

# ***1. Basics of software code development***

## ***1. Объясните, что имеется в виду, когда говорится: Java-язык программирования и Java-платформа.***

**Язык Java** — это статически типизированный объектно-ориентированный язык, над которым мы немного пошутили в разделе «Об этой книге». Надеемся, что вы уже довольно хорошо знакомы с ним. Одно из самых очевидных качеств языка Java заключается в том, что он пригоден для чтения человеком (или, по крайней мере, должен таким быть!).

**Платформа Java** — это программное обеспечение, предоставляющее нам среду времени исполнения. Это виртуальная машина Java (JVM), линкующая и выполняющая ваш код в том виде, в каком он ей предоставляется. Код предоставляется в виде файлов классов, непригодных для чтения человеком. Иными словами, машина не интерпретирует непосредственно файлы с исходным кодом на языке Java, а требует предварительного преобразования этого кода в файлы классов.

## ***2. Поясните, как связаны имя java-файла и класса, которые в этом файле объявляются.***

Каждый открытый класс должен находиться в файле, где `FileName` соответствует `ClassName` и пакету, где `Packagename` представляет структуру `Directory`, написанную в пунктирной форме (слэши становятся точками, например `com / example / app` становится `com.example.app`).

Это соглашение не является случайным. Компилятор должен иметь возможность находить исходные файлы, и загрузчик классов должен иметь возможность найти реализацию. Соответствие имен пакетов и имен классов делает это очень простым и, что более важно, быстрым.

Это соглашение не применяется к непубличным классам. Это связано с тем, что непубличные классы имеют очень ограниченную видимость и могут использоваться только в пакете, где они определены. Таким образом, в обоих случаях компилятор и среда выполнения уже располагают правильными файлами.

## ***3. Расшифруйте аббревиатуры JVM, JDK и JRE; покажите, где они находятся и что собой представляют.***

JDK (Java Development Kit) нужен для разработки (это компилятор, отладчик и т.д.).

JRE (Java Runtime Environment) нужен для запуска Java программ (содержит в себе JVM).

JDK и JRE содержат JVM, которая нужна для запуска программ на Java.

JVM (Java Virtual Machine) является сердцем языка программирования Java и обеспечивает независимость от платформы.

**4. Объясните, как скомпилировать и запустить приложение из командной строки, а также зачем в переменных среды окружения прописывать пути к установленному jdk.**

**Скомпилировать** `javac filename.java`

**Запустить** `java filename`

Чтобы было можно компилировать и выполнять программы в переменных среды окружения прописывают пути к установленному jdk.

**5. Перечислите атрибуты доступа, объясните их действия.**

`private;`

`protected;`

`default (package visible);`

`public.`

**6. Что такое пакеты в java-программе, что представляют собой пакеты на диске? Каково соглашение по именованию пакетов. Как создавать пакет?**

Пакет является специальным библиотечным модулем, который содержит группу классов, объединённых в одном пространстве имён.

Пакеты на диске представляют обычные папки.

Имена пакетов записываются во всех строчных, чтобы избежать конфликтов с именами классов или интерфейсов. Для написания пакетов используют свое обратное доменное имя в Интернете.

Создаем пакет `new | Package`, либо при создании класса `packageName.Class`.

**7. 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.**

```
java.util.Date today = new java.util.Date();
```

```
import java.util.*;
```

При импорте пакета имеется риск непреднамеренного импорта неправильного класса, если используется \* нотация. Как правило, это не проблема, но если \* соответствует классу, который хотели извлечь из другого пакета, может возникнуть ошибка, которую сложно будет найти.

компилятор неявно импортирует `java.lang` для всех программ.

**8. Объясните различия между терминами «объект» и «ссылка на объект».**

Объект-это экземпляр класса, а ссылка- это переменная, которая позволяет обратиться к объекту или классу, но в тоже время не является экземпляром класса.

**9. Какие примитивные типы Java вы знаете, как создать переменные примитивных типов? Объясните процедуру, по которой переменные примитивных типов передаются в методы как параметры.**

В Java есть 8 примитивных типов, которые делят на 4 группы:

- Целые числа — `byte`, `short`, `int`, `long`;
- Числа с плавающей точкой (иначе дробные) — `float`, `double`;

- Логический — boolean;
- Символьный — char.

В Java существует 2 способа для передачи переменной или объекта класса в функцию:

- передача по значению. В этом случае значение аргумента копируется в формальный параметр функции. Поскольку создается копия аргумента в функции, то все изменения над копией не повлияют на значение аргумента;

- передача по ссылке (по адресу). В этом случае параметру передается ссылка на аргумент, который используется при вызове. По этой ссылке есть доступ к аргументу. Таким образом, все изменения, сделанные в теле функции над значением параметра, будут изменять значение аргумента который был передан в функцию.

**10. 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.**

Размер в битах: byte – 8; short и char – 16; int и float – 32; long и double – 64; boolean – 1 или 32.

У всех типов размер памяти не зависит от разрядности.

В Java возможны преобразования между целыми значениями и значениями с плавающей точкой. Кроме того, можно преобразовывать значения целых типов и типов с плавающей точкой в значения типа char и наоборот, поскольку каждый символ соответствует цифре в кодировке Unicode.

Фактически тип boolean является единственным примитивным типом в Java, который нельзя преобразовать в другой примитивный тип. Кроме того, любой другой примитивный тип нельзя преобразовать в boolean.

**11. Объясните, что такое явное и неявное приведение типов, приведите примеры, когда такое преобразование имеет место.**

Неявное преобразование типов выполняется в случае если выполняются условия:

- Оба типа совместимы
- Длина целевого типа больше или равна длине исходного типа

Во всех остальных случаях должно использоваться явное преобразование типов.

`short s = (short) 0xffff; // Данные биты представляют число – 1`

`char c = '\uffff'; // Те же биты представляют символ юникода`

`int i1 = s; // Преобразование типа short в int дает –1`

`int i2 = c; // Преобразование char в int дает 65535`

**12. Что такое литералы в Java-программе, какую классификацию литералов вы знаете, как записываются литералы различных видов и типов в Java-программе?**

Литералы — это явно заданные значения в коде программы — константы определенного типа, которые находятся в коде в момент запуска.

Все литералы — это примитивные значения (строки, числа, символы, булевы значения). Нельзя создать литерал-объект. Единственный литерал, связанный с объектом — это null.

```
int a = 0b1101010110; в двоичной системе
int b = 012314; в восьмеричной системе
int c = 456; в десятичной системе
int d = 0x141D12; в шестнадцатеричной системе
long b = 1342352352351351353L;
double a = 2.718281828459045;
double d = 4.05E-13;
float d = 4.05E-13F
```

### **13. Как осуществляется работа с типами при вычислении арифметических выражений в Java?**

Все целочисленные литералы в выражениях, а также типы byte, short и char расширяются до int (если в выражении не присутствуют более большие типы данных - long, float, double).

Есть унарные операции (выполняются над одним операндом), бинарные - над двумя операндами, а также тернарные - выполняются над тремя операндами. Операндом является переменная или значение (например, число), участвующее в операции.

### **14. Что такое классы-оболочки, для чего они предназначены? Объясните, что значит: объект класса оболочки – константный объект.**

Классы-оболочки Java являются Объектным представлением восьми примитивных типов в Java. Все классы-оболочки в Java являются неизменными и final.

### **15. Объясните разницу между примитивными и ссылочными типами данных. Поясните существующие различия, при передаче параметров примитивных и ссылочных типов в метод. Объясните, как константные объекты ведут себя при передаче в метод.**

В Java есть 2 группы типов: ссылочные и примитивные. Ссылочные являются наследниками класса Object и передаются в методы(функции) по ссылке, т.е. если изменить в функции ссылочный тип - то они изменятся и в точке вызова (в месте, откуда вы его передали). Примитивные же передаются по значению: т.е. что бы не сделать с передаваемым примитивным типом в функции/методе, это всё навсегда останется внутри функции/метода, и снаружи (в точке вызова) он (и его значение) останется таким же после отработывания функции.

### **16. Поясните, что такое автоупаковка и автораспаковка.**

Автоупаковка и распаковка это функция преобразования примитивных типов в объектные и наоборот.

### **17. 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.**

Арифметические операторы: + - \* / % ++ --

Побитовые операторы: & (побитовое и), | (побитовое или), ^ (побитовое логическое или), ~ (побитовое дополнение), << (сдвиг влево), >> (сдвиг вправо), >>> (нулевой сдвиг вправо).

Логические операторы: &&, ||, !.

Приоритет операторов: Постфикс, Унарный, Мультипликативный, Аддитивный, Сдвиг, Реляционный, Равенство, Побитовые, Логические, Условный, Присвоение, Запятая.

**18. Укажите правила выполнения операций с плавающей точкой в Java (согласно стандарту IEEE754). Как определить, что результатам вычисления стала бесконечность или нечисло.**

Операций с плавающей точкой в Java, если нужна точность выполняем с помощью BigDecimal. В классе BigDecimal сложение осуществляется с помощью метода add(). умножение осуществляется с помощью метода multiply(). объекты чисел нужно сравнивать между собой с помощью специального метода compareTo().

Если хотим проверить, что нечисло или бесконечность, то нужно использовать статические методы Float.isNaN(float) или Double.isInfinite(res), определённые в пакете java.lang.

**19. Что такое статический импорт, какие элементы можно импортировать при статическом импорте.**

Оператор import, предваряемый ключевым словом static, можно применять для импорта статических членов класса или интерфейса. Благодаря статическому импорту появляется возможность ссылаться на статические члены непосредственно по именам, не уточняя их именем класса.

**20. Объясните работу операторов if, switch, while, do-while, for, for-each. Напишите корректные примеры работы этих операторов.**

Инструкции (операторы) в программе выполняются сверху вниз по исходному файлу. Операторы управления порядком выполнения позволяют прервать обычный ход выполнения, позволив выполнить один кусок кода несколько раз, выполнить кусок кода только при выполнении определённого условия.

**21. Объясните работу оператора instanceof. Что будет результатом работы оператора, если слева от него будет стоять ссылка, равная null?**

Оператор instanceof нужен, чтобы проверить, был ли объект, на который ссылается переменная X, создан на основе какого-либо класса Y.

Использование нулевой ссылки в качестве первого операнда для instanceof возвращает false.