

# 《JAVA》冲刺题库（一）

——【大学生求职考试网/微信平台】教研团队——

第1组

一、选择题

**解释执行：**在执行程序时，先将程序转化成中间码，再将中间码一行行的解释成机器码进行执行。这个运行过程是解释一行，执行一行。（如Java字节码通过JVM解释成机器码）（Java, Python, Ruby）

**编译执行：**所有语句都编译成了机器语言，并保存成可执行的机器码，执行时直接执行机器语言，不需再进行解释/编译。（C, C++）

1. 下列选项中，不属于 Java 语言特点的一项是（ C ）。

- (A) 分布式 (B) 安全性  
(C) 编译执行 (D) 面向对象

【解析】Java 程序采用解释执行的方法。在系统编译运行 Java 程序时，Java 编译器将 Java 程序转化为字节码，在运行时，解释器将编译得到的字节码进行解释执行。

2. 在 Java 语言中，（ C ）是最基本的元素？

- (A) 方法 (B) 包  
(C) 对象 (D) 接口

【解析】构成 Java 程序的基本元素类（抽象的对象）。

3. 编译一个定义了 3 个类和 10 个方法的 Java 源文件后，会产生（ D ）个字节码文件？扩展名是（ D ）？

- (A) 13 个字节码文件，扩展名为 .class  
(B) 1 个字节码文件，扩展名为 .class  
(C) 3 个字节码文件，扩展名为 .java  
(D) 3 个字节码文件，扩展名为 .class

【解析】源文件中的每一个类编译后都会生成一个字节码文件，字节码文件的扩展名是 .class。

4. 在创建 Applet 应用程序时，需要用户考虑问题是（ B ）。

- (A) 窗口如何创建 (B) 绘制的图形在窗口中的位置  
(C) 程序的框架 (D) 事件处理

【解析】创建 Applet 程序时必须继承系统类 Applet，而 Applet 类中已经包含了如何创建窗口以及事件处理等内容，这类程序的框架也都是固定的，而绘制图形在窗口中的位置则需要由用户确定。

5. Java 语言属于（ B ）种语言？

- (A) 面向机器的语言 (B) 面向对象的语言  
(C) 面向过程的语言 (D) 面向操作系统的语言

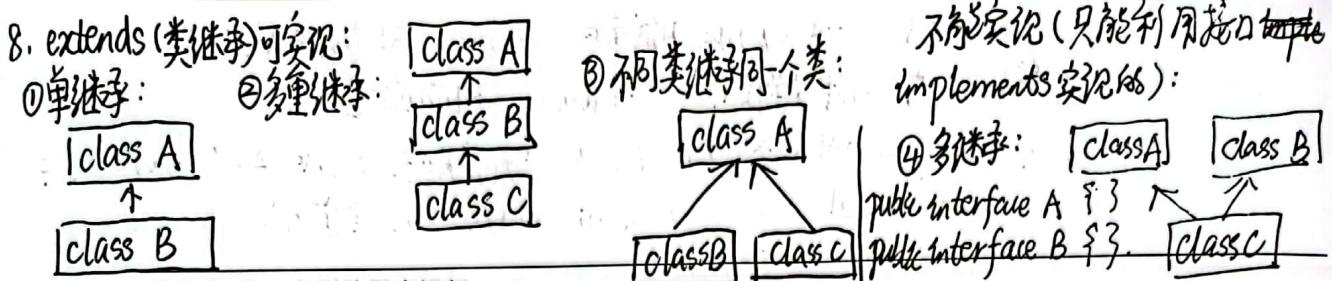
【解析】Java 语言是一种纯面向对象的语言。

6. 下列关于 Application 和 Applet 程序的说法中不正确的一项是（ B ）。

- (A) Application 使用解释器 java.exe  
(B) Application 不使用独立的解释器

① Application: Java 桌面应用程序，不能用 Jsp 加载。Application 是完整程序，可独立运行。Java Application 必含有一个且只有一个 main 方法，程序执行时，首先找 main 方法，以此为入口开始运行。Application 使用独立的解释器运行。

② Applet: Java 小应用程序，是一种在 Web 环境下，运行于客户端的 Java 程序组件，其功能比较简单（例如仅用于展示一个舞动的 Logo），Applet 必须运行于某个特定的容器，如浏览器、插件。Applet 不是通过 main 方法来运行的，因此 Applet 不能独立运行。Applet 的解释器不是独立的软件，而是嵌在浏览器中作为浏览器软件的一部分。



(C) Applet 在浏览器中运行

(D) Applet 必须继承 Java 的 Applet 类

【解析】Application 程序包含 main() 方法，它是一种 独立执行 的程序，因此必须使用 独立的解释器 解释执行。

7. 下列选项中，不属于 Java 核心包的一项是 (A)。

(A) javax.swing Java 设计 GUI 包 (B) java.io

(C) java.util (D) java.lang

【解析】凡是以 java 开头的包都是 Java 核心包，以 javax 开头的包则属于 Java 扩展包。

8. 下列描述中，不正确的是 (A)。

(A) 不支持多线程

(B) 一个 Java 源文件不允许有多个公共类

(C) Java 通过接口支持多重继承

(D) Java 程序分为 Application 和 Applet 两类

【解析】Java 是 支持多线程的语言。

9. 阅读下列代码，选出该代码段正确的文件名 (C)。

```
class A{
    void method1(){
        System.out.println("Method1 in class A");
    }
}

public class B{
    void method2(){
        System.out.println("Method2 in class B");
    }

    public static void main(String[] args){
        System.out.println("main() in class B");
    }
}
```

(A) A.java

(B) A.class

(C) B.java

(D) B.class

【解析】Java 源文件名必须和公共类的名字完全一样，源文件的扩展名为 .java。

10. 编译下面源程序会得到哪些文件 (D)？

```
class A1{
}

class A2{
}
```



11. -d: 指定存放生成的类文件的位置 -g: 生成所有调试信息 -g:none 不生成任何调试信息

-verbose: 输出有关编译器正在执行的操作的信息 -nowarn: 不生成任何警告

12. 标识符命名原则: ①由26个英文字母大小写、数字、下划线、美元符号组成(a-z, A-Z, 0-9, \_, \$)  
②不能以数字开头, 标识符不能是关键字 ③严格区分大小写 ④标识符可为任意长度

⑤包名多个单词组成时所有字母小写 (例: package com.csdn)

public class B {

public static void main(String[] args) {

} (例: HelloWorld)

字母大写 (例: lastAccessTime, getTime)

(A) 只有 B.class 文件

(B) 只有 A1.class 和 A2.class 文件

(C) 编译不成功

(D) A1.class、A2.class 和 B.class 文件

【解析】由于该程序包含3个类, 每个类编译后都会生成1个字节码文件, 因此编译后会生成以这3个类命名的字节码文件。

13. Java支持三种注释方法:

11. 编译Java程序时, 用于指定生成class文件位置的选项是 (A)。

A. -d B. -g C. -verbose D. -nowarn

12. 下列标识符 (名字) 命名原则中, 正确的是 (C)。

A. 类名的首字母小写 B. 接口名的首字母小写

C. 常量全部大写 D. 变量名和方法名的首字母大写

13. 下面哪种注释方法能够支持 javadoc 命令? (C)

A. // B. /\*...\*/ C. /\*\*...\*/ D. /\*\*...\*/

说明注释, 它以/\*\*开始, 以\*/结束。  
14. main()方法定义必须是: "public static void main(String[] args)" 字符串数组参数名, 一个类只有一个main()方法, 调用main方法时, 可以传参也可以不传参, 参数是一个String数组, 数组元素个数可以是0个。

14. 下面哪些选项是正确的 main 方法说明? (D)

A. void main()

B. private static void main(String args[])

C. public main(String args[])

D. public static void main(String args[])

## 二、填空题

1. Java程序的编译和执行模式包括2点, 是半编译和半解释。

2. Java语言支持TCP/IP协议, 从而使得Java程序在分布式环境中能够很方便地访问处于不同地点的对象。

3. 开发Java程序的一般步骤是: 源程序编辑、生成字节码和解释执行。

4. 每个Java Application程序可以包括许多方法, 但是必须有且只能有一个main()方法, 统一格式为 public static void main(String[] args), 它是程序执行的入口。

5. JVM把字节码程序与各种不同的操作系统和硬件分开, 使得Java程序独立于平台。

6. 在Java程序中, 能在WWW浏览器上运行的是Applet程序。



8. 当前目录: . 上层目录: .. 9. Java编译器 (javac) 不严格区分大小写, Java解释器 (java) 严格区分大小写. 另外: javac需要文件名 (文件名 = 文件名 + 扩展名), 要带后缀 .java, 而 java只需类名 (类名默认后缀 .class), 不需指明后缀. 命令例: javac HelloWorld.java; java HelloWorld.

7. Java 源程序文件和字节码文件的扩展名分别为 .java 和 .class .

8. 如果在 Java 程序中需要使用 java.util 包中的所有类, 则应该在程序开始处加上 import java.util.\* 语句.

### 三、判断题

1. Java 语言具有较好的安全性和可移植性及与平台无关等特性. (✓) *3. 意为: 调用 math() 方法时, 传不传参数都可以*
2. Java 语言的源程序不是编译型的, 而是编译解释型的. (✓)
3. Java Application 程序中, 必有一个主方法 main(), 该方法有没有参数都可以. (✓)
4. java.util.Scanner(System.in) 可以接收用户从键盘输入的简单数据. (✓)
5. Java 程序中不区分大小写字母. (×) *区分*
6. 机器不能直接识别字节码文件, 它要经过 JVM 中的解释器边解释边执行. (✓)
7. System 类中的 println() 方法分行显示信息, 而 print() 方法不分行显示信息. (✓)
8. 当前路径的标识是 ".". (✓)
9. java 命令不区分大小写, 而 javac 命令区分大小写. (×)
10. printf() 和 format() 方法使用指定格式字符串和参数, 将格式化字符串写入到 PrintStream 类型的输出流 (System.out 对象) 中. (✓)
11. 在运行字节码文件时, 使用 java 命令, 一定要给出字节码文件的扩展名 .class. (×) *见9题解释*

### 4. Java 基本数据类型:

5-6. 布尔型 {  
字符型 {

数值型 {

### 第2组 一、选择题

1. 下列 (D) 是合法的标识符?  
(A) 12class (B) void (C) -5 (D) \_blank

2. 下列 (B) 不是 Java 中的保留字?  
(A) if (B) sizeof (C) private (D) null

3. 下列 (C) 不是合法的标识符?  
(A) \$million (B) \$\_million  
(C) 2\$\_million (D) \$2\_million

4. 下列选项中, (B) 不属于 Java 语言的基本数据类型?  
(A) 整数型 (B) 数组 (C) 浮点型 (D) 字符型

5. 下列关于基本数据类型的说法中, 不正确的一项是 (D) .

- (A) boolean 类型变量的值只能取真或假
- (B) float 是带符号的 32 位浮点数

数据类型	默认值	大小	取值范围
boolean	false	1比特	true, false
char	'\u0000'	2字节	0~65535
byte	0	1字节	-128~127
short	0	2字节	-32768~32767
int	0	4字节	$-2^{31} \sim 2^{31}-1$
long	0L	8字节	$-2^{63} \sim 2^{63}-1$
float	0.0f	4字节	大约 $\pm 3.402 \times 10^{38}$ (6~7位有效数字)
double	0.0d	8字节	大约 $\pm 1.797693134862315 \times 10^{308}$ (15位有效数字)

### 8. Java 中有两种类型的转换:

自动(隐式)转换: 较小的类型转换为较大类型.

byte → short → char → int → long → float → double.

强制转换(手动转换): 反序.

(C) 应为 float e = (float) 0.0d;

(A) 应是印错了, 应为: int fe, a, b = 10;



(10) (C) 输出 Infinity (D) 转为 ASCII 码为 1 的字符.

(C) 注意正确写法有:  $\text{double } d = 1.1/0.0;$   $\text{float } d = 1/(\text{float})0.0;$   $\text{float } d = (\text{float})1.1/0;$

$\text{double } d = 1/(\text{float})0.0;$   $\text{double } d = 1/(\text{float})0;$   $\text{float } d = 1/(\text{float})0;$

错误写法有:  $\text{float } d = 1.1/0.0;$  (会提示无法将 double 转为 float, Java 默认小数是 double)

(C) double 是带符号的 64 位浮点数

(D) char 是 8 位 Unicode 字符

【解析】Java 中的字符采用的是 16 位的 Unicode 字符.

$\text{float } d = 1/0;$  (提示不能除以 0, 这里 1 和 0 都被当成整型看成了)  
关键: 0 被看作整型, 则语句错误.  
0 被看作浮点型, 则正确.

6. 下列关于基本数据类型的取值范围的描述中, 正确的一个是 (B).

(A) byte 类型的取值范围是 -128~128

-128~127

(B) boolean 类型的取值范围是真或假

(C) char 类型的取值范围是 0~65536

0~65535

(D) short 类型的取值范围是 -32767~32767

-32768~32767

【解析】根据每种类型占几个字节即可推算出其取值范围, 如 byte 类型占 1 个字节, 即共包含 256 个数值, 其取值范围范围应为 -128~127, 其他类型也是如此.

7. 下列关于 Java 语言简单数据类型的说法中, 正确的一项是 (A).

(A) 以 0 开头的整数代表 8 进制整型常量.

(B) 以 0x 或 0X 开头的整数代表 16 进制整型常量

(C) boolean 类型的数据作为类成员变量的时候, 相同默认的初始值为 true

(D) double 型数据占计算机存储的 32 位

8. 下列 Java 语句中, 不正确的一项是 (C).

(A)  $\$e, a, b = 10;$

(B)  $\text{char } c, d = 'a';$

(C)  $\text{float } e = 0.0d;$

(D)  $\text{double } c = 0.0f;$

【解析】不能将 double 类型的常量赋值给 float 类型的变量.

9. 在编写 Java 程序时, 如果不为类的成员变量定义初始值, Java 会给出它们的默认值, 下列说法中不正确的一个是 (D).

(A) byte 的默认值是 0

(B) boolean 的默认值是 false

(C) char 类型的默认值是 '\0'

(D) long 类型的默认值是 0.0L

10. 下列语句中不正确的一个是 (B).

(A)  $\text{float } f = 1.1f;$

(B)  $\text{byte } b = 128;$

(C)  $\text{double } d = 1.1/0.0;$

(D)  $\text{char } c = (\text{char})1.1f;$

【解析】byte 类型变量的取值范围是 -128~127.

11. 下列表达式  $1+2+ "aa"+3$  的值是 (B).

(A) "12aa3"

(B) "3aa3"

(C) "12aa"

(D) "aa3"

【解析】整数和整数相加得到两个整数的和, 而整数和字符串相加得到的是字符串.

12. 已知  $y=2, z=3, n=4$ , 则经过  $n=n+ -y*z/n$  运算后 n 的值为 (A).

(A) 3

(B) -1

(C) -1

(D) -3

【解析】根据运算符的优先级, 该表达式相当于  $n=n+ ((-y)*z)/n$ .

13. 已知  $a=2, b=3$ , 则表达式  $a\%b*4\%b$  的值为 (A).

(A) 2

(B) 1

(C) -1

(D) -2

【解析】根据运算符的优先级, 该表达式相当于  $((a\%b)*4)\%b$ .

(13)  $2\%3*4\%3 = 2*4\%3 = 8\%3 = 2$ .

(11) 另一更有迷惑性的例子:

$1+2+ "aa"+3+5$  输出: 3aa35.

$1+2+7+ "aa"+3+5+ "b"+7 \rightarrow 10aa35b$

$1+2+ "b"+ "aa"+3+5 \rightarrow 101aa35$

char 上转为 int \*b 的 ascii 码为 98.

int 上转为 string.

(12)  $-2*3/4 = -6/4 = -1$  (注意都是 int)

$4+(-2)*3/4 = 4-2*3/4 = 4-1=3$

$2*3/4 = 6/4 = 1$

(15) && (短路与), & (与), || (短路或), | (或)

&& 只要第一个条件不满足, 后面条件就不再判断. 而 & 要对所有条件都进行判断. & 可用作位运算符. 当 & 两边的表达式不是 Boolean 类型时, & 表示按位操作.

|| 同理.

0 & true 这种表达式是错的.

转 string: int 上转为 string.

Java 中 & 两边不能是一边 boolean, 一边 int.

(17) 注意下面两语句结果:

$\text{System.out.println}("a"+"b"); \rightarrow ab$

$\text{System.out.println}('a'+'b'); \rightarrow 195$

注意下面写法是对的:

$\text{char } c1 = 66;$

$\text{System.out.println}(c1) \rightarrow B$

另外:  $\text{char } c1 = 66; \text{System.out.print}(c1);$

$\text{int } i = 2;$

$i = i + c1;$

$\downarrow$   
68.



⑧ 注意 Java 的转义符: \f 换页 \n 换行 \r 回车 \t 水平制表 \v 垂直制表  
A, \b 退格 \\ 代表一个反斜线字符 \ ' 一个单引号 \" 一个双引号.

另外: \\\\ 代表 \\, 以下写法是错误的 String a = "1a", 会提示元 1a ② String a = "\"8\"", 提示 需转义.

另外: a = "'8'"; 是可以的, 就代表 '8'.

String 14. 已知 x=2, y=3, z=4, 则经过 z- --y - x-- 运算后, z 的值为 ( D ).

(A) 1 (B) 2 (C) 3 (D) 4

【解析】在表达式运算过程中, --y 的值变为 2, x-- 的值还是 2, 等号右侧运算后的值为 0, 因此 z 的值没有变化.

15. 表达式 (12==0) && (1/0 < 1) 的值为 ( B ).

(A) true (B) false (C) 0 (D) 运行时抛出异常

【解析】由于 (12==0) 的值为 false, 因此整个表达式发生短路运算, 即 (1/0 < 1) 就没有参与运算, 整个表达式的值为 false.

16. 设有类型定义 short i=32; long j=64; 下面赋值语句中不正确的一个是 ( B ).

(A) j=i; (B) i=j; (C) i=(short)j; (D) j=(long)i;

【解析】long 类型的数据不能自动转变为 short 类型, 因此不能将 long 类型的变量直接赋值给 short 类型.

17. 现有 1 个 char 类型的变量 c1=66 和 1 个整型变量 i=2, 当执行 c1=c1+(char)i; 语句后, c1 的值为 ( D ).

(A) 'd' (B) 'D' (C) 68 (D) 语句在编译时出错

【解析】两个字符型的数据相加, 得到的是一个整数, 而如果把整数再赋值给一个字符型变量则会在编译时出错.

18. 下列说法中, 正确的一项是 ( D ).

(A) 字符串 "\\abcd" 的长度为 5 18. (C) ① 小数不能声明为长整型 L

(B) False 是 Java 的保留字 不是

(C) 123.45L 代表单精度浮点型

(D) False 是合法的 Java 标识符

【解析】Java 语言对字符的大小写是敏感的, False 不是 false, 因此 False 是合法的 Java 标识符.

19. 以下的变量定义语句中, 合法的是 ( D ).

(A) float \_\*5 = 123.456F; (B) byte \$\_b1 = 12345;

(C) int \_long\_ = 123456L;

(D) double d = Double.MAX\_VALUE;

【解析】(A) 中 \*\_5 不是合法的标识符, (B) 中 12345 超出 byte 范围, (C) 中不能将 long 类型的常量赋值给 int 型的变量.

20. 下列关于运算符优先级的说法中, 不正确的一个是 ( C ).

(A) 运算符按照优先级顺序进行运算

(B) 同一优先级的运算符在表达式中都是按照从左到右的顺序进行运算的

(C) 同一优先级的运算符在表达式中都是按照从右到左的顺序进行运算的

(D) 括号可以改变运算的优先次序

【解析】同一优先级的运算符在表达式中都是按照从左到右的顺序进行运算的.

二、填空题

1



⑧ final 定义的变量初次赋值后就不能更改了。  
需转义。另一个 final 的例：  
final long i;  
i = 100;  
System.out.println(i); → 100

⑨  $5 + 11 + 6 * 11 = 16 + 66 = 82$   
a 不变 5 a 变 6 注意 a 变的时机，第二次用时变！

1. 变量是 Java 程序的基本存储单元之一，变量的主要类型包括 2 大类：字符型和数值型。布尔型。

2. Java 语言的整数类型变量和常量一样，各自都包括 4 种类型的数据，它们分别是 byte、int、short 和 long。整型。

boolean 类型数据不可以做类型转换。

4. 在 Java 语言的基本数据类型中，占存储空间最少的类型是 boolean，该类型占用的存储空间为 1 位。

5. Java 语言中的保留字具有特殊意义和作用，不能作为普通标识符使用。

6. 在 Java 语言中，浮点类型数据属于实型数据，可以分为单精度和双精度两种。

7. char 类型的数据可以表示的字符数共为 65536。0~65535。long。

8. 定义初始值为 10 的 8 次方的常整型变量 iLong 的语句是 final iLong = 100000000L。

9. Java 语言中的数据类型转换包括自动转换和强制转换两种。(隐式转换、手动转换)

10. Java 中的字符采用的是 16 位的 Unicode 编码。(即 char)。

11. 数据类型中存储空间均为 64 位的两种数据类型是 long 和 double。

12. 表达式  $9 * 4 / -5 \% 5$  的值为 -2。(十进制表示) 13. 表达式  $5 \& 2$  的值为 0。(十进制表示) ⑫  $36 / -5 \% 5 = -7 \% 5 = -2$  ⑬  $4 \& 010$  (按位与) ⑭  $101010!0000_{(2)} = 32 + 128 + 512 = 672$

14. 表达式  $42 \ll 4$  的值为 672。(十进制表示) 15. 表达式  $11010011 \gg 3$  的值为 11010。(二进制表示) ⑮ 11010011 ⑯ 11011 (按位或)

16. 表达式  $7 | 3$  的值为 7。(十进制表示) 17. 表达式  $10^2$  的值为 8。(十进制表示) ⑰  $1000$  (按位异或) ⑱  $!(10 < 50) || !(20 \leq 20) = false || false = false$

18. Java 语言中的逻辑与(&&)和逻辑或(||)运算采用短路方式进行运算。

若 a、b 为 int 型变量，并且已分别赋值为 5 和 10，则表达式  $(a++) + (++b) + a * b$  的值为 82。

20. 假设 i=10, j=20, k=30，则表达式  $!(i < j + k) || !(i + 10 \leq j)$  的值为 false。

三、判断题 (参考在电脑上写的文档) ③ 实例变量(成员变量)、静态变量(类变量)可不用初始化。

1. Java 语言使用的是 Unicode 字符集，每个字符在内存中占 2 位。( × )

2. Java 语言中不同数据类型的长度是固定的，不随机器硬件不同而改变。( √ )

3. 所有的变量在使用前都必须进行初始化。( × ) ② 最后一行空白处

4. 已知 byte i = (byte)127; i = i + 1; 这两个语句能被成功编译。( × )

5. String str = "abcdefghi"; char chr = str.charAt(9); ( × ) 下标应为 0~8，越界了

6. char[] chrArray = { 'a', 'b', 'c', 'd', 'e', 'f', 'g' }; char chr = chrArray[6]; ( √ ) 下标范围 0~6

7. int i, j; boolean booleanValue = (i == j); ( × ) 默认该语句写在 main 方法中，则 i, j 为局部变量。

8. int intArray[] = {0, 2, 4, 6, 8}; int length = intArray.length; ( × ) 没有括号

9. String str = "abcedf"; int length = str.length; ( × ) 应为: int length = str.length();

10. short shortValue = 220; byte byteValue = shortValue; ( × ) 向下类型转换不能自动发生 需强转。

⑪ int[] intArray[60]; ( × ) 声明数组时不能带数字，应为 int[] intArray[]; 不能自动转

12. char[] str = "abcdefgh"; ( × ) 应为: char[] str = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' }; 换

13. 说明或声明数组时不分配内存大小，创建数组时分配内存大小。( √ )

14. 强制类型转换运算符的功能是将一个表达式的类型转换为所指定的类型。( √ )

↓  
指强制类型转换。