Basics of software code development

1.Платформа Java это ряд программ, которые позволяют запускать и разрабатывать программы, написанные на языке Java она разработана так чтобы работать на разных платформах одинаково. А язык программирования Java это формальный язык, предназначенный для создания компьютерных программ как C++ и другие.

2.Имя файла должно совпадать с именем public класса который объявляется в этом классе.

3.JWM –Java Virtual Machine она исполняет байт-код, который генерирует компилятор из кода, который мы написали, она является основной частью исполняющей системы Джава JRE

JRE - Java Runtime Environment это среда выполнения Джава приложений (минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений, без компилятора)

JDK-Java Development Kit-программное обеспечение необходимое программисту для написания программы включает в себя JRE, компилятор и библиотеки с примерами уже созданными классами.

4. Первое действие открыть командную строку и javac “имя нашего java файла в котором находится код”.java

Второе действие написать java и имя скомпилированного класса обычно оно такое, как и у файла, который вы выбрали для компиляции.

5.public-к переменной/классу, помеченному как public можно получить из любого места программы.

Private –к переменной/классы можно обращаться только внутри класса в котором он объявлен для всех остальных он невиден. Такие методы нельзя наследовать и нельзя переопределять.

Default- модификатор по умолчанию методы помеченные этим модификатором видны всем классам одного пакета однако вне этого пакета она невидимы. Класс наследники также не могут к нему обращаться.

Protected-тоже самое что и default но классы наследники имеют доступ.

Protected- доступен наследникам и в пакете.

6.Если java класс находится в некотором пакете в, то в проекте класс будет находится в папке с названием идентичным названию пакета. Имена пакетов должны быть записаны нижем регистре, и первая часть имени пакета должна состоять из перевернутого имени разработчика класса. В разных пакетах одного проекта могут находится классы с одним и тем же именем. Создается пакет через команду New| Package или, когда вы создаёте в студии

новый класс, то указывая его имя можете использовать точечную нотацию, например, database.ClassB. В этом случае пакет database будет создан автоматически и в нём будет находиться создаваемый класс. Пакет в Java это библиотечный модуль, который содержит группу классов, объединённых в одном пространстве имен.

7.импортирование происходит через команду import если использовать звездочку, то можно импортировать не только 1 метод, а весь класс.

8.объект экземпляр класса. Ссылка на объект- это переменная содержащая адрес ячейки памяти в которой хранится объект. Ссылка позволяет нам обратится к экземпляру класса, но сама экземпляром она не является.

9. Существует 8 примитивных типов byte, short, int, long, float, double, boolean, char.

Создается по след примеру Int x. Пример передачи в метод как параметр void prim (int a) {} пишем в скобочках какой тип нам нужен и как он будет называться в нашем методе, в который мы передаем.

10. Размеры примитивных типов определяется в битах, Размеры примитивных типов не зависят от разрядности JVM. Приведение типа преобразование значения одного типа в значение другого, например, double в int в таком случае откидывается часть после запятой.bollean нельзя преобразовать ни к одному другому типу у него только 2 значения true и false.

byte	8 бит	от -128 до 127
short	16 бит	от -32768 до 32767
char	16 бит	беззнаковое целое число, представляющее собой символ UTF-16 (буквы и цифры)
int	32 бит	от -2147483648 до 2147483647
long	64 бит	от -9223372036854775808L до 9223372036854775807L
float	32	от 1.4e-45f до 3.4e+38f

double	64	от 4.9e-324 до 1.7e+308

11.явное приведение short b=1; int a= (int)b; Неявное a=b при неявном мы можем переводить только меньший тип в больший, но не наоборот иначе компилятор выдаст ошибку, а при явном он ее не выдаст. Явное нам нужно тогда, когда мы хотим, например, обрезать лишнюю дробную часть это легко можно сделать через приведение, но компилятор нам не позволит сделать это неявно поэтому мы используем явное приведение.

12.Литерал — это некое фиксированное значение, записанное в коде программы. Все литералы — это примитивные значения (строки, числа, символы, булевы значения). Нельзя создать литерал-объект. Литералы делятся на числовые (целочисленные, с плавающей запятой), строковые, символьные, логические. Числовые, например, можно записывать числа в разных системах счисления, но обрабатываться в коде они будут как десятичные. Любой целочисленный литерал имеет тип int если он выходит за границу, то выдается ошибка. У long в конце ставя букву L.y float букву f и т.д. Символьные литералы записываются в виде ‘@’ или \ (и некий код числа в 4 символа) логический литерал true or false.

13 При вычислении компилятор смотрит на то какой тип у чисел сейчас и выдает ответ того же типа, например, int a=10 int b=4 10/4=2 потому что мы делили int на int, и компилятор округлил ответ для того чтобы привести его к int, а вот если бы мы делили бы double, то ответ был бы 2,5.

14.Классы оболочки — это объекты аналоги примитивных типов. Например, int Integer, например, они нам нужны, когда мы хотим создать коллекцию интов мы пишем List<Integer> list = new ArrayList<>(); и создаётся коллекция для хранения int по факту классы оболочки — это те же самые примитивные типы только с куда большим функционалом. Объект класса оболочки-константный объект. Константный объект — это значение примитива, для которого объект является оболочкой. К примеру, для int - оболочка java.lang.Integer. По факту это работает так если бы объявим переменную Integer a=1, а потом напишем a=2 то a будет указывать на новый объект, а не перезапишет предыдущее значение.

15.Примитивные типы данных хранят фактическое значение, а ссылочные хранят адреса объектов, на которые они ссылаются (например, на адресное пространство в котором находится наше число). В Java переменные передаются в методы по ссылке передача константного объекта в метод означает что мы можем менять его, а его ссылку нет. А вот примитивные типы передаются по значению и если они const, то менять мы их не сможем.

16.Автоупаковка и автораспаковка –это преобразование примитивных типов в объекты и наоборот. Автоупаковка –это автоматическая функция преобразования примитивных типов в эквивалентные им объекты. Автораспаковка наоборот. Срабатывает при передаче перем. методу или присваивании автоматичвески.

17.Арифметические операторы: -, +, /, * логические операторы: &&, ||, +,! и т.д. битовые операторы :>>, ^, & и т.д. Приоритет оператора — это то в каком порядке будут выполняться действия над значением как в математике * имеет больший приоритет чем + или -. В случае если операторы будут иметь одинаковый приоритет, то действия будут идти с лева на право.

18. Стандарт IEEE 754 говорит нам, что представление действительных чисел должно записываться в экспоненциальном виде. Это значит, что часть битов кодирует собой так называемую мантиссу числа, другая часть — показатель порядка (степени), и ещё один бит используется для указания знака числа (0 — если число положительное, 1 — если число отрицательное).

Деление на 0 чисел с плавающей точкой является бесконечность (или NaN в случае деления 0.0/0). Согласно стандарту IEEE 754 ввели также и значения NEGATIVE_INFINITY и POSITIVE_INFINITY, равные -1.0 / 0.0 и 1.0 / 0.0 соответственно.

19.Статический импорт позволяет ссылаться на статические члены непосредственно по именам, не уточняя их именем класса. Например, есть статический метод pow обычно его вызывают так Math.pow, но, если использовать статический импорт “import static java.lang.Math.pow;” благодаря этой строке в дальнейшем мы сможем писать просто pow, и он будет работать. При статическом импорте можно импортировать статические члены класса или интерфейса. Коротка говоря оно работает как using namespace std.

20. if() смотрит на выражение, записанное в скобках и если оно true, то выполняет некоторое действие. Swith() принимает некоторое значение и case-ми выбирает какое действие выполнять.while-работает пока не перестанет сущ. то что записано в скобках.do while-выполняет то что записано в кавычках после do пока работает while. For() пербор пока что-то не станет равным чему-то. For-each выполняет некоторое действие над каждым элементом в масииве.

21.instanceof- позволяет проверить к какому классу принадлежит объект с учетом наследования. Если слева будет стоят null, то выдаст false.