикл потока. Сравните работу методов `isAlive()` и `getState()` класса `Thread`.
- Q Что делает метод `join()` класса `Thread`?
- Q Как прервать поток. У чем отличие методов `interrupt()`, `interrupted()` и `isInterrupted()`
- Q Объясните, зачем нужна синхронизация и что такое монитор(мьютекс). Как вы понимаете, что такое критическая секция?
- Q Объясните, как работают синхронизированные методы и синхронизированные блоки. Чем использование синхронизированных методов отличается от применения синхронизированных блоков. Может ли код “отказаться” от вызова синхронизированного метода или входа в синхронизированную секцию, если блокировка не отдается долгое время.
- Q Объясните, что значит, поток, взявший монитор, может взять его повторно?
- Q Зачем нужны потоки-демоны? Как создать поток-демон? Когда следует применять потоки-демоны?
- Q Объясните назначение методов `wait()`, `notify()` и `notifyAll()`. Каков будет результат, если на ресурсе вызвать метод `notify()` до этого не вызвав соответствующий ему `wait()`.
- Q Что такое группа потоков? Зачем потоки объединять в группы? Как создать группу потоков? Как добавить потоки в группу потоков? Когда группа потоков уничтожается? Что такое группа-демон?
- Q На что влияет приоритет потока? Как потоку установить приоритет?

- Q Приведите пример (примеры), как в Java можно обработать непроверяемые исключения, выброшенные потоком.
- Q Объясните, что такое взаимная блокировка (deadlock), приведите примеры, когда потоки могут попасть в ситуацию взаимной блокировки. Предложите решения, позволяющие потокам не попадать в ситуацию взаимной блокировки.
- Q Объясните основные положения старой и новой модели памяти в Java? Как вы полагаете, почему появилась необходимость замены представления о модели памяти? Как вы понимаете, что такое кэш потока? Поясните основные свойства новой модели памяти: atomicity, visibility, ordering.
- Q Что такое атомарная операция? Какие операции в Java являются атомарными? Перед какими полями имеет смысл использовать модификатор volatile? Как изменится поведение программы, если перед полями использовать модификатор volatile?
- Q Как в Java можно осуществить приостановку/возобновление работы потоков?
- Q Объясните, зачем в Java была введена библиотека concurrent (java.util.concurrent)? Что она содержит?
- Q Приведите известные вам синхронизированные коллекции и их назначение из пакета java.util.concurrent.
- Q Приведите известные вам синхронизаторы(synchronizers) и блокираторы(locks) и их назначение из пакета java.util.concurrent.
- Q В чем состоит назначение классов Atomic из пакета java.util.concurrent.
- Q Что такое Executors в пакете java.util.concurrent? Зачем они нужны? Что представляет собой пул потоков? Попробуйте привести пример самостоятельно написанного пула потоков.

## 2.11 Garbage Collector

- Q Назовите JVM Run-Time Memory Areas. Как вы думаете, с какой областью памяти работает GC? Перечислите основные действия, предпринимаемые JVM, при выполнении приложения? Какие объекты считаются мусором? Какие методики определения (подсчета) мусора вы знаете?
- Q Опишите стратегии (алгоритмы) работы GC: Sweeping GC, Compacting GC, Copying GC. Укажите их достоинства и недостатки.
- Q Опишите стратегии (алгоритмы) управления памятью heap (monolithic heap, generational heap).

- Q Что входит в состав HotSpot VM (версии 1.6 или старше)? Какие флаги JVM позволяют конфигурировать работу GC? Какие Generation Collectors существуют в HotSpot? Объясните принцип работы GC G1, приведите достоинства и недостатки этого GC.

## 2.12 Collections

- Q Приведите иерархию интерфейсов Collections Framework, укажите основные отличия между ними.
- Q Приведите иерархию интерфейсов Set. Перечислите основные конкретные классы, реализующие интерфейс Set. Укажите, какие структуры данных лежат в основе этих реализаций.
- Q Перечислите основные конкретные классы, реализующие интерфейс List. Укажите, какие структуры данных лежат в основе этих реализаций.
- Q Приведите иерархию интерфейсов Map. Перечислите основные конкретные классы, реализующие интерфейс Map. Укажите, какие структуры данных лежат в основе этих реализаций.
- Q Что такое Map.Entry? Как получить коллекцию этих сущностей? Как происходит модификация(изменения значения) сущностей Map.Entry?
- Q Что такое итератор. Приведите примеры взятия и использования итераторов для коллекций. Чем интерфейс Iterator отличается от интерфейса ListIterator? Укажите ситуации, когда взятый для коллекции итератор становится невалидным.
- Q Перечислите известные вам методы класса Collections, для чего они служат.
- Q Что такое Legacy Collections. Какие классы унаследованных коллекций вы знаете? Укажите характеристики этих классов.