基础部分的知识

(a)java的代码初始化的顺序

1 静态优先于非静态

2 父类优先于子类

3 按照定义顺序进行初始化

具体的初始化的顺序

父类静态变量

父类静态代码块

子类静态变量

子类静态代码块

父类非静态变量

父类非静态代码块

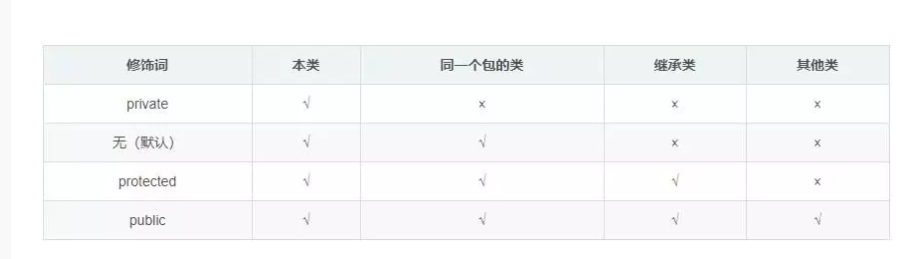
父类构造函数

子类非静态变量

子类非静态代码块

子类构造函数

(b)关于java修饰符



(c)关于继承

继承是复用的一种很好的方式,通过继承,子类可以拥有父类的方法和属性,从而达到复用的目的

继承的特点

子类只能继承一个父类

子类只能继承父类非私有的成员变量和方法

当子类定义了和父类相同的方法(名字和参数类型,参数顺序都相同)或者相同的成员变量的时候则会覆盖父类的成员变量和方法

(d)关于重载

重载是指在一个类里面定义了多个同名的但是参数不同方法,重载是通过参数来进行的,如参数名字,参数类型,参数个数,参数顺序,不能通过返回值,抛出异常等进行重载

(e)关于覆盖和重写

覆盖是指在派生类里面用完全一样的函数覆盖父类的方法然后进行重写,这个完全一样指的是方法名,返回值,参数,异常都一样,并且private的方法不能被覆盖

(f)关于抽象

抽象就是忽略一个主题中与当前目标无关的那些方面,以便更好地注意到与当前有关的方面,抽象并不打算了解全部问题,而是选择其中的一部分去了解,有过程抽象和数据抽象

过程抽象:忽略任务具体完成的过程，只精确描述该任务所要完成的功能  
数据抽象:利用数学结构(如集合、关系、函数、序列等)进行功能性描述，而不关心这些抽象数学结构在计算机中是如何具体表示和实现的

(g)构造函数

构造函数与类名相同,没有返回值

没有构造函数的时候会自动生成一个无参的构造函数

构造函数可以有多个,构成重载

构造函数主要完成类的初始化工作,因此它和new关键字相关

子类不能继承父类的构造函数,在子类进行new的时候会先执行父类的构造函数.

父类没有提供无参数的构造函数的时候,子类需要显式地用(super)调用父类的构造函数,如果父类拥有无参的构造函数则子类不用显式调用,需要注意的是,每次new都会先调用父类的构造函数,然后调用子类的构造函数

当父类和子类都没有构造函数的时候,会自动为父类子类都生成一个构造函数

(h)java中的clone

Clone是用原来的对象生成一个新的对象,新的对象的修改对原来的对象是没有影响的

浅复制:是指被复制的所有对象和源对象的值相同,如果原来的对象里面存在对象的引用,那么复制出来的对象和原来的对象还是会直线同一个对象,没有实现完全脱离

深复制:会将对象里买的对象也进行复制,复制出来的对象和原来的对象完全没有联系

1. 反射

反射机制可以提供的主要功能有

得到对象所属的类

得到类的所有成员变量和方法

在运行时创建对象,运行时调用方法

(j)抽象类

抽象类和普通类的使用相似,只是抽象类里面即可以有非抽象方法又可以有抽象方法,并且抽象类不能实例化

(k)接口

接口是一种门面(外观模式),主要用于功能方法的一个抽象,接口里面全部都是未实现的方法,边切方法的修饰符都是public,接口可以定义成员变量,但是成员变量都是static finial的

(l)final关键字

Final用于修饰属性,方法,和类,分表表示属性不可变,方法不可覆盖,类不可继承

Final修饰变量:final修饰变量指的是引用不可变,它只能只能指向初始的时候指向的对象,不关心指向对象的内容的变化,所以final变量需要初始化

Final修饰方法:这个方法不能被覆盖,也就是子类不能重写这个方法,但是可以使用这个方法

Final修饰类:这个类不能被集成,并且所有方法不能重写,但是final里面的成员变量是可以改变的,想要做到final类里面的成员变量也不可变的话,需要将成员变量也声明为final的类型.

(m)关于方法参数传递

在java 中处理基本类型如int char double 等的时候才用值传递,除此之外才用的凑是引用传递,”=”采用的也是引用传递

(o)关于强制转换

在进行计算的时候byte short char 首先会转换为int然后进行计算,对于大转小则需要强制转换short i = 0; i=i+1;会报错,但是由于java机制 i +=1能够编译通过

(p)string stringbuffer stringBuilder

String 是不可变字符串

StringBuffer和stringBuilder是可变字符串,区别在于stringBuffer是线程安全的,stringBuilder非线程安全

(q)序列化与反序列化

序列化的目的是让对象可以在网络中传输,因为只有序列化之后java对象才可以传输,而拿到输出流之后,反序列化就可以实现将流转换为对象

(r)线程

Java中有前台线程和后台线程,只要存在前台线程在运行就不会终止整个进程,换句话说只有后台线程在运行的时候,进程会自动终止,

(s)后台线程作用

后台线程又叫做守护线程,其中垃圾回收期就是典型的守护线程,他的作用就是为其他线程服务.

(t)线程的锁

对于锁代码片段,那个锁对象是我们自己指定的,但是记住这个对象对于所有线程来说必须是只有一个的,这样线程才会有竞争一个线程的情况,对于synchronized锁的是方法的时候不用指定锁对象,他的默认的锁对象就是当前类的class对象,.

(u)wait notify

Wait放弃锁然后进入阻塞转台

Notify是唤醒此同步锁上第一次调用wait的线程

(v)线程安全的集合和字符串

线程安全的有vactor hashtable Stringbuffer

1. 关于线程安全的理解

对java虚拟机中,有主内存,每个线程也有自己的内存,线程同步采用minotor命令开启,当一个线程操作同步代码的时候,只有完成read(读取变量到自己的内存) load,usage,store之后然后将变量写回到主内存才会释放锁,从而实现线程同步

1. 逃逸分析

在进行对象分配的时候大部分对象都是直接在堆上分配,但是java虚拟机也可在栈上分配,首先要经过逃逸分析,如果一个局部对象的不会被多个对象和线程使用,那么这个对象是不会逃逸的,可以考虑栈上分配,如果一个对象存在被多个线程使用的可能则是会对象逃逸的

逃逸的例子:

Private StringBuffer getString(){

StringBuffer s = new StringBuffer(“a”);

1. append(“asdas”);

Return s;

}

s对象会逃逸出去,因为他通过返回值逃逸了

1. 对象栈上分配

对于逃逸分析的结果是非逃逸对象那么可以考虑栈上分配,因为在栈上分配的时候对象可以随着方法执行结束而销毁,无需垃圾回收器的介入,这样的话可以大大降低堆内存的压力,加快垃圾回收的速度,降低gc时间.

(z)类中包含了方法表,属性表,常量池,字段表等,在创建对象的时候是在对象里面存储类元信息,一家将属性表的东西复制一份给对象,(每个对象里面的属性是不一样的),在调用方法的时候是采用类元信息找到类,然后通过方法名和修饰符以及参数从类方法表找到对应的的方法,然后进行调用.多一方法是只有一份的,属性是每个对象都特有的,可以理解为,类是模板,属性是具体的,每个对象都要有的.

(aa)mysql数据库引擎

Isam与myisam:isqm和myisam是数据库的主流引擎之一,和innodb比较起来,myisam不支持事务,不支持细粒度的锁,在进行更新操作的时候会直接锁定整个表,因此更新效率较低,并且在数据损坏后难以修复,但是他的数据读取的效率很高,因此对于一些对数据安全性要求不高,数据读取的速度有要求的话可以使用,自动保存数据行数

Innodb :innbodb是一个事务型的存储引擎,提供了对事务的支持,支持行锁和外键,不保存数据的行数,在数据崩溃后可修复,使用b树索引和ha索引,在数据读取的速度上没有myisam的快,InnoDB行锁是通过给索引上的索引项加锁来实现的，这一点MySQL与[Oracle](http://www.2cto.com/database/Oracle/" \t "https://www.cnblogs.com/lushilin/p/_blank)不同，后者是通过在数据块中对相应数据行加锁来实现的。InnoDB这种行锁实现特点意味着：只有通过索引条件检索数据，InnoDB才使用行级锁，否则，InnoDB将使用表锁！

Memory:memory是在内存中创建的表,它是一个临时表,使用hsah索引,断电后直接消失,主要用于做一些临时表存储中间查询结果,他的查询效率很高,用完表格之后便被删除.

(ab)索引

①全值匹配。

②最佳左前缀法则：带头大哥不能死，中间兄弟不能断；带头大哥可跑路，老二也可跟着跑，其余兄弟只能死。

③索引列上不计算。

④覆盖索引记住用。

⑤不等于、is null、is not null导致索引失效。

⑥like百分加右边，加左边导致索引失效，解决方法：使用覆盖索引。

⑦字符串不加单引号导致索引失效。

⑧少用or，用or导致索引失效

(ac)mysql和oracle 的区别

1Mysql默认提交事务,oracle需要手动comit

2分页查询,mysql用limit,oracle使用rownum

Select \* from

(select rownum rn,t.\* from table t where rownum>20) table\_alis ta

Where ta.rn>10

3事务的隔离级别

Mysql使用的是读提交,oracle使用的是可重复读

4对于事务的支持,mysql只有innodb支持,oracle则是完全支持

5在锁的使用上,mysql只有在索引的支持上才能使用行锁,并且锁的是索引,没有索引的支持直接锁住表,oracle不需要索引的支持直接锁住数据.

6oracle 的函数更加丰富,如decode listtage 等,dblink,物化视图

7oracle收费不开源,mysql不收费开源