Язык программирования Java был разработан в 1990 году инженером компании Sun Microsystems Джеймсом Гослингом. Первоначальное название языка было – Oak (в переводе с английского – дуб). В 1995 году язык был переименован в Java.

Основная сущность языка Java – это библиотеки файлов, которые называются классами. Каждый класс содержит небольшой законченный кусок кода. Программа на Java создается из таких вот небольших модулей. Это позволяет легче отлаживать готовые программы.

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

Дело в том, что Java – это одновременно и язык программирования и платформа для выполнения написанной программы. Исходный код программы сохраняется с расширением .java. Потом этот код компилируется в файл с расширением .class. И уже этот файл виртуальная машина (JVM) выполняет.

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

Имя файла и имя класса должны полностью совпадать. Если этого не будет, то при компиляции будет возникать ошибка. В одном .java файле содержится один класс

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

JVM – Java Virtual Machine – виртуальная машина Java. Это программа, которая установлена на компьютере и которая выполняет код из файла .class

JRE – Java Runtime Environment – среда выполнения Java. Это пакет инструментов для запуска программ, написанных на Java. JRE создает и запускает виртуальную машину Java (JVM).

JDK – Java Development Kit – пакет разработчика для создания программ на языке Java. В этот комплект входят все необходимые библиотеки классов, которые уже написаны другими разработчиками. Входит компилятор для преобразования файлов .java в .class. Сюда также включены и JRE и JVM. Данный пакет предназначен для разработчиков.

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

Путь к установленному JDK необходимо прописывать в переменных среды окружения для того, чтобы система (Windows, Linux, Mac) понимала где лежат исполняемые файлы среды разработки чтобы их можно было запустить из любого «места» на компьютере, а не только при нахождении в подпапке /bin.

Для запуска приложения необходимо сохранить файл с расширением .java. После этого исходный код необходимо скомпилировать и потом запустить.

Для компиляции исходного кода запускаем терминал (или командную строку), в терминале переходим в тот каталог, где лежит исходный код программы и вводим команду:

**javac имя\_файла.java**

В результате будет создан файл с расширением .class, который можно запустить командой:

**java имя\_файла** (без расширения .class)

***Перечислите атрибуты доступа, объясните их действие***

private –

public –

static –

protected –

В языке Java имена переменных могут начинаться только с латинской буквы, знака $, либо знака \_. Все имена чувствительны к регистру. Пробелы не допускаются. Первое слово начинается с маленькой буквы, последующие с большой (Camel Case).

Переменные с типом **float** обязательно задаются с префиксом **f,** иначе компилятор принимает такую переменную как тип **double**.

Константы задаются ключевым словом **final** и их принято описывать в верхнем регистре.

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

***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?***

***Объясните различия между терминами “объект” и “ссылка на объект”.***

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

К примитивным типам данных относятся: **byte, short, int, long, char, float, double, boolean**. Для создания переменной сначала указывается ее тип, затем имя переменной. При необходимости можно сразу задать ее значение.

При вызове какого-либо метода в скобках можно задать параметры, которые мы хотели бы передать для работы в этом методе. Внутри метода эти переменные становятся доступны и мы можем на основании их значений производить какие-либо действия. При создании метода в скобках после имени метода мы задаем количество, типы и имена переменных, которые мы будем потом передавать в наш метод. Переменные передаются в метод в виде ссылки на исходную переменную.

***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.***

Размеры примитивных типов:

* byte — 8 бит
* short — 16 бит
* char — 16 бит
* int — 32 бит
* long — 64 бит
* float — 32 бит
* double — 64 бит
* boolean — 32 бит

Использование в программе примитивного типа будет зависеть от разрядности платформы, на которой будет работать приложение. Нельзя будет использовать 64 битные типы на 32 разрядной платформе

Преобразование (приведение) типов — это преобразование типа из одного в другой. Например из типа int в long или наоборот. При преобразовании типов возможны ситуации, когда полученное значение будет отличаться от исходного. Это связано с тем, что разные типы имеют разную длину. Очень часто используется преобразование из целочисленных типов в строковые и наоборот.

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

Явное преобразование типов — это ситуация, когда мы в программе явно задаем такое преобразование. Например, (int) var, (long) var, (double) var

Неявное преобразование типов — это ситуация, когда в процессе выполнения программы производятся операции над различными типами переменных и программа автоматически их преобразовывает. Самый яркий пример — это когда мы к строкой переменной прибавляем целочистенную переменную. В этом случае целочисленная переменная автоматически приведется к строковому типу.

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

***Как осуществляется работа с типами при вычислении арифметически выражений в Java?***

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

***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.***

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

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

Логические операторы: >, >=, <, <=, ==, !=, &&, ||, !

Битовые операторы: &, |,

Приоритет операторов — это порядок, в котором будут выполняться операции. Если несколько операторов имеют одинаковый приоритет, то они будут выполняться по порядку слева направо. Приоритет выполнения операторов можно изменить с помощью скобок.

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

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

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

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