

基础语法面试题

1. 什么是 JVM? 为什么称 Java 为跨平台的编程语言?

Java 虚拟机(Java Virtual Machine)是可以执行 Java 字节码的虚拟机, 每个 Java 源文件将被编译成字节码文件, 然后在 JVM 中执行。Java 之所以被设计成可以在任意的平台运行, 而不需要重写或者在不同的平台下重新编译, 这些都要归功于 Java 虚拟机(JVM), 因为 JVM 非常了解特定的指令的长度以及底层硬件平台的特殊性。

2. JDK 和 JRE 之间的差异是什么?

Java 运行环境(Java Runtime Enviroment) 是运行 Java 程序的基本的 Java 虚拟机, 包括执行 applet 的浏览器插件。JDK (Java Development Kit) 是为了开发, 编译和执行 Java 应用程序, 针对 Java 的全功能的软件开发包, 包含了 JRE, 编译器和工具(比如说 JavaDoc 和 Java Debugger)。

3. “static” 关键字是什么意思? 在 Java 里可以 override private 或 static 的方法吗? keyword mean? Can you override private or static method in Java?

static 关键字表示, 访问这个成员变量或方法时, 不必获取它属于的类的实例。

Java 里的 static 方法不能被 override, 因为 override 的机制是运行时(runtime)的动态绑定, 而 static 方法是在编译时静态绑定的。static 方法并不与任何类的具体实例有关, 因此无法应用继承的概念。

4. 在静态方法里可以访问非静态变量吗?

Java 中的 static 变量归相应的类所有, 它的值对于类的所有实例都是相同的。static 变量是在 JVM 加载类的时候初始化的。如果代码试图访问非静态的变量, 而且不是通过类的实例去访问, 编译器会报错, 因为这些非静态变量还没有被创建呢, 并且它们没有与实例相关联。

5. Java 支持哪些数据类型？什么是 Autoboxing 和 Unboxing？

Java 语言支持的 8 个基本数据类型如下：

byte

short

int

long

float

double

boolean

char

Autoboxing 是指在基本数据类型和对应的包装(wrapper)类之间 Java 编译器所做的自动转换。例如，编译器将 int 转换为 Integer，将 double 转换为 Double，等等。逆向的转换称为 unboxing。

6. 在 Java 中什么是方法的 Override(覆盖) 和 Overload(重载)?

Java 中方法的 overload 发生的条件是，同一个类里，有两个或以上的方法名称完全相同，但参数列表不同。另一方面，方法的 override 是指，子类重定义了父类里的同一个方法。Override 的方法必须方法名、参数列表和返回类型都完全相同。Override 的方法不会限制原方法的访问权限。

7. Java 中构造函数、构造函数重载的概念和拷贝构造函数

当类的对象被创建的时候，调用它的构造函数。每个类都有一个构造函数。如果程序员没有为类编写构造函数，Java 编译器自动为类创建一个缺省的构造函数。

构造函数重载和 Java 中函数重载类似，可以为同一个类创建不同的构造函数，每个构造函数必须拥有唯一的参数列表。

Java 与 C++ 不同，它不支持拷贝构造函数，但是区别仅仅是，如果你没有编写类的拷贝构造函数，Java 不会自动创建它。

8. Java 支持多继承吗？

Java 不支持多继承，每个类只允许继承一个类，但是可以实现多个接口。

9. 接口和抽象类有什么不同？

Java 同时提供和支持抽象类和接口，它们的实现有一些共同的特点，也有如下不同：

接口中所有的方法默认都是抽象的，而抽象类可以同时包含抽象和非抽象的方法。

一个类可以实现多个接口，但它只能继承一个抽象类。

一个类要实现某个接口，必须实现这个接口声明的所有方法。而一个类不需要实现抽象父类中声明的所有方法，不过，这时候这个类也必须声明为抽象类。

抽象类可以实现接口，而且不需要实现接口中的方法。

接口中声明的变量默认是 final 的，而抽象类可以包含非 final 的变量。

接口中的成员默认是 public 的，而抽象类的成员可以是 private，protected，或 public 的。

接口是绝对抽象的，不可实例化，抽象类也不可以实例化，但可以在 main 方法中触发实例化（注：通过匿名类实现）。

也可以查阅 [Abstract class and Interface differences for JDK 8](#)。

10. 传引用和传值

当对象通过传值调用时，传递的是这个对象的一个拷贝。因此，即使函数修改这个对象，也不会影响原对象的值。

当对象通过传引用调用时，对象本身没有被传递，而传递的是对象的一个引用。因此，外部函数对这个对象的修改，也会反映到任何出现这个对象的地方。

11. 简述 Java 的基本历史

java 起源于 SUN 公司的一个 GREEN 的项目，其原先目的是：为家用消费电子产品发送一个信息的分布式代码系统，通过发送信息控制电视机、冰箱等

12. 简单写出 Java 特点，写出 5 个以上，越多越好

简单的、面向对象的、分布式的、安全的、稳定的、与平台无关的、可解释的、多线程的、动态的语言。

13. 什么是 Java?

JAVA：一种编程语言

一种开发环境

一种应用环境

一种部署环境

14. 请写出 Java 的版本分类，以及每种版本的应用方向

三种版本：

JME：是面向内存有限的移动终端，为消费性产品设计的；

JSE：是 SUN 公司针对桌面开发和低端商务计算方案开发的版本。为笔记本电脑、PC 机设计的；

JEE：是一种利用 JAVA 2 平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构。为企业级的、服务器端的高端应用而设计的；

15. 描述 Java 技术的主要特性

java 虚拟机

垃圾回收

代码安全性

16. 描述 Java 虚拟机的功能

(1) 通过 ClassLoader 寻找和装载 class 文件

(2) 解释字节码成为指令并执行，提供 class 文件的运行环境

- (3) 进行运行期间垃圾回收
- (4) 提供与硬件交互的平台

17. 简述 Java 代码安全性

- (1) 编译时的错误检验
- (2) 类装载检查, 把本机上的类和网络资源类相分离, 在调入类的时候进行检查, 因而可以限制任何“特洛伊木马”的应用
- (3) 字节码校验
- (4) 沙箱机制, 限定访问权限

18. 描述 Java 垃圾回收机制

Java 编程语言解除了程序员释放内存的责任。它可提供一种系统级线程以跟踪每一次内存的分配情况。在 Java 虚拟机的空闲周期, 垃圾收集线程检查并释放那些可被释放的内存。垃圾收集在 Java 技术程序的生命周期中自动进行, 它解除了释放内存的要求, 这样能够有效避免内存漏洞和内存泄露(内存泄露就是程序运行期间, 所占用的内存一直往上涨, 很容易造成系统资源耗尽而降低性能或崩溃)。

垃圾回收有以下特点:

- (1) 垃圾回收是一个自动的系统行为, 程序员不能控制垃圾回收的功能和行为。
- (2) 有一些跟垃圾回收相关的方法, 比如: `System.gc()`, 调用这些方法, 仅仅是在通知垃圾回收程序, 至于垃圾回收程序运不运行, 什么时候运行, 都是无法控制的。
- (3) 程序员可以通过设置对象为 `null`, 来标示某个对象不再被需要了, 这只是表示这个对象可以被回收了, 并不是马上被回收。

19. 简述 Java 代码的运行过程

- (1) 加载代码 通过类装载器装载 CLASS 文件
- (2) 校验代码 由字节码校验
- (3) 执行代码 由解释器将字节码转换成机器码

20. 简述如何搭建 Java 开发环境

首先下载安装 JDK 然后配置环境

(1) 配置 PATH ,

(2) 配置 CLASSPATH

(3) 配置 JAVA_HOME

21. 简述 classpath , path , JAVA_HOME 的意义和配置方法

path 操作系统运行环境的路径

classpath JAVA 运行应用程序时所需要的类包的路径

JAVA_HOME 供需要运行 JAVA 的程序使用

22. 请描述：一个完整的 Java 文件的代码分几个部分，每部分都起什么作用，大致的写法

package 当前类所在位置

import 当前类运行时所需要的包或类

public class 类名 {

属性定义;

方法定义:

构造方法定义;

public static void main(String args []) 例外{ 程序入口

语句;

}

}

//注释

23. Java 代码的注释有几种？分别举例说明

(1) // 单行注解

(2) /* */ 块注释

(3) /**

* 文档注释

*/

24. 什么是 Java 代码的语句，什么是 Java 代码的块，举例说明

1.

语句 是一行以分号终止的代码,例：int a;

块 是以{ }为边界的一些语句的集合 例：public void tt(){}


25. 什么是标识符？

标识符：是赋予变量、类、方法的名称。

26. 标识符定义的规则？

首字母只能以字母、下划线、\$开头,其后可以跟字母、下划线、\$和数字

首字母小写中间用大写字母隔开

标识符具有一定的含义

27. 什么是关键字？

关键字就是编程语言与机器语言的编码约定

28. true、false、null、sizeof、goto、const 那些是 Java 关键字

true、false、null 为 JAVA 的关键字

29. Java 的基本数据类型有几种？分别是？

short int long boolean float double char byte

30. 请写出每种基本类型的默认值？取值范围？分别给出一个定义的例子

默认值 取值范围 示例

字节型：0 -2⁷—2⁷-1 byte b=10;

字符型：'\u0000' 0—2¹⁶-1 char c='c';

short：0 -2¹⁵—2¹⁵-1 short s=10;

int：0 -2³¹—2³¹-1 int i=10;

long：0 -2⁶³—2⁶³-1 long o=10L;

float：0.0f -2³¹—2³¹-1 float f=10.0F

double：0.0d -2⁶³—2⁶³-1 double d=10.0;

boolean: false true \false boolean flag=true;

31. 在基本类型中，如果不明确指定，整数型的默认是什么类型？带小数的默认是什么类型？

整数类型 默认为 int

带小数的默认为 double

32. 如何定义 float 类型和 long 型

float f = 1.2f

long l = 1.2L

33. 什么是变量？

变量是用来引用一块内存区域或一个值，用标识符来表示，可以通过操作变量来操作变量所对应的内存区域或值块的值。

34. 变量的定义规则？

以字母、\$、下划线开头，其后可以跟字母、下划线、\$和数字；

首字母小写，中间单词用大写字母隔开；

名称具有一定含义；

35. 请写出 Java 编码约定中对下列部分的要求：类、属性、方法、包、文件名、变量、常量、控制结构、语句行、注释

类：一个类文件中类名要和文件名相同，类名一定要以大写字母开头，单词之间用大写字母分隔

属性：属性名首字母小写，中间用大写字母隔开。

方法：方法名首字母小写，中间用大写字母隔开。

包：引用包必须写在类文件的开头，有且只能有一个包，全部用小写字母。

控制结构：当语句是控制结构的一部分时，即使是单个语句，也应使用大括号将语句封闭。

语句行：每行只写一个语句，并使用四个缩进的空格，使代码更易读，注意用分号结束。

注释：用注释来说明那些不明显代码的段落。

常量：常量名一般用大写字母，单词之间用下划线分隔，一旦赋值不允许修改。

36. 什么是 Javadoc?

按照一定格式生成程序的文档的工具。

37. 什么是引用类型？

用户定义类型, 它可引用类和数组。

除去基本数据类型的其他类型都是引用数据类型。

38. 什么是按值传递？什么是按引用传递？

按值传递：就是将该值的副本传过去（基本数据类型+String 类型的传递，就是按值传递）

按引用传递：就是将值的内存地址传过去（除基本数据类型+String 以外类型的传递，就是引用传递）

39. 如何创建一个新对象？如何使用对象中的属性和方法？

使用 new 关键字来创建一个新的对象；

通过对象的实例用"."(点)来调用对象中的属性和方法；

静态的方法和属性，也可以直接用类名"."（点）来调用；

40. 简述 new 关键字所做的工作

new 创建一个对象，并为对象分配一块内存。

41. 简述"="和"=="的功能和区别

"=" 赋值,

"=="当比较基本数据类型的时候，判断前后两个值是否相等；

当比较引用数据类型的时候，判断==前后两个值的内存地址是否相等；

区别：

= ：为赋值表达式

= = : 为逻辑表达式

42. 什么是实例变量？什么是局部变量？什么是类变量？什么是 final 变量？

实例变量： 类中定义的变量，即类成员变量，如果没有初始化，会有默认值；

局部变量： 在方法中定义的变量，必须初始化；

类变量： 用 static 修饰的属性；

final 变量： 用 final 修饰的变量，

43. 简述上述各种变量的定义规则和使用规则？

实例变量： 不需要 static 关键字来声明，需要类的实例（对象）调用（用"."）；

类变量： 用 static 修饰，可以用类名调用，也可以用类的实例调用；

局部变量： 在方法内任意定义变量即为局部变量；

final 变量： 一旦赋值，不可以再修改的变量；

final 属性只允许赋值一次，且只能通过构造方法赋值，定义后也就是一个常量；

final 修饰的变量，只允许赋值一次

44. a++和++a 的区别？

a++ : 先使用，后加 1

++a : 先加 1，后使用

45. 请描述 instanceof、?:、&、&&各自的功能

instanceof : 用来判断某个实例变量是否属于某种类的类型。

?: 三目运算符 :

表达式 1?表达式 2:表达式 3

如果表达式 1 为 true , 执行表达式 2 , 否则执行表达式 3

& : 位运算 : 按位与 |

&& : 逻辑运算 : 逻辑与

46. 请描述>>、<<、>>>的功能

10>>4 : 算术或符号右移位运算符

<< : 算术或符号左移位运算符

>>> : 逻辑或无符号右移位运算符

47. 请描述什么是强制类型转换? 什么是自动类型转换? 什么是向上造型? 并分别举例说明

强制类型转换 : 在一个类型前面加() , 来强制转换

```
long l = 9L;
```

```
int i = (int)l;
```

自动类型转换 :

```
int i = 5;
```

```
String str = ""+i;
```

向上造型 : 把范围小的造型为范围大的类型 :

```
int i = 2;
```

```
long l = i;
```

48. 请写出完整的 if 条件控制语句的结构

```
if(布尔表达式){  
    语句  
}else{  
    语句  
}
```

49. 请写出完整的 switch 语句的结构

```
switch(字符){  
    case 字符: 语句  
        break;  
    case 字符: 语句  
        break;  
    default:语句  
}
```

50. 请写出完整的 for 语句的结构

```
for(初始语句;条件语句;步长){  
}
```

51. 请写出完整的 while 语句的结构

```
while(boolean 语句){
```

```
}
```

52. 请写出完整的 do while 语句的结构

```
do{  
  
}while(boolean 语句);
```

53. 请描述 break 和 continue 的功能和用法

break：终止（跳出）最近的循环

continue：跳出本次循环，执行下一次循环

54. 定义一个一维的 int 数组，先创建它，并初始化它，给它赋值，然后输出其中的一个值

```
public class Arr{  
    public static void main(String args[]){  
        int a[] = new int[5];  
        a={1,2,3,4,5};//错误，只能在初始化时这样做  
        a[0]=1;  
        a[1]=2;  
        System.out.println(a[0]);  
    }  
}
```

55. 定义一个一维的 A 类型数组，直接定义并赋值，然后输出其中的一个值

```
public class A{  
    public static int i;  
    public static void main(String args[]){  
        A aa = new A( );  
        A bb = new A( );  
        A a[] = {aa,bb};  
        a[0].i=2;  
        System.out.println(a[0]);  
    }  
}
```

56. 把上面的数组改成 2 维的数组

```
public class A{  
    public static int i;  
    public static void main(String args[]){  
        A a[ ][ ] = new A[5][5];  
        a[0][0].i=2;  
        System.out.println(a[0][0]);  
    }  
}
```

57. 举例说明数组拷贝方法的使用：arraycopy 方法

```
public class A{  
    public static void main(String args[]){  
        int a[] = new int[5];  
    }  
}
```

```
int b[] = new int[5];  
System.arraycopy(a[5],0,b[5],0,a.length)  
System.out.println(b[0][0]);  
}  
}
```

58. 什么是构造和析构方法？功能是？

构造方法：每个类至少有一个构造方法，类初始化时调用的方法

1. 方法名和类名相同

2. 无返回值类型

格式：访问权限 类名(参数列表) {};

1. 自己定义构造后，就没有默认的构造方法

2. 无构造方法时，默认为空参的构造方法(无参数,方法体为空)

析构方法：finalize

类销毁时，自动调用方法

当对象在内存中被删除时,自动调用该方法

在此方法中可以写回收对象内部的动态空间的代码

59. 简述 Java 的访问修饰符类型？分别有什么功能？

public 公开的，任何一个类都有可以访问

protected 同包同类,只可被它的子类访问

default 只能被同一个包中类访问

private 私有的，只能在本类中被访问

60. 分别说明：在类上、在属性上、在方法上能使用哪些访问修饰符

在类上：public default final

在方法上：访问权限：public protected private default

方法修饰符：static

final

返回类型：void

在属性上：public protected private default

static

final

61. 简述什么是包？包的功能

包：对所定义的多组 JAVA 类进行分组，将多个功能相关的类定义到一个包(文件)中。

功能：1.易于查找和使用适当的类

2.包不止包含类和接口,还可以包含其它包,形成层次的包空间

3.有助于避免命名冲突

62. 请写出 5 个以上的 JDK 中的包，以及他们的基本功能

java.awt：包含构成抽象窗口工具集的多个类，用来构建和管理应用程序的图形用户界面

java.lang：提供 java 编成语言的程序设计的基础类

java.io：包含提供多种输出输入功能的类，

java.net：包含执行与网络有关的类，如 URL，SCOKET，SEVERSOCKET，

java.applet：包含 java 小应用程序的类

java.util：包含一些实用性的类

63. 什么是包装类？Java 中都有哪些包装类

在 JDK 中针对各种基本类型分别定义相应的引用类型 —— 称为封装类

Boolean Byte Short Integer Long Float Double Character

北京动力节点