# 1 怎么理解oop

和面向对象编程不同的，是面向过程编程。面向过程编程，是把模型分解成一步一步的过程。比如，老板告诉你，要编写一个TODO任务，必须按照以下步骤一步一步来：

1. 读取文件；
2. 编写TODO；
3. 保存文件。

而面向对象编程，顾名思义，你得首先有个对象：



有了对象后，就可以和对象进行互动：

GirlFriend gf = **new** GirlFriend();//注意**new** GirlFriend()有括号

gf.name = "Alice";

gf.send("flowers");

也就是oop我要做一件事是一种我需要从A拿到什么资源我需要从B拿到什么资源的思想，所以操作都是基于谁谁谁，而pop就是我需要先做什么再做什么的思想，操作是基于步骤。

# 内容

面向对象的基本概念，包括：

### 类

* **实例**
* **方法**

class是一种对象模版，它定义了如何创建实例（众对象），因此，class本身就是一种数据类型，里面的包含字段（变量）和方法，字段就是类的特征，方法是类的接口

面向对象的实现方式，包括：

* **继承**

instanceof实际上判断一个变量所指向的实例是否是指定类型，或者这个类型的子类（本类或子类）

* **多态**

### 多态

多态是指，针对某个类型的方法调用，其真正执行的方法取决于运行时期实际类型的方法

**——Java的多态只需要直接在子类覆写override（或者叫重写）就可以了，不用再父类加什么virtual。**

Java语言本身提供的机制，包括：

* package
* classpath
* jar

以及Java标准库提供的核心类，包括：

* 字符串
* 包装类型
* JavaBean
* 枚举
* 常用工具类

# 常用类

## 1 字符串—既是基本类也成为Java的基本数据类型之一了

在Java中，String是一个引用类型（大概就是引入作为数据类型的意思），因为String太常用了，所以Java提供了"..."这种字符串字面量表示方法。

## 2 编码

起初是有了ASCII，后各国，再Unicode来统一，再UTF-8来改进--变长。综上UTF-8是最高级的，掌握两种ASCII和UTF-8

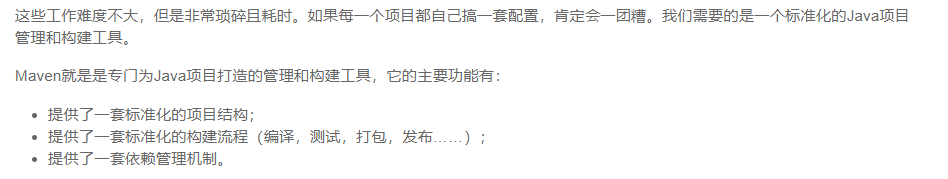
始终牢记：Java的String和char在内存中总是以Unicode编码表示。

PS: 还有一个也很常见的GBK是国标码的意思。也就是汉字编码方式。也常常要跟UTF-8转换

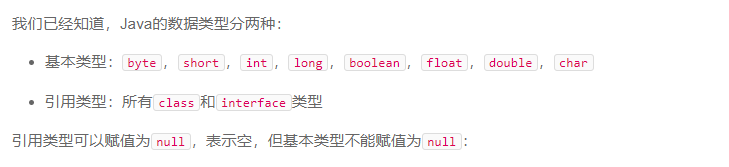
# 作用域



# Maven—Java项目标准化管理与构建工具



# Java数据类型--基本类型和引用类型



基本类型编程引用类型通过java核心库中的包装类来实现

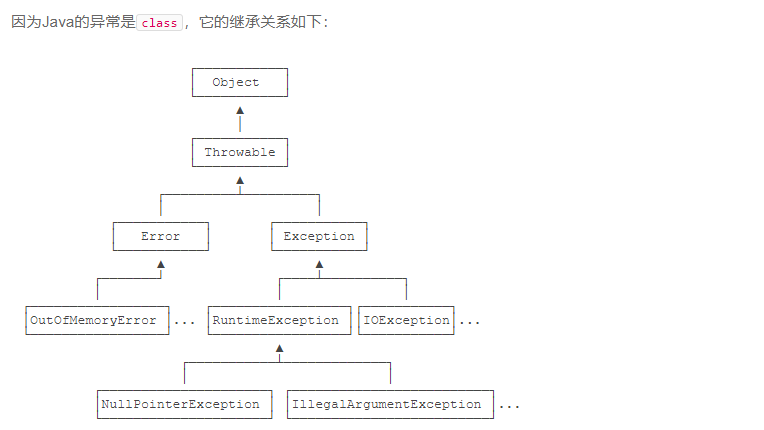
<https://www.liaoxuefeng.com/wiki/1252599548343744/1260473794166400>



# 优先静态工厂方法



# 异常—也就是代码块会由于一些错误操作或者我们自定义的一些禁止操作产生异常，那么我们就需要对异常进行处理



异常处理就是try{}catch{}，格式跟if else 差不多的，逻辑就是try里面的代码有异常则转向catch去处理

static byte[] toGBK(String s) {

try {

// 用指定编码转换String为byte[]:

return s.getBytes("GBK");

} catch (UnsupportedEncodingException e) {

// 如果系统不支持GBK编码，会捕获到UnsupportedEncodingException:

System.out.println(e); // 打印异常信息

return s.getBytes(); // 尝试使用用默认编码

}

}



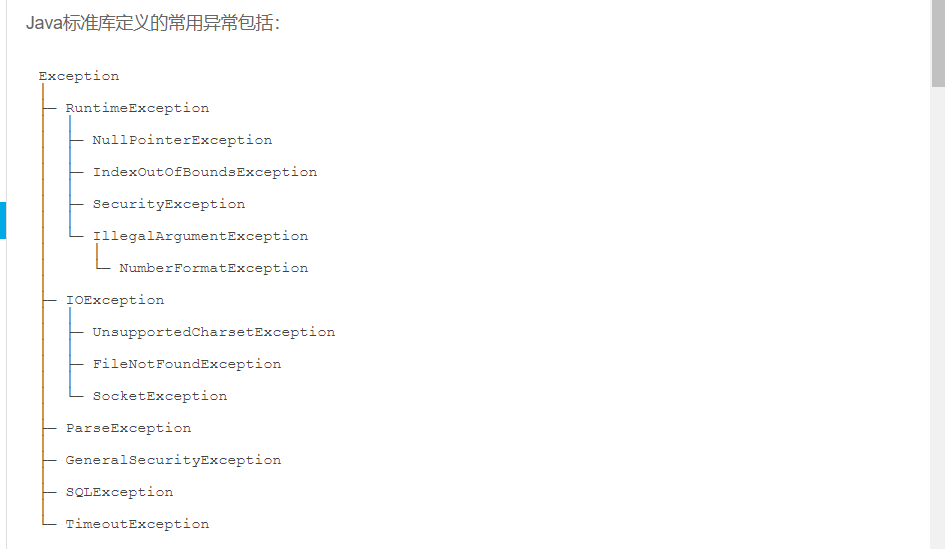
## 子类写前面—就相当于大集合小集合的关系，父类大集合包含子集



## 抛出异常



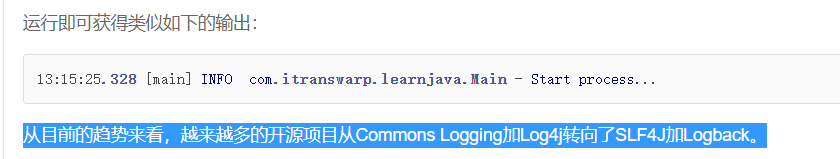
## 常用异常



## NullPointerException—Java并没有指针，只用引用类型。



# 日志 logger



# 注解—不是被忽略而是被看到的注释

# 小点

1. 可变参数（相当于数组类型）和for(类型 n:ns)的数组遍历操作

**public** **double** totalTax(Income... incomes) {

**double** total = 0;

**for** (Income income: incomes) {

total = total + income.getTax();

}

**return** total;

}

2 if else常见的写法，就是直接if，然后else下一行语句。

public double getTax() {

if (income <= 5000) {

return 0;

}

return (income - 5000) \* 0.2;//实际上这个就是else的逻辑内容，跟 else { return //(income - 5000) \* 0.2;}一样的，算法题经常这么写

}

}