

第一部分 Java 基础方面

1、作用域 public,private,protected,以及不写时的区别

答: 区别如下:

作用域	当前类	同一 package	子孙类	其他 package
public	√	√	√	√
protected	√	√	√	×
friendly	√	√	×	×
private	√	×	×	×

不写时默认为 friendly

2、Anonymous Inner Class (匿名内部类) 是否可以 extends(继承)其它类, 是否可以 implements(实现)interface(接口)

答: 匿名的内部类是没有名字的内部类。不能 extends(继承) 其它类, 但一个内部类可以作为一个接口, 由另一个内部类实现

3、Static Nested Class 和 Inner Class 的不同

答: Nested Class (一般是 C++的说法), Inner Class (一般是 JAVA 的说法)。Java 内部类与 C++嵌套类最大的不同就在于是否有指向外部的引用上。注: 静态内部类 (Inner Class) 意味着 1 创建一个 static 内部类的对象, 不需要一个外部类对象, 2 不能从一个 static 内部类的一个对象访问一个外部类对象

4、&和&&的区别

答: &是位运算符, 表示按位与运算, &&是逻辑运算符, 表示逻辑与 (and)

5、Collection 和 Collections 的区别

答: Collection 是集合类的上级接口, 继承与他的接口主要有 Set 和 List.

Collections 是针对集合类的一个帮助类, 他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作

6、什么时候用 assert

答: assertion(断言)在软件开发中是一种常用的调试方式, 很多开发语言中都支持这种机制。在实现中, assertion 就是在程序中的一条语句, 它对一个 boolean 表达式进行检查, 一个正确程序必须保证这个 boolean 表达式的值为 true; 如果该值为 false, 说明程序已经处于不正确的状态下, 系统将给出警告或退出。一般来说, assertion 用于保证程序最基本、关键的正确性。assertion 检查通常在开发和测试时开启。为了提高性能, 在软件发布后, assertion 检查通常是关闭的

7、String s = new String("xyz");创建了几个 String Object

答: 两个, 一个字符对象, 一个字符对象引用对象

8、Math.round(11.5)等於多少? Math.round(-11.5)等於多少

答: Math.round(11.5)==12;Math.round(-11.5)=-11;round 方法返回与参数最接近的长整数, 参数加 1/2 后求其 floor

9、short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错

答: short s1 = 1; s1 = s1 + 1; (s1+1 运算结果是 int 型, 需要强制转换类型) short s1 = 1; s1 += 1; (可以正确编译)

10、Java 有没有 goto

答: java 中的保留字, 现在没有在 java 中使用

11、数组有没有 length()这个方法? String 有没有 length()这个方法

答: 数组没有 length()这个方法, 有 length 的属性。String 有有 length()这个方法

12、Overload 和 Override 的区别。Overloaded 的方法是否可以改变返回值的类型

答: 方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。重写 Overriding 是父类与子类之间多态性的一种表现, 重载 Overloading 是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数, 我们说该方法被重写 (Overriding)。子类的对象使用这个方法时, 将调用子类中的定义, 对它而言, 父类中的定义如同被“屏蔽”了。如果在一个类中定义了多个同名的方法, 它们或有不同的参数个数或有不同的参数类型, 则称为方法的重载(Overloading)。Overloaded 的方法是可以改变返回值的类型

13、Set 里的元素是不能重复的, 那么用什么方法来区分重复与否呢? 是用==还是 equals()? 它们有何区别

答: Set 里的元素是不能重复的, 那么用 iterator()方法来区分重复与否。equals()是判读两个 Set 是否相等

equals()和==方法决定引用值是否指向同一对象 equals()在类中被覆盖, 为的是当两个分离的对象的内容和类型相配的话, 返回真值

14、给我一个你最常见到的 runtime exception

答: 常见的运行时异常有如下这些 ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException, EmptyStackException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException, IllegalStateException, ImagingOpException, IndexOutOfBoundsException, MissingResourceException, NegativeArraySizeException, NoSuchElementException, NullPointerException, ProfileDataException, ProviderException, RasterFormatException, SecurityException, SystemException, UndeclaredThrowableException, UnmodifiableSetException, UnsupportedOperationException

15、error 和 exception 有什么区别

答: error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况

exception 表示一种设计或实现问题。也就是说, 它表示如果程序运行正常, 从不会发生的情况

16、List, Set, Map 是否继承自 Collection 接口

答: List, Set 是, Map 不是

17、abstract class 和 interface 有什么区别

答: 声明方法的存在而不去实现它的类被叫做抽象类 (abstract class), 它用于要创建一个体现某些基本行为的类, 并为该类声明方法, 但不能在该类中实现该类的情况。不能创建 abstract 类的实例。然而可以创建一个变量, 其类型是一个抽象类, 并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现, 否则它们也是抽象类为。取而代之, 在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法

接口 (interface) 是抽象类的变体。在接口中, 所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的, 没有一个有程序体。接口只可以定义 static final 成员变量。接口的实现与子类相似, 除了该实现类不能从接口定义中继承行为。当类实现特殊接口时, 它定义 (即将程序体给予) 所有这种接口的方法。然后, 它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类, 它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换, instanceof 运算符可以用来决定某对象的类是否实现了接口

18、abstract 的 method 是否可同时是 static,是否可同时是 native, 是否可同时是 synchronized

答: 都不能

19、接口是否可继承接口? 抽象类是否可实现(implements)接口? 抽象类是否可继承实体类(concrete class)

答: 接口可以继承接口。抽象类可以实现(implements)接口, 抽象类是否可继承实体类, 但前提是实体类必须有明确的构造函数

20、构造器 Constructor 是否可被 override

答: 构造器 Constructor 不能被继承, 因此不能重写 Overriding, 但可以被重载 Overloading