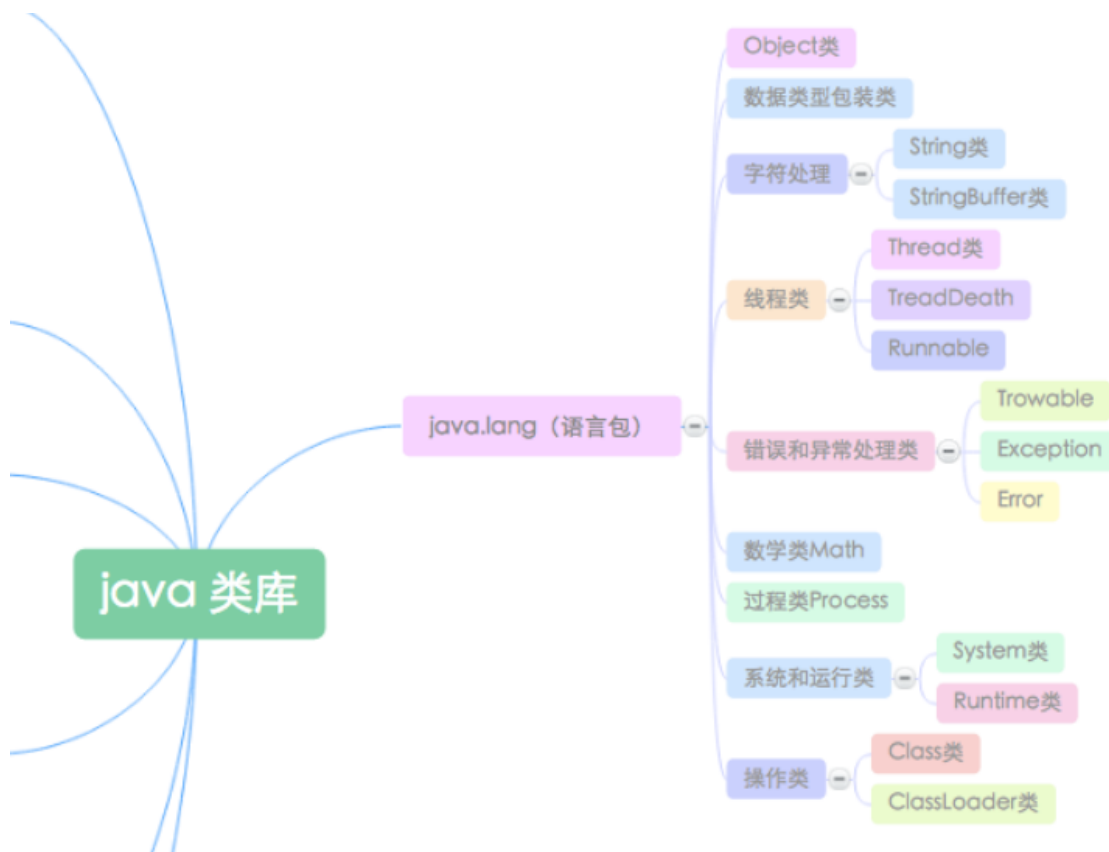


java.lang包细讲

开篇

在java中，java.lang包是特殊的一种，使用lang包不需要像其他包一样需要在程序写import关键字进行引入。系统会自动加载，所以我们可以直接使用其中的所有类。在这个类中所包含的类如下图：



在lang包中包含的类有Object类、包装类、字符处理类、线程类、错误和异常处理类、数学类、过程类、系统和运行类、操作类。

lang包类

01、Object类

在Object类中，相关的方法就不用说了吧：这些都是要掌握的方法

还是简单提一下吧: 在Object类中有 clone() (浅克隆)、equals() (比较方法, 一般重写)、hashCode() (用在集合的hash码)、getClass() (final方法, 获取运行时类型)、wait() (多线程知识点, 使当前线程睡眠)、notify() (唤醒线程, 一般与synchronized和wait一起使用)、notifyAll() (作用也是唤醒线程, 但唤醒的是所用线程)、toString() (转化为字符串)。

02、包装类

包装类的话:

原始数据类型	包装类
byte (字节)	Byte
char (字符)	Character
int (整型)	Integer
long (长整型)	Long
float (浮点型)	Float
double (双精度)	Double
boolean (布尔)	Boolean
short (短整型)	Short

在了解这些包装的时候，其实重点了解他们在对象和基本类型的比较问题上(因为这里涉及到包装类 中Cache问题)，具体的可以上网查查范围什么的，下面我就简单的介绍下缓存机制是什么和包装类中各Cache情况:

1、java 包装类的缓存机制，是在java 5中引入的一个有助于节省内存、提高性能的功能，只有在自动装箱时有效

2、java的包装类中: Byte, Short, Integer, Long, Character使用static代码块进行初始化缓存，其中Integer的最大值可以通过java.lang.Integer.IntegerCache.high设置; Boolean使用static final实例化的对象; Float和Double直接new的对象没有使用缓存

03、字符处理类

String、StringBuffer、StringBuilder三者之间的关系是经常被用来提问的问题，而这三者的比较主要是从三方面去比较(不变性、线程安全、性能)。

字符串String中常用的提取方法

方法	返回值	功能描述
indexOf(int ch)	int	搜索字符 ch 第一次出现的索引
indexOf(String value)	int	搜索字符串 value 第一次出现的索引
lastIndexOf(int ch)	int	搜索字符ch最后一次出现的索引
lastIndexOf(String value)	int	搜索字符串 value 最后一次出现的索引
substring(int index)	String	提取从位置索引开始到结束的字符串
substring(int beginindex, int endindex)	String	提取beginindex和endindex之间的字符串部分
trim()	String	返回一个前后不含任何空格的调用字符串的副本

StringBuffer中的构造方法:

构造方法	说明
StringBuffer()	构造一个其中不带字符的字符串缓冲区，其初始容量为 16 个字符
StringBuffer(CharSequence seq)	构造一个字符串缓冲区，它包含与指定的 CharSequence 相同的字符
StringBuffer(int capacity)	构造一个不带字符，但具有指定初始容量的字符串缓冲区
StringBuffer(String str)	构造一个字符串缓冲区，并将其内容初始化为指定的字符串内容

StringBuffer类的常用方法:

方法	返回值	功能描述
insert(int offsetm, Object s)	StringBuffer	在 offsetm 的位置插入字符串s
append(Object s)	StringBuffer	在字符串末尾追加字符串s
length()	int	确定 StringBuffer 对象的长度
setCharAt(int pos, char ch)	void	使用 ch 指定的新值设置 pos 指定的位置上的字符
toString()	String	转换为字符串形式
reverse()	StringBuffer	反转字符串
delete(int start, int end)	StringBuffer	删除调用对象中从 start 位置开始直到 end 指定的索引 (end-1) 位置的字符序列
replace(int start, int end, String s)	StringBuffer	使用一组字符替换另一组字符。将用替换字符串从 start 指定的位置开始替换，直到 end 指定的位置结束

不仅就这些方法，其他的可以去参考下JDK文档。

有人有疑问了,是不是讲漏了一个字符串类StringBuilder?

其实这个类和StringBuffer差不多，里面的方法和StringBuffer基本相同，因为他们都是继承同一个接口，但只是StringBuilder是牺牲安全去换取性能，而StringBuffer是牺牲性能去换取安全的差别而已。

04、线程类

别看这里面就几个类而已，但学起来的话，这里是写不完的。故这里就简单的提一下。

Thread类是java.lang包下的一个很重要的类，他是继承自Runnable接口，里面的知识点就与今后你的高度有关了。

在Thread有很多常用的方法:

常用方法	功能描述
static Thread currentThread()	返回对当前正在执行的线程对象的引用。
long getId()	返回该线程的标识符。
String getName()	返回该线程的名称。
int getPriority()	返回线程的优先级。
void interrupt()	中断线程。
boolean isAlive()	测试线程是否处于活动状态。
void join()	等待该线程终止。
void join(long millis)	等待该线程终止的时间最长为 millis 毫秒。
void join(long millis, int nanos)	等待该线程终止的时间最长为 millis 毫秒 + nanos 纳秒。
void setDaemon(boolean on)	将该线程标记为守护线程或用户线程。
void setPriority(int newPriority)	更改线程的优先级。
static void sleep(long millis)	在指定的毫秒数内让当前正在执行的线程休眠（暂停执行），此操作受到系统计时器和调度程序精度和准确性的影响。
static void sleep(long millis, int nanos)	在指定的毫秒数加指定的纳秒数内让当前正在执行的线程休眠（暂停执行），此操作受到系统计时器和调度程序精度和准确性的影响。
void start()	使该线程开始执行；Java 虚拟机调用该线程的 run 方法。
static void yield()	暂停当前正在执行的线程对象，并执行其他线程。

想想就恐怖，不过这些知识点掌握起来还是可以的，学过操作系统之后来学习就更容易一点，学习这方面的时候，要注意：**实践+分析！实践+分析！实践+分析！**

05、错误和异常处理类

这一部分，在往后学习中不会经常遇到，但在企业的时候是经常接触的，并且是自己去定义异常。

在java中，如果我们的程序中出现了不可挽回的错误(即代码编写错误等)，这类的错误叫error错误，而由于JVM中发生的错误等，大多都是Exception错误。

这一类的话，可以去看看网上的，这里也不细究

06、数学类

在编程的时候，有时候我们需要用到一些数学上的知识点的时候，例如sin等，如果要自己去写的话，虽然可以去编写程序，但一直这样的话，太浪费时间了，故java中自己提供了一个Math类，进行一些数学操作。

方法	返回值	功能描述
sin(double numvalue)	double	计算角 numvalue 的正弦值
cos(double numvalue)	double	计算角 numvalue 的余弦值
acos(double numvalue)	double	计算 numvalue 的反余弦
asin(double numvalue)	double	计算 numvalue 的反正弦
atan(double numvalue)	double	计算 numvalue 的反正切
pow(double a, double b)	double	计算 a 的 b 次方
sqrt(double numvalue)	double	计算给定值的正平方根
abs(int numvalue)	int	计算 int 类型值 numvalue 的绝对值，也接收 long、float 和 double 类型的参数
ceil(double numvalue)	double	返回大于等于 numvalue 的最小整数值
floor(double numvalue)	double	返回小于等于 numvalue 的最大整数值
max(int a, int b)	int	返回 int 型 a 和 b 中的较大值，也接收 long、float 和 double 类型的参数
min(int a, int b)	int	返回 a 和 b 中的较小值，也可接受 long、float 和 double 类型的参数
rint(double numvalue)	double	返回最接近 numvalue 的整数值
round(T arg)	arg 为 double 时返回 long，为 float 时返回 int	返回最接近arg的整数值
random()	double	返回带正号的 double 值，该值大于等于 0.0 且小于 1.0

上面都是一些常用的方法，如果以后还会用到极坐标、对数等，就去查一查手册吧。

07、过程类

对于过程类，就以后在讲吧。

我不会告诉你，我也不太会的。

08、系统和运行类

也就是相当于C语言中的print语句一样，《Java 编程思想》一书中也有介绍到这些类，对于java他不像C语言一样，它没有格式化的输入和输出，但在后来java引进了一系列的格式化输入输出，在format类中，如果有需要的话，可以去了解这个类。

老规矩，不细讲。这里倒不是不会的问题，在《Java 编程思想》这里面讲的比我的详细，这里吊大家胃口，让大家学会自己去看书，并学会自己去解决知识的盲点。

09、操作类

对于java.lang包下的Class在反射的时候经常会接触到。

Class 类的实例表示正在运行的 Java 应用程序中的类或接口。在 Java 中，每个 Class 都有一个相应的 Class 对象，即每一个类，在生成的 .class 文件中，就会产生一个 Class 对象，用于表示这个类的类型信息。我们获取 Class 实例有三种方法：

利用对象调用 `getClass()` 方法获取该对象的 Class 实例

使用 Class 类的静态方法 `forName(String className)`，用类的名字获取一个 Class 实例

运用 .class 的方式来获取 Class 实例，对于基本数据类型的封装类，还可以采用 .TYPE 来获取相对应的基本数据类型的 Class 实例

`forName()`这一方法可以与`newInstance`搭配去创建对象，具体的，可以去自行查阅书籍。