

信息技术类知识点-JAVA 基础知识点总结

Java 程序设计概述

1. 开发 Java 语言的公司？

Sun （Sun Microsystems）太阳微电子有限公司开发，后被 Oracle 公司收购

2. Java 的 3 个版本？

J2SE(Java2 Standard Edition) 标准版

J2EE(Java 2 Platform,Enterprise Edition) 企业版

J2ME(Java 2 Micro Edition) 微小版

在 5.0 以后新版本中称作：JavaSE JavaEE JavaME

3. Java 语言的几个主要特点？

平台无关性：能运行于不同的平台上

安全性：去掉了指针操作，内存由操作系统分配

面向对象：利用类使程序更加利于维护

分布式：可以使用网络文件和使用本机文件一样容易

健壮性：检查程序编译和运行的错误

4. Java 的三大核心机制

虚拟机机制：利用虚拟机解释字节码执行 java 程序实现跨平台

垃圾收集机制：自动内存回收

代码安全性机制：加载代码后校验代码后再执行代码

Java 语言的开发环境

5. JDK 是什么？

JDK(Java Development Kit)Java 开发工具包。包括了 JRE，是开发 Java 的人员必备的开发运行 Java 程序的工具。

6. JRE 是什么？

JRE (Java Runtime Environment) Java 运行环境, 运行 JAVA 程序所必须的环境的集合, 包含 JVM 标准实现及 Java 核心类库。

7. JVM 是什么?

JVM (Java Virtual Machine) Java 虚拟机, 它是一个虚构出来的计算机, 是通过在实际的计算机上仿真模拟各种计算机功能来实现的。

8. 安装 JDK 的操作系统问题?

必须针对于不同的操作系统下载安装不同的 JDK, 例如 windows 下安装针对于 windows 的 JDK, linux 下安装针对于 linux 的 JDK。

9. 需要配置的相关环境变量?

path: 用于搜索外部命令, 例如编译程序的 javac 命令, 执行程序的 java 命令

classpath: 用于搜索类, 即 class 文件, 例如可以在不同的位置执行类文件

10. 配置环境变量的方式?

有两种: 临时配置和持久配置

临时配置: 在命令行下用 set 变量名=变量值

, 仅在当前命令行窗口有效

持久配置: 在我的电脑——属性中配置, 持久有效

11. 可以编写 Java 程序的常见开发工具?

记事本, EditPlus, ULTRAEDIT, Eclipse, Jbuilder, NetBeans

12. Java 文件的类型?

有两种:

扩展名为 java 的文件: 是 Java 源文件, 属于编译之前的纯文本文件, 存储 java 源代码。

扩展名为 class 的文件: 是 Java 类文件, 属于编译之后的二进制文件, 存储字节码。

13. Java 的编译运行原理?

首先编写 java 源文件, 然后编译成 class 类文件, 然后通过虚拟机解释成机器码到硬件执行。

14. Java 是编译型语言还是解释型的语言?

既不是完全编译型, 也不是完全解释型, 是半编译半解释的。

Java 程序的基本结构

15. Java 语言区分大小写吗?

区分, 不管是数据还是标识符或者关键字都区分大小写, 例如 public 不能写成 Public, “abc” 不能写成” ABC”, int i 不能写成 INT I 等等。

16. main 方法是作什么用处的?

这是程序的入口方法, 是由虚拟机自动调用的。

17. 语句必须用分号结束吗?

必须的, 但是代码块{}可以用分号结束, 也可以不用分号结束

18. Java 中有几种注释，各自的功能

有三种：

单行注释//：加在行首只能注释一行

多行注释/* */：注释多行，但不能嵌套使用

文档注释/** */：用于生成 HTML 形式的文档，需要配合 javadoc 命令实现

19. Java 语言的数据类型是强类型还是弱类型？

是强类型，意味着每一个变量都必须明确声明数据类型。

20. Java 语言中数据类型的划分？有什么区别？

分成基本类型和引用类型两大类。

基本类型：有 8 种，只能存储单一的值，不以对象的形式存在

引用类型：除了基本类型之外其他的都是引用类型，能够存储复杂数据类型，以对象的形式存在

21. 基本类型都有哪 8 种类型？

布尔类型：boolean 占 1 字节

字符类型：char 占 2 字节

整型：byte 占 1 字节 short 占 2 字节 int 占 4 字节 long 占 8 字节

浮点型：float 占 4 字节 double 占 8 字节

22. 布尔值可不可以用 0 或者非 0 数代替？

不可以，布尔值只能是 true 和 false，不能使用数字，这和 C 语言是不一样的。

23. 直接写个整数 10 默认是什么类型的？

整型的直接量例如 10，100 等默认的数据类型都是 int 型

24. 如何让一个整型的直接量变成 long 型的？

在直接量后面加大写 L 后者小写 l，例如 10L

25. 整型数据在程序中直接使用可以使用什么进制？

10 进制

，8 进制

，16 进制

，但不能使用二进制

26. 浮点运算在 Java 中有没有误差？

有误差，所以精密运算请使用 BigDecimal 类

27. 直接写一个 3.14 的直接量是什么数据类型的？

是 double 型，如果想表示是 float 的 3.14 必须写成 3.14F 或者 3.14f，double 型的 3.14 也可以写成 3.14D

28. char 型采取的是什么类型的字符集？

是 Unicode 全球统一编码，支持任何世界上官方的语言中的字符

29. 常见的转义字符有哪些？

转义序列

\b	退格
\t	制表
\n	换行
\r	回车
\"	双引号
\'	单引号
\\	反斜杠

1. 常见的 Unicode 字符对应的编码?

大写的 A 到 Z 是对应 65 到 90, 小写的 a 到 z 是对应 97 到 122, 字符 0 到 9 是对应 48 到 57

2. 空字符是不是用" (两个并列的单引号) 来表示?

不是, 空字符用 '\u0000' 来表示, 两个并列的单引号什么也不是。

3. 点, 分号, 花括号的各自的作用?

点的作用有两个: 一个是使用类或对象中的成员, 用对象.成员, 再一个就是包名和子包名之间的分隔符, 例如顶层包名.子包名

分号的作用是表示语句结束

花括号的作用是代表一个语句块

4. 语句和关键字之间可以有任意的空白吗?

可以的, 例如 `System.out.println(100);` 就是正确的语句, 但是关键字本身不能加空白或换行

5. 标识符是什么?

标识符就是指在程序中给包, 类, 接口, 方法, 变量起的名字

6. 标识符有什么起名规则?

1. 应以字母、下划线、美元符开头
2. 后跟字母、下划线、美元符或数字
3. 不能用关键字和保留字
4. 严格区分大小写
5. 名称长度没有限制

7. 中文的变量名可以吗?

是可以的, 例如 `int 数字 = 10;` 这个语句是正确的。因为 java 中采取 Unicode 字符集, 所以任何的 Unicode 中的字符都可以作为标识符的名称, 但考虑字符集的支持问题推荐还是使用英文字母命名。

8. 关键字和保留字有区别吗?

在 1.5 后没有区别, 都被称作关键字, 但是 1.5 之前认为关键字是在 Java 中有特殊用途的不允许编程人员擅自改作其他用途, 而保留字是在 Java 中没有特殊用途但不允许编程人员擅自改作其他用途, 按此说法 `goto` 和 `const` 应该是此意义上的保留字。

9. true, false, null 可以用作标识符命名吗?

不可以, 这 3 个虽然不是关键字, 但都是有特殊用途的值, 也不能作为标识符命名

10. 一条声明语句可以声明多个变量吗?

可以的, 例如以下声明都是正确的

```
int i;  
int i,j;  
int i = 10,j;  
int i,j = 10;  
int i = 10,j = 20;
```

1. 变量在没有赋值的情况下可以使用吗?

不可以, 变量没有赋值使用会报出变量未初始化的编译错误。

2. 常量有什么特点?

在声明变量的时候加上 final 修饰符即为常量, 表示这个变量只能被赋一次值。

3. 算数运算符有哪些?

基本算数运算符有 + - * / % ++ -

4. 单目运算符, 双目运算符, 三目运算符都是什么意思?

其实也有叫一元运算符, 二元操作符, 三元操作符的, 指的是有几个操作数形成一个完整表达式, 例如 ++ 就是一个单目运算符, 只需要一个操作数就可以形成表达式, 像 i++, 而 + 运算符就是双目运算符, 需要两个操作数形成表达式, 例如 5 + 3, 还有三目运算符, 例如 5 > 3 ? 0 : -1

5. ++ 和 - 放在前面和放在后面有什么区别?

如果是在一个表达式中, 那 ++ 放后面表示先运算表达式, 然后再进行自增运算, 如果在一个表达式中, ++ 放前面, 则表示先进行自增运算然后在运算表达式。- 同样道理。

例如:

```
int i = 10;
```

```
int j = i ++; // 先进行赋值运算然后再进行自增运算  
j 的结果是 10
```

```
int i = 10;
```

```
int j = ++ i ; // 先进行自增运算然后再进行赋值运算  
j 的结果是 11
```

1. -5 % 2 等于几, 5 % -2 等于几?

-5 % 2 等于 -1, 5 % -2 等于 -1, 运算得到是正数还是负数取决于第一个操作数的符号

2. 5 / 2 结果是不是 2.5?

不是, 是 2, 因为 5 和 2 都是 int 类型, 所以不可能得到 double 型的结果。

3. 比较运算符有哪些?

有 > < >= <= == != 六项基本比较运算

4. 任何数据类型都可以使用 $><=$ $<=>$ 运算符吗？

不是所有数据类型，例如 `true>>false` 这个表达式就是编译错误，布尔类型和引用类型的表达式只能用 `==` 和 `!=` 两个比较运算，例如 `"abc"<="xyz"` 这个表达式也是编译错误的

5. `'a'>'b'` 这个表达式编译正确吗？

编译正确，`'a'` 和 `'b'` 是字符类型，内部用 Unicode 编码表示的话 a 是 97，b 是 98，所以实际上比较的是 `97>98`，结果为 `false`

6. 逻辑运算符有哪些？

有 `&` 逻辑与，`|` 逻辑或，`!` 逻辑非，`^` 异或，`&&` 短路与，`||` 短路或

7. 逻辑与，逻辑或和短路与，短路或有什么区别？

逻辑与和短路与的区别是：

- 如果是逻辑与 `&`，不管第一个表达式是 `true` 还是 `false`，都会执行第二个表达式，

例如 `a>b & x<y`，即使 `a>b` 为 `false`，也会执行 `x>y`

- 如果是短路与 `&&`，如果第一个表达式为 `false`，将不会执行第二个表达式

例如 `a>b && x<y`，如果 `a>b` 为 `false`，就不会执行 `x>y`

以此类推，逻辑或和短路或的区别是：

- 如果是逻辑或 `|`，不管第一个表达式是 `true` 还是 `false`，都会执行第二个表达式
- 如果是短路或 `||`，如果第一个表达式为 `true`，将不会执行第二个表达式

1. 与或非三个逻辑运算符的优先级是怎么排列的？

“非”优先级最高，“与”其次，“或”优先级最低

2. 原码，反码，补码分别是什么意思？

- 原码：将数字转换为二进制，第一位为符号位，正数为 0，负数为 1

例如：12 的原码是 00001100 -12 的原码是 10001100

- 反码：正数的反码和原码相同，负数的反码将原码符号位之外的位逐位取反

例如：12 的反码是 00001100 -12 的反码是 11110011

- 补码：正数的补码和原码相同，负数的补码将反码加 1

例如：12 的补码是 00001100 -12 的补码是 11110100

最后强调一点，计算机存储数据用补码形式来存储。

1. 位运算符都有哪些？

`&` 与 `|` 或 `^` 异或 `~` 非 `>>` 右移 `<<` 左移 `>>>` 无符号右移

2. `1<<35` 怎么理解？

1 是 `int` 类型是 32 位，所以需把 `35%32=3`，所以事实上是和执行 `1<<3` 等价的

3. 数据类型转换需要遵循什么原则？

小类型向大类型转换，是自动转换的，或者说隐式转换的

大类型向小类型转换，是需要强制转换的，或者说显式转换

例如：

`byte b = 1;`

```
short s = b;//自动转换
b = (byte) s;//强制转换
```

4. `byte b = 10` 这个语句是正确的，为什么 10 是 `int` 型比 `byte` 大还不需要强制转换呢？

因为在 `java` 中做了特殊处理，其中规定 `byte`，`short`，`char` 类型可以直接赋予 `int` 类型的直接量，例如：

```
byte b = 10;
```

```
short s = 10;
```

```
char c = 10;
```

以上赋值语句都是正确的。

但是不能超出超出类型的范围，例如以下赋值就是错误的：

```
byte b = 128;
```

如果超出范围，必须得作强制转换，例如以下语句是正确的：

```
byte b = (byte) 128;
```

需要强调的是，必须是 `int` 类型的直接量，以下语句就是错误的：

```
byte b = 10L;
```

10L 不是 `int` 类型的直接量。

5. `byte b = (byte) 300`;强制转换后的结果是 44，是按照什么规律得出来的？

是先把 300 转换成二进制，然后根据 `byte` 类型的大小截取了后 8 个位

300 的补码是 00000000 00000000 00000001 00101100

b 的结果是：00000000-00000000-00000001 00101100 结果为 44

1. `byte b = (byte) true`;这个转换成立吗？

不成立，布尔值和数字是不兼容的，不能作任何转换。

2. `int i = (int) "100"`;这个转换成立吗？

不成立，引用类型和基本类型是不能直接强制转换的，需要用包装类的方法来转换。

3. `char` 型和 `short` 型哪个比较大？

不能说哪个大，哪个小，这两个类型 `char` 的取值是从 0~65535，而 `short` 类型的取值是-32768~32767，这两个类型之间的转换都需要强制转换。

4. 字符串是不是基本类型？

不是，`String` 是 `java.lang` 包中的一个类，是引用类型的，8 种基本类型中没有一个是 `String` 类型。

5. 空字符串用什么来表示？

用“”（两个并列的双引号）表示，代表 0 个长度的字符串对象。

6. `String s = ""`;和 `String s = null`;有什么区别？

第一个表示 `s` 被赋值，指向了一个字符串对象，第二个表示 `s` 没有指向任何对象实例，是真空

7. 两个字符串用什么运算符可以连成一个字符串？

用+运算符，例如“abc”+“xyz”最后的结果就是“abcxyz”

8. 数学运算符加法和字符串连接符都是+，怎么区分？

是通过运算符两边的操作数来判断的，如果两边的操作数都是数字型的基本类型，则进行数学运算，如果两边的操作数有一个是字符串，则把另一个操作数也转换为则进行字符串连接运算。

例如：

5 + 4 的结果是 9

“5” + 4 的结果是 54

9. 判断字符串内容相同用什么方法？

用 equals 方法判断内容相同，==只能判断引用是否相同

10. 提供用户输入使用哪个类

使用 java.util.Scanner 类

11. if 语句可以用 if(-1)形式判断吗？

不可以，数字不能代替布尔值

12. if 语句和循环语句的花括号什么时候可以省略？

当执行的语句只有一句的时候

13. switch 语句有哪些规则？

表达式 expr 的返回值必须是下述几种类型之一：int, byte, char, short 或枚举常量；

case 子句中的值必须是常量，且所有 case 子句中的值应是不同的，也必须是下述几种类型之一：int, byte, char, short 或枚举常量

default 子句是任选的；

break 语句用来在执行完一个 case 分支后使程序跳出 switch 语句块；

14. 对于退出循环 break 和 continue 有什么区别？

break 是退出整个循环

continue 是退出当前循环继续下一次循环

15. 声明数组创建数组的方法？

以下声明都是合法的：

```
int [] a;
```

```
int a[];
```

```
String[] a;
```

以下创建数组对象都是合法的：

```
int[] a = new int[3];
```

```
int[] a = {1,2,3};
```

```
int[] a = new int[] {1,2,3};
```

16. int[] x 和 String[] s 这两个数组对象，哪个是基本类型，哪个是引用类型？

两个都是数组对象，都是引用类型的

17. 数组元素下标最小是多少，最大是多少，超出范围会怎么样？

最小是 0，最大是数组元素个数减一，超出范围将会出现下标越界异常

18. 得到数组长度用 length 属性还是 length () 方法？

是 `length` 属性，得到字符串长度才有 `length` 方法

19. 声明 `int[][] a = new int[3][]`，`a` 是一个二维数组，`a[0]` 本身是什么类型的？
二维数组的元素都是一维数组，`a[0]` 是 `int[]` 类型的

20. 数组的大小一经创建可不可以改变大小了？
不可以，数组是定长的，一经创建就不能再改变大小了

21. 可以把 2 个 `int` 型，3 个 `double` 型，4 个 `String` 型的值放到一个数组中吗？
不可以，数组的元素类型必须是相同类型的。

22. `java` 中的多维数组必须是要形成矩阵排列吗？
不一定，也可以是不规则的。

23. 操作数组比较方便的实用类？
`java.util.Arrays` 类提供了很多关于数组的实用方法。

对象和类

1. OOP 是什么意思？
Object Oriented Programming(面向对象编程)

2. 类和对象的区别？
类：对象的蓝图，生成对象的模板，是对一类事物的描述，是抽象的概念上的定义
对象：对象是实际存在的该类事物的每个个体，因而也称为实例

3. 类之间的 3 种关系？
依赖关系（`uses-a`）
聚集关系（`has-a`）
继承关系（`is-a`）

4. 一个 `java` 源文件可以写多个类吗？
可以，但必须只能有一个公有类，并且文件名和公有类的类名相同。

5. 类中有哪些成员？
有属性（数据域），方法，构造器（构造方法）

6. 构造方法和普通方法的区别？
构造方法没有返回值，方法名必须和类名相同，需要用 `new` 语句调用构造方法。

7. 普通方法可不可以也类名作为方法名？
是可以的，因为调用普通方法的语法和调用构造器的语法不一样。

8. 一个构造器调用另外一个构造器需要什么语法？
用 `this()` 语法调用另外一个重载构造器，需要注意的是调用语句必须是构造器中的第一句

9. 什么时候虚拟机会自动添加一个默认构造器？

当类中没有显式定义任何构造器的时候，虚拟机会自动添加一个无参的构造器

10. 实例成员和静态成员的区别？

实例成员：是属于对象实例的，用对象.成员名称语法调用

静态成员：和对象实例无关，是属于类本身的，用类名.成员名称语法调用

11. 静态初始化块 static{} 的执行时机是什么时候？

在类被加载的时候，仅执行一次。

12. 静态方法可不可以直接调用同一个类中的非静态方法？

不可以，静态方法不能直接调用一个非静态成员，需要用对象调用。

13. 包是什么？

包实际上就是物理上的文件夹，可以把类放到不同的包中。

14. 把一个类放到包中，需要在类中加上什么语句？

需要加上包声明 package 语句，必须写在源文件的第一句。

15. 什么时候需要导入类？用什么语句？

当使用的类和当前类并不在同一个包中的时候，用 import 语句导入。

16. 默认包中的类可以导入吗？

不可以，不在默认包中的类无法使用默认包中的类。

17. java 中哪个包中的类是自动导入的？

java.lang 包

18. 什么叫做完整类名？

顶层包名.子包名.类名

19. 静态导入语句有什么作用？

在使用类中的静态成员的时候前面不需要加类名前缀。

20. 局部变量和全局变量的区别？

区别	局部变量	全局变量
声明位置	方法或块内部	方法外类内部
访问权限修饰符	没有访问权限	有访问权限
默认值	没有默认值	有默认值
使用形式	直接使用	对象调用
作用域	在方法或语句块内部	整个类
生命周期	进入方法或语句块创建退出方法或语句块销毁	随着对象的创建而创建随着对象的销毁而销毁

1. 赋值语句赋给目标的是值还是引用？

如果是基本类型则赋的是值本身，引用类型则赋的是引用地址

继承和多态

1. 类的继承是谁继承谁？

是子类继承了父类，子类也称为派生类，父类也称为超类或基类

2. 子类继承了父类的什么东西？

子类可以继承父类的数据域（属性），方法，但构造器不能继承

3. 私有成员能被子类继承吗？

能继承，但是由于访问权限的控制，在子类中不能直接使用父类的私有成员

4. Object 是什么类？

这个所有类的根类，任何类如果没有写显式的继承语句，默认都是直接继承 Object 类的

5. java 中是单继承还是多继承？

是单继承，一个子类只能有一个父类

6. 访问权限修饰符有几种，访问的范围又是多大？

有四种：

private: 私用

只能在被类中被访问

default: 默认

只能被同一个包中的类访问

protected: 受保护的

只能被同一个包中的类和不同包中的子类访问

public: 公有

可以被任何类访问

7. 覆盖方法的规则有哪些？

子类覆盖方法和父类被覆盖方法的方法返回类型，方法名称，参数列表必须相同

子类覆盖方法的访问权限必须大于等于父类的方法的访问权限

方法覆盖只能存在于子类和父类之间

子类覆盖方法不能比父类被覆盖方法抛出更多异常

8. 参数列表是否相同从哪几方面看？

从三方面看：参数个数，类型和顺序，需要注意的是参数的名字不相同是无所谓的

9. 覆盖方法的规则有哪些？

方法名相同

方法的参数类型,个数,顺序至少有一项不同

方法的返回类型和修饰符均不受限制

10. 可变长参数的声明格式是什么？实际上是什么类型的？

类型 ... 变量，实际上是一个一维数组

11. this 和 super 是作什么用处的？

this: 本身代表当前的对象引用，可以用 `this.成员名称` 调用当前对象的属性和方法，如果属性和局部变量重名，可以用 `this.属性名` 和局部变量区分开。也可以用 `this()` 语法调用本类重载构造器。

super: 本身是一个特殊关键字，可以用 `super.成员名称` 调用从父类继承的属性和方法，也可以用 `super()` 语法调用父类构造器。

12. `super` 代表子类中的父类对象引用吗？

不，这个错误的说法，`super` 不代表任何对象引用，仅仅是一个关键字

13. 什么时候子类构造器会自动调用父类无参构造器？

当子类构造器中没有任何显式构造器调用的时候，即既没有 `this()` 调用也没有 `super()` 调用的时候

14. 一个父类类型的引用变量可以引用一个子类的实例吗？

可以，这个引用变量是一个多态对象

15. 多态对象有什么特点？

多态对象只能使用父类的成员，但如果多态对象使用了被子类覆盖的方法，则会动态绑定使用子类的成员，称为虚拟方法调用

16. 引用类型之间的类型转换需要遵循哪些规则？

自动转换：从子类到父类的类型可以自动转换

强制转换：从父类到子类的类型要强制转换

无继承关系的类型转换是非法的

17. `instanceof` 操作符是作什么用处的？

可以判断一个对象的类型是否是某一个类的类型或子类，也可以判断这个对象的类型是否是一个接口的实现类

18. `toString` 方法什么时候会自动被调用？

在打印语句中传入引用类型的对象或者对象和字符串作连接时

19. 默认的 `equals` 方法实现的是比较内容还是比较引用地址？

在 `Object` 中是比较引用地址。

20. 覆盖 `equals` 方法和 `hashCode` 方法有什么规则？

必须保证当两个对象 `equals` 比较为 `true` 的时候，返回的 `hashCode` 相同，当比较为 `false` 的时候，返回的 `hashCode` 不同

21. `java` 中有几个包装类？

有 8 个包装类，正好对应 8 种基本数据类型。

2. 分别是：Boolean Byte Short Integer Long Character Float Double

3.

1. 装箱和拆箱分别是什么意思？

装箱指的是把基本类型包装成包装类的引用类型

拆箱指的是把包装类形式的引用类型转换为基本类型

例如：

```
Integer objInt = new Integer(10); //装箱
```

```
int i = objInt.intValue(); //拆箱
```

2. JDK1.4 和 1.5 的装箱和拆箱有什么区别?

自动拆装箱, 例如:

```
Integer intObj = 10;//自动装箱
```

```
int i = intObj + 5;//自动拆箱
```

3. final 修饰符如何使用, 使用了 final 修饰符有什么特点?

final 修饰符可以修饰类, 方法, 变量

final 类为最终类, 不能被子类继承

final 方法为最终方法, 不能被子类覆盖

final 变量为常量, 只能赋一次值

4. abstract 修饰符如何使用, 使用了 abstract 修饰符有什么特点?

abstract 修饰符可以修饰类和方法

abstract 类为抽象类, 是不能创建对象的

abstract 方法为抽象方法, 只有方法的声明, 没有具体的实现

需要注意的是:

有抽象方法的类一定是抽象类

抽象类不一定有抽象方法

抽象类也有构造器, 但不能创建对象实例

5. abstract 不能和哪些修饰符配合使用?

不能和 private, static, final 一起使用

6. 可以在非抽象方法中调用抽象方法吗?

可以, 模板方法设计模式就是典型的这种调用

泛型程序设计

7. 泛型的特点是什么?

主要特点是类型参数化, 后期绑定数据类型

8. 泛型支持基本类型吗?

不支持, 只支持引用类型

9. 如果没有传入类型参数, 默认是什么类型?

默认是 Object 类型

10. 泛型方法有什么特点?

将泛型参数声明在方法中, 调用方法的时候绑定泛型

11. 泛型类声明泛型参数的时候, <T extends A 类型>表示什么意思?

表示绑定的类型必须是 A 类型后者是 A 类型的子类或实现类

接口和内部类

12. 声明一个接口的关键字是什么?

interface 关键字

13. 接口中的成员有什么特点？

数据域必须是（`public static final` 修饰）公有静态常量

方法必须是（`public abstract` 修饰）公有抽象

没有构造方法

14. 接口能创建对象实例吗？

不能，接口没有构造器，不能创建对象

15. 接口和类的关系？

接口和类之间是实现关系，一个类用 `implements` 语句实现了某接口，这个类被称为接口的实现类，一个类可以实现多个接口，一个类实现了某接口，必须覆盖这个接口中的所以方法，否则这个类必须是个抽象类

16. 接口和接口的关系？

接口和接口之间是继承关系，一个接口用 `extends` 继承另外一个接口，称为这个接口的子接口。

一个接口可以继承多个父接口

17. 内部类如何定义？

一个类声明在另一个类的里面，这个类就被称为内部类，而外面的类被称为外部类

18. 内部类可以分成几种？

实例内部类：声明在方法外边没有 `static` 修饰符

静态内部类：声明在方法外边有 `static` 修饰符

局部内部类：声明在方法里边

19. 如何创建实例内部类的对象实例？

需要用外部类的对象来创建

例如：

```
class A {  
    class B {  
    }  
}
```

如果像创建 `B` 类的实例，则必须用以下语句：

```
A a = new A();
```

```
A.B b = a.new B();
```

1. 如何在实例内部类中访问外部类的成员？

在实例内部类中可以直接使用外部类的成员，不需要创建外部类的对象实例，但是如果在内部类中的成员名称屏蔽隐藏了外部类的成员（即外部类和内部类的成员名相同），则需要用“外部类名.this.成员名”访问外部类成员

异常处理

1. 异常的对象根类是哪个类？

是 `Throwable` 类

2. `Throwable` 类有哪两个分支？

分成 Error 类和 Exception 类两个分支

Error 类表示严重错误

Exception 类表示可以在程序中处理的错误

3. Exception 类有哪两个分支？

分成运行时异常和非运行时异常（已检查异常）

运行时异常编译不检查，在运行时才会发生异常

已检查异常编译时就会作检查，不处理异常则通不过编译

4. 如果在 try 块中发生异常，那么程序将会跳转到哪？

会跳转到 catch 语句块

5. catch 语句可以用父类类型捕获子类异常吗？

因为有多态特点，是可以的

6. 处理异常代码 finally 语句会在什么时候执行？

finally 代码在任何时候都会执行，不管有没有发生异常。

7. 如何声明已检查异常，由谁来处理这个异常？

用 throws 语句来声明一个已检查异常，由方法的调用者来处理这个异常

8. 如何抛出一个异常？

用 throw 语句抛出

9. 如何创建一个自定义的异常类？

创建一个类，只要让它继承任何一个现有的异常类就可以了

10. 覆盖方法对于异常有什么要求？

覆盖方法不能比父类方法抛出更多异常

反射，代理，标注

11. 获得一个类型的 Class 对象有哪几种途径？

方法 1：对象.getClass()

方法 2：类型名.class

方法 3：Class.forName()

12. 反射相关的类型都有哪些？

Class 代表一个运行时类

Field 代表一个类的属性

Method 代表一个类的方法

Constructor 代表一个类的构造器

13. 如何声明一个自定义标注？

[<修饰符>] @interface <标注名>{

返回值方法名称（）；

返回值方法名称（）；

.....

```
}
```

例如:

```
public @interface MyComment {  
    int id();  
    String info();  
}
```

14. java 有几个内置标注?

有 3 个

@Override——用于修饰此方法覆盖了父类的方法，而非重载。

@Deprecated——用于修饰已经过时的方法。

@SuppressWarnings——用于通知 Java 编译器禁止特定的警告。

java 常用类

15. 字符串 String 是可变类还是不可变类?

是不可变类，一经创建就不能修改其内容了

16. 字符串的常用方法有哪些?

charAt 方法: 可以返回指定位置的字符

substring 方法: 可以截取子字符串

length 方法: 返回字符串长度

indexOf 方法: 查找子串的位置

lastIndexOf 方法: 反向搜索字符串的位置

replaceAll 方法: 替换字符串

split 方法: 分解字符串

toLowerCase, toUpperCase 方法: 字符串大小写转换

17. StringBuffer 类和 String 类有什么区别?

StringBuffer 是可变类，String 是不可变类

18. StringBuffer 类和 StringBuilder 类有什么区别?

StringBuffer 类是线程安全的，StringBuilder 类是线程不安全的

19. Math 类主要提供了哪些方法?

Math 类提供了大量的和数学运算相关的静态方法

20. Set 集合和 List 集合共同的父接口是什么?

是 Collection 集合

21. Set 集合和 List 集合有什么区别?

Set: Collection 的子接口，不记录元素的保存顺序，且不允许有重复元素

List: Collection 的子接口，记录元素的保存顺序，且允许有重复元素

22. ArrayList 和 LinkedList 有什么不同?

ArrayList 的内部实现是基于内部数组 Object[], 所以从概念上讲它更像数组

LinkedList 的内部实现是基于一组连接的记录，所以它更象一个链表结构，所以它们在性能上有很大的差别。

ArrayList 随机访问比较快，但插入和删除元素比较慢

LinkedList 但插入和删除元素比较快，但随机访问比较慢

23. Hashtable 和 HashMap 有什么不同？

Hashtable 是线程安全的，HashMap 是线程不安全的

24. Vector 和 ArrayList 有什么不同？

Vector 是线程安全的 ArrayList 是线程不安全的

IO 流

25. 流分成哪两类？

按照单位划分分成字节流和字符流。

按输入输出分成输入流和输出流

26. 和文件流相关的有那几个类？

FileInputStream 文件输入字节流

FileOutputStream 文件输出字节流

FileReader 文件输入字符流

FileWriter 文件输出字符流

27. 文件流操作应该用字节流还是字符流？

如果是纯文本文件应该用字符流

如果是二进制文件应该用字节流

28. 文件操作用哪个类？

用 java.io.File 类

线程

29. 创建一个线程类的两种方式？

继承 Thread 类

实现 Runnable 接口

30. 线程体应该放在哪个方法里面？

run 方法

31. 启动新的线程用什么方法？

用 start 方法

32. 一个线程对象能够用 start 多次启动吗？

不可以，一个线程方法只能 start 一次

33. 线程有哪几种主要状态？

有四种，分别是：

新生状态，可运行状态，阻塞状态，死亡状态

34. 线程的同步需要什么方法？

旧的方法是用 `synchronized` 关键字

新的方法是用 `ReentrantLock` 类的 `lock` 和 `unlock` 方法

书诚教育专营店