

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

Java-язык программирования — это программный код, которым программист описывает логику поведения программы, ее интерфейс и т.д.

Java-платформа — это среда выполнения JDK. JRE, Javac и стандартные библиотеки Java,

2. Поясните, как связаны имя Java-файла и классы, которые в этом файле объявляются.

Java файлы всегда называются именем класса.

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

Java Development Kit - набор инструментов, необходимый для написания программ. Включает в себя JVM, JRE, библиотеки, JAVAC (компилятор java-кода в байт-код).

Java Runtime Environment, сокращенно JRE – это исполнительная среда Java в которой выполняются программы, написанные на этом языке. Среда состоит из виртуальной машины – Java Virtual Machine(JVM) и библиотеки Java классов.

Java Virtual Machine, сокращенно JVM – это виртуальная машина Java — основная часть исполняющей среды JRE. Виртуальная машина Java интерпретирует и исполняет байт-код Java. Байт код получают посредством компиляции исходного кода программы с помощью компилятора Java (стандартный - javac).

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

Компиляция программы из командной строки: `javac filename.java`

На выходе получаем `filename.class` содержащий байт-код

Запуск программы из командной строки: `java filename`

Переменные среды нужны для того, чтобы обращаться к нужному исполняемому файлу по имени переменной.

5. Перечислите атрибуты доступа, объясните их действие.

В Java используются следующие модификаторы доступа:

`private`: Доступ разрешен только в текущем классе.

`default (package-private)`: - Доступ на уровне пакета. Если класс будет так объявлен он будет доступен только внутри пакета.

`protected`: Модификатор доступа на уровне пакета и в иерархии наследования.

`public`: - Модификатор доступен всем.

Последовательность модификаторов по убыванию уровня закрытости: `private`, `default`, `protected`, `public`).

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

Пакеты позволяют организовать классы логически в наборы. Пакеты на диске представлены в виде папок. Чтобы создать пакет, нужно в файл перед определением классов вставить следующую строку: `package ИмяПакета`;

Имена пакетов должны быть в нижнем регистре. Используется инвертированное имя домена `com.epam.package`

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?

Импортировать нужно те пакеты или классы, которых нет в `java.lang` (он импортируется по умолчанию), но которые используются в нашем проекте. Чтобы импортировать есть оператор `import java.util.Date` или `import java.util.*` (чтобы импортировать всё содержимое пакета).

8. Объясните различия между терминами "объект" и "ссылка на объект".

Объект — это экземпляр класса, создается при помощи кодового слова `new`. Ссылка же хранит адрес ячейки памяти, в которой хранится объект.

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

- `byte` (целые числа, 1 байт, [-128, 127])
- `short` (целые числа, 2 байта, [-32768, 32767])
- `int` (целые числа, 4 байта, [-2147483648, 2147483647])
- `long` (целые числа, 8 байт, [-9223372036854775808, 9223372036854775807])
- `float` (вещественные числа, 4 байта)
- `double` (вещественные числа, 8 байт)
- `char` (символ Unicode, 2 байта, [0, 65536])

- `boolean` (значение истина/ложь, используется `int`, зависит от JVM)

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 типов не зависит от разрядности системы.

Преобразование может быть неявным и явным (приведение типов). Неявное преобразование может выполняться если:

1. типы совместимы (например — оба целочисленные)
2. размер «принимающего» типа больше чем у того, который преобразуется (так называемое «преобразование с расширением»)

`boolean` не приводится ни к одному другому типу

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

Явное преобразование имеет вид:

```
int a;  
byte b = (byte) a; //b будет остатком от деления a на диапазон byte, может быть потеря данных
```

Неявное преобразование может выполняться если:

1. типы совместимы (например — оба целочисленные)
2. размер «принимающего» типа больше чем у того, который преобразуется (так называемое «преобразование с расширением»)

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

Любая константа в Java является литералом.

Классификация литералов:

- Целочисленные литералы
- Литералы с плавающей точкой
- Символьные литералы

- Строковые литералы
- Булевы литералы

13. Как осуществляется работа с типами при вычислении арифметических выражений в java?

При вычислении арифметических выражений тип конечного результата продвигается до int для меньших типов либо до double для Long. При этом при вычислении выражения с разными типами, конечный результат продвигается до большего типа среди всех переменных.

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. Классы-оболочки позволяют работать с примитивными типами, как с объектами, что упрощает работу.

15. 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.

Ссылочные типы данных ссылаются на объекты. У них есть методы, облегчающие работу с ними.

Java is always pass-by-value - Java передает всё по значению. Для примитивов передаются копии значений. Для объектов в качестве значений выступают ссылки на эти объекты.

- Строки (String) immutable, т.е. неизменяемые. Когда мы присваиваем строке новое значение, то всегда будет создан новый объект в памяти.

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: +, -, *, /, %, ++, --

Логические: &&, ||, !

Битовые: & И (AND), | ИЛИ (OR), ^ исключающее ИЛИ (XOR), >> сдвиг вправо, >>> сдвиг вправо с заполнением нулями, << сдвиг влево, ~ побитовое унарное отрицание (NOT). Если выражение включает несколько операторов с одинаковым приоритетом, то порядком выполнения операций управляет ассоциативность операторов. Большинство операторов ассоциативны слева направо, то есть операции выполняются слева направо. Однако операторы присваивания и унарные операторы обратны ассоциативны (справа налево).

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

Язык Java поддерживает два простых типа с плавающей точкой: float и double, и их прототип - класс-оболочка Float и Double. Они основаны на стандарте IEEE 754, определяющем двоичный стандарт для двоично-десятичных чисел с 32-битовой плавающей точкой и 64-битовой плавающей точкой удвоенной точности.

IEEE 754 представляет числа с плавающей точкой как десятичные числа с основанием 2 в экспоненциальном формате. В IEEE числе с плавающей точкой выделяется 1 бит на знак, 8 бит на порядок и 23 бита на мантиссу, или дробную часть числа. Порядок расшифровывается как целое число со знаком, допускающее как положительный, так и отрицательный экспоненты. Дробь представляется как двоично-десятичное (основание 2) число, где самый старший бит соответствует значению $\frac{1}{2}$ (2^{-1}), следующий бит $\frac{1}{4}$ (2^{-2}), и так далее. Для плавающей точки с удвоенной точностью на порядок выделяется 11 бит, а на мантиссу - 52 бит.

В Java тип double имеет специальные значения для понятий «плюс бесконечность» и «минус бесконечность». Положительное число, разделенное на 0.0, дает «плюс бесконечность», а отрицательное – «минус бесконечность».

NaN одно из особых состояний

https://d.docs.live.net/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D1%81_%D0%BF%D0%BB%D0%B0%D0%B2%D0%B0%D1%8E%D1%89%D0%B5%D0%B9_%D0%B7%D0%B0%D0%BF%D1%8F%D1%82%D0%BE%D0%B9 числа с плавающей запятой. Используется во многих математических библиотеках и

https://d.docs.live.net/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%81%D0%BE%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80 математических сопроцессорах. Данное состояние может возникнуть в различных случаях, например, когда предыдущая математическая операция завершилась с неопределённым результатом или если в ячейку памяти попало не удовлетворяющее условиям число.

Например

```
double ZERO = 0;
```

```
System.out.println("ZERO / ZERO = " + (ZERO / ZERO));
```

```
ZERO / ZERO = NaN
```

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. При объявлении статического импорта статические члены импортируются из классов, что позволяет им быть использованными без указания имени содержащего их класса.

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

if - оператор условия, например

```
if (nubmer == 1){  
(выполняются действия в скобках)  
}
```

switch - оператор выбора

```
switch(month) {  
case 1:
```

```
monthString = "Январь";  
break;  
default: monthString = "Нет такого";  
}
```

while - пока условие истина выполняем while (number =1){действия}
do while (тоже что и while, только сначала выполняет а потом проверяет)
for - оператор цикла for (a=1; a< 3; a++){действия}
for-each - int[] array = {51,136,387};
for (int i:array) {
System.out.println(i);
}

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

Используя оператор instanceof, можно узнать, от какого класса произошел объект. Этот оператор имеет два аргумента. Слева указывается ссылка на объект, а справа — имя типа, на совместимость с которым проверяется объект.

```
String s; s = ""; System.out.println((s instanceof String));  
вывод true
```

```
s = null; System.out.println((s instanceof String));
```

вывод false

Если слева будет стоять null, то instanceof для любого типа вернёт false.