# Type —— Java类型

　　Type是一个空接口，所有类型的公共接口（父接口），其意义表示Java所有类型，这里所谓的类型是从Java整个语言角度来看的，比如原始类型、参数化类型（泛型）、类型变量及其数组等，可以理解为，Class（类）是Java对现实对象的抽象，而Type是对Java语言对象的抽象。

## Type的子孙

### 概述

　　注意区分类型（Type）与类（Class）的区别，这里Class是Type的一种，而像数组、枚举等“类型”是相对于Class来说。

### Type的直接子接口

1.ParameterizedType： 表示一种参数化的类型，比如Collection，即普通的泛型。   
2.TypeVariable：是各种类型变量的公共父接口，就是泛型里面的类似T、E。   
3.GenericArrayType：表示一种元素类型是参数化类型或者类型变量的数组类型，比如List<>[]，T[]这种。   
4.WildcardType：代表一种通配符类型表达式，类似? super T这样的通配符表达式。

### Type的直接子类

　　Type的直接子类只有一个，也就是Class，代表着类型中的原始类型以及基本类型。

## Type子接口解析

### 1.ParameterizedType

　　参数化类型，即泛型，类似List<T>、Map<Integer, String>、List<? extends Number>带有类型参数的类型，当然，也可以是自定义的，再调用getRawType()与getActualTypeArguments()两个方法，就可以得到声明此参数化类型的类(java.lang.Comparable)和实际的类型参数数组([? super T])，而这个? super T又是一个WildcardType类型。

### 2.TypeVariable

　　类型变量，如参数化类型中的E、K等类型变量，表示泛指任何类，如果加上extends/super限定，则就会有相应的上限、下限。

### 3.GenericArrayType

　　泛型数组，表示上面两种的数组类型，即形如：A<T>[]，T[][]类型。

### 4.WildcardType

　　通配符表达式，或泛型表达式，它虽然是Type的一个子接口，但并不是Java类型中的一种，表示的仅仅是类似 ? extends T、? super K这样的通配符表达式。

## Java类型分类

Java的所有类型包括：   
1. raw type：原始类型，对应Class   
2. parameterized types：参数化类型，对应ParameterizedType   
3. array types：数组类型，对应GenericArrayType   
4. type variables：类型变量，对应TypeVariable   
5. primitive types：基本类型，仍然对应Class   
1). 1中的Class，不仅仅指平常所指的类，还包括数组、接口、注解、枚举等结构。   
2). 3中的数组类型GenericArrayType，应该指的是2、4类型数组，而不是一般我们说的数组，我们一般所说的数组是指1、5类型数组，他们还是1，也就是Class类型。   
3). List<T ? entends>[]：这里的List就是ParameterizedType，T就是TypeVariable，T ? entends就是WildcardType（注意，WildcardType不是Java类型，而是一个表达式），整个List<T ? entends>[]就是GenericArrayType。

## Type的来历

　　我们知道，Type是JDK5开始引入的，其引入主要是为了泛型，没有泛型的之前，只有所谓的原始类型。此时，所有的原始类型都通过字节码文件类Class类进行抽象。Class类的一个具体对象就代表一个指定的原始类型。   
　　泛型出现之后，也就扩充了数据类型。从只有原始类型扩充了参数化类型、类型变量类型、泛型数组类型，也就是Type的子接口。   
　　那为什么没有统一到Class下，而是增加一个Type呢？（Class也是种类的意思，Type是类型的意思）   
　　是为了程序的扩展性，最终引入了Type接口作为Class，ParameterizedType，GenericArrayType，TypeVariable和WildcardType这几种类型的总的父接口。这样实现了Type类型参数接受以上五种子类的实参或者返回值类型就是Type类型的参数。