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

* Java-язык – высокоуровневый строго типизированный объектно-ориентированный язык программирования общего назначения с С-подобным синтаксисом.
* Java-платформа – это программное обеспечение, представляющее собой рабочую среду для работы программ, написанных на Java.

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

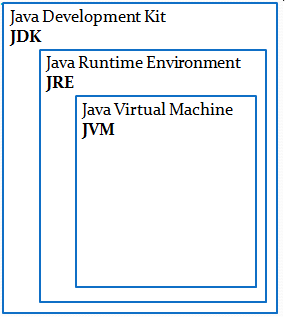
- Каждый класс хранится в отдельном файле. Имя файла совпадает с именем класса.

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

- JVM (Java Virtual Machine) - программа, которая исполняет\выполняет код и предоставляет среду исполнения\выполнения для этого кода;

- JRE (Java Runtime Environment) - пакет, который включает в себя JVM и минимальный набор библиотек для работы программ;

- JDK (Java Development Kit) - полноценный набор библиотек и инструментов для создания, компилирования и дебага программ;

****

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

- компиляция – прописываем в КС следующее:

javaс /путь к исполняемому файлу/имя файла.java

- запуск - прописываем в КС следующее:

java /путь к исполняемому файлу/имя файла

- прописывание в переменных среды окружения пути к установленному jdk позволяет вызывать в КС сам компилятор и другие исполняемые программы (javac.exe, java.exe).

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

- public: общедоступный класс или член класса. Поля и методы видны другим классам из текущего пакета и из внешних пакетов.

- private: закрытый класс или член класса доступен только из кода в том же классе.

- protected: такой класс или член класса доступен из любого места в текущем классе или пакете или в производных классах, даже если они находятся в других пакетах.

- Отсутствие модификатора у поля или метода класса предполагает применение к нему модификатора по умолчанию. Такие поля или методы видны всем классам в текущем пакете.

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

- Классы в Java (каждый класс лежит в отдельном файле) группируются по пакетам, которые являются папками на диске. Существует общепринятая схема, где первая часть имени пакета должна состоять из перевёрнутого доменного имени разработчика класса. Так как доменные имена в интернете уникальны, соблюдение этого правила обеспечивает уникальность имён пакетов и предотврати конфликты. Для создания пакета необходимо выбрать ему имя и включить оператор «package» вместе с именем исходного файла, который содержит классы. (после package нужно указать полное имя пакета, с корня пакета до пакета(папки) в которой размещен класс. Каждая отдельная папка, в полном имени пакета, отделяется точкой. Имя самого класса не входит в имя объявляемого пакета. Реальное расположение файла и полное имя пакета, после слова package, — должны соответствовать друг другу.).

1. **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?**

- если для работы приложения необходимо использовать все классы из собственного пакета и все общедоступные классы из других пакетов, то их можно импортировать при помощи оператор «import». Импортировать можно как отдельные классы, так и весь пакет. Пакет java.lang импортируется по умолчанию.

1. **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. Переменная, ссылающаяся на объект, хранит биты, которые описывают путь для доступа к объекту или классу.

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

- boolean

- byte

- short

- int

- long

- char

- float

- double

Для объявления переменной примитивного типа необходимо задать:

Тип примитивной переменной + имя объявляемой переменной;

Переменные примитивных типов передаются в методы по ЗНАЧЕНИЮ. Т.Е. параметр копируется в метод.

1. **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.**

-- boolean – булево значение true или false

- byte - 8-разрядное целое число

- short - 16-разрядное целое число

- int - 32-разрядное целое число

- long - 64-разрядное целое число

- char - 16-разрядное беззнаковое целое, представляющее собой символ UTF-16 (буквы и цифры)

- float - 32-разрядное число в формате IEEE 754 с плавающей точкой

- double - 64-разрядное число в формате IEEE 754 с плавающей точкой

Размер примитивных типов не зависит от разрядности платформы.

Преобразование (приведение) – присвоение переменной одного типа значения переменной другого типа.

Тип boolean не приводится ни к одному из типов.

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

- Явное преобразование может потребоваться в тех случаях, когда значение типа позволяющего хранить большее количество знаков надо привести к типу, способному хранить меньшее количество знаков числа.

int a = 258;

byte b = (byte) a; // от типа int к типу byte

System.out.println(b); // 2

Неявное (автоматическое) преобразование соответственно наоборот.

byte b = 7;

int d = b; // преобразование от byte к int

1. **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.

- В соответствии с примитивами, литералы также делятся на разделы:

Числовые:

Целочисленные (Числа записываются в их стандартном виде без указывающих символов и прочего.);

С плавающей точкой (записываются в классическом виде десятичной дроби или в научном виде (экспоненциальную запись записывают в виде MEp (пример 1e-10), где:

M — мантисса,

E (exponent) — буква E в числе, означающая «\*10^» («…умножить на десять в степени…»),

p — порядок.);

Строковые (набор символов, заключенных в двойные кавычки);

Символьные (в Java представлены кодовой таблицей Unicode);

Логические (2 значения false и true).

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

- В арифметических выражениях автоматически выполняется неявное (расширяющее) преобразования типов. Byte->short->int->long->float->double.

(Осуществляется через унарные и бинарные знаки.

Унарные с одним аргументом :

• унарный минус "-" – меняет знак числа или выражения на противоположный;

• унарный плюс "+" – не выполняет никаких действий над числом или выражением;

• побитовое дополнение "~" (только для целых) – инвертирует все биты поля числа (меняет 0 на 1 и 1 на 0);

• инкремент "++" (только для целых) – увеличивает значение переменной на 1;

• декремент "--" (только для целых) – уменьшает значение переменной на 1.

Бинарные с двумя аргументами:

• сложение "+";

• вычитание "-";

• умножение "\*";

• деление "/";

• вычисление остатка от деления целых чисел "%")

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

- Класс-оболочка - это специальный класс, который хранит внутри себя значение примитива. Они хранят внутри нужные значения примитивов, при этом будут являться настоящими объектами. Класс оболочки является неизменяемым классом, его состояние не может быть изменено после создания.

1. **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.**

- Примитивные типы у себя внутри хранят значение, а ссылочные – ссылку на бъект. Соответственно в метод передается копия значения и копия ссылки.

Константные объекты при передаче в метод дают возможность считывать и использовать их значение без изменения самого объекта.

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

- Автоупаковка и распаковка это функция(механизм) преобразования примитивных типов в бъектные и наоборот.

1. **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.**

- Бинарные арифметические операторы

+, +=, -, -=, \*,\*=, /, /=, %, %=.

- Унарные арифметические операторы

+, -, ++, --.

- логические

==

! — оператор отрицания;

&& — оператор логическое И;

|| — оператор логическое ИЛИ ;

- битовые

|, |=, &, &=, ^, ^=,>>, >>=, >>>, >>>=, <<, <<=, ~.

Приоритет оператора определяет порядок, в котором операторы выполняются. Операторы с более высоким приоритетом выполняются первыми. 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 обратно ассоциативны (справа налево).)

1. **Укажите правила выполнения операций с плавающий точкой в Java (согласно стандарту IEEE754)!!!!. -** *Тут скорее речь идёт об особенностях представления чисел с плавающей точкой в бинарном представлении и вычислений с ними. (*При сравнении чисел с плавающей точкой нельзя использовать ==.*)* **Как определить, что результатом вычисления стала бесконечность или нечисло? –** *Могут ли нам тут как-то помочь предопределённые константы классов-оболочек для типов чисел с плавающей точкой.*

|  |  |
| --- | --- |
| **Действия с бесконечностью:** | |
| Выражение | Результат |
| n ÷ ±Infinity | 0 |
| ±Infinity × ±Infinity | ±Infinity |
| ±(не ноль) ÷ | ±Infinity |
| Infinity + Infinity | Infinity |
| ±0 ÷ ±0 | NaN |
| Infinity - Infinity | NaN |
| ±Infinity ÷ ±Infinity | NaN |
| ±Infinity × 0 | NaN |

- Результат бесконечность\ -бесконечность будет при делении одного операнда типа double или float на 0. Нечисло (NaN) получается при извлечении квадратного корня из отрицательно операнда.

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

- Импорт статических членов класса или интерфейса. Благодаря статическому импорту появляется возможность ссылаться на статические члены непосредственно по именам, не уточняя их именем класса.

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

- Оператор if позволяет вашей программе в зависимости от условий выполнить оператор или группу операторов, основываясь на значении булевой переменной или выражения:

if (turn\_left) {

System.out.println(«милиционером станешь»);

}

else

System.out.println(“пожарным станешь”);

- Оператор switch (оператор множественного выбора) позволяет выполнять различные части программы в зависимости от того, какое значение будет иметь некоторая целочисленная переменная (её называют «переменной-переключателем»)

Int n=6;

switch (n)

{

case 1:

s = "Понедельник";

break;

case 2:

s = "Вторник";

break;

…………………………………..

case 7:

s = "Воскресенье";

break;

default:

s = "Неправильно введен день";

}

- Оператор цикла while выполняет часть программы до тех пор, пока условие выполнения цикла имеет истинное значение (true). Истинность логического выражения (условия) проверяется перед каждым шагом цикла, в том числе и перед первым. while так же называют оператором предусловия.

int i = 1;

while(i < 11){

System.out.println("i= " + i);

i++;

}

- В отличие от оператора while, do-while является оператором постусловия, который сначала выполняет тело цикла, а потом осуществляет проверку условия. Таким образом, тело цикла выполнится всегда хотя бы один раз.

int i = 1;

do{

System.out.println("i = " + i);

i++;

}while(i < 11);

- При использовании оператора for заранее известно, сколько раз необходимо осуществить повторение.

for (int i = 1; i <= 10; i++) {

System.out.print(i + " ");

}

- оператор for-each это разновидность цикла for используется для перебора элементов массива или коллекции.

int[] array = {51,136,387};

for (int i:array) {

System.out.println(i);

1. **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. Если слева от него будет стоять ссылка, равная null – результат false.