转自：http://www.cnblogs.com/rollenholt/archive/2011/09/02/2163758.html

【案例】：获得其他类中的全部构造函数

这个例子需要在程序开头添加import java.lang.reflect.*;

然后将主类编写为：

```
class hello{
    public static void main(String[] args) {
        Class<?> demo=null;
        try{
            demo=Class.forName("Reflect.Person");
        }catch (Exception e) {
            e.printStackTrace();
        }
        Constructor<?>cons[]=demo.getConstructors();
        for (int i = 0; i < cons.length; i++) {
            System.out.println("构造方法:    "+cons[i]);
        }
    }
}
//~out:构造方法:    public Reflect.Person()
//~out:构造方法：  public Reflect.Person(java.lang.String)
```

但是细心的读者会发现，上面的构造函数没有public 或者private这一类的修饰符

下面这个例子我们就来获取修饰符

```
class hello{
    public static void main(String[] args) {
        Class<?> demo=null;
        try{
            demo=Class.forName("Reflect.Person");
        }catch (Exception e) {
            e.printStackTrace();
        }
        Constructor<?>cons[]=demo.getConstructors();
        for (int i = 0; i < cons.length; i++) {
            //参数的类型(也是Class,比如Integer, String)
            Class<?> p[] = cons[i].getParameterTypes();
            System.out.print("构造方法:    ");
            //Modifiers修饰符,PUBLIC:1, PRIVATE:2, PROTECTED:4,
            //STATIC:8, FINAL:16,SYNCHRONIZED:32, VOLATILE:64
            int mo = cons[i].getModifiers();
            System.out.print(Modifier.toString(mo)+" ");
            System.out.print(cons[i].getName());
```

```
            System.out.print("(");
            for(int j=0;j<p.length;++j){
                System.out.print(p[j].getName()+" arg"+i);
                if(j<p.length-1){
                    System.out.print(",");
                }
            }
            System.out.println("){}");
        }
    }
}
//~out:构造方法： public Reflect.Person(){}
//~out:构造方法： public Reflect.Person(java.lang.String arg1){}
```

有时候一个方法可能还有异常，呵呵。下面看看：

```
class hello{
    public static void main(String[] args) {
        Class<?> demo=null;
        try{
            demo=Class.forName("Reflect.Person");
        }catch (Exception e) {
            e.printStackTrace();
        }
        Method method[] = demo.getMethods();
        for(int i=0;i<method.length;++i){
            //返回类型
            Class<?> returnType = method[i].getReturnType();
            //参数类型
            Class<?> para[] = method[i].getParameterTypes();
             //修饰符
            int temp = method[i].getModifiers();
            System.out.print(Modifier.toString(temp)+" ");
            System.out.print(returnType.getName()+"  ");
            System.out.print(method[i].getName()+" ");
            System.out.print("(");
            for(int j=0;j<para.length;++j){
                System.out.print(para[j].getName()+" "+"arg"+j);
                if(j<para.length-1){
                    System.out.print(",");
                }
            }
            //异常
            Class<?> exce[]  =method[i].getExceptionTypes();
            if(exce.length>0){
                System.out.print(") throws ");
```

```
                for(int k=0;k<exce.length;++k){
                    System.out.print(exce[k].getName()+" ");
                    if(k<exce.length-1){
                        System.out.print(",");
                    }
                }
            }else{
                System.out.print(")");
            }
            System.out.println();
        }
    }
}
```

【运行结果】：

```
public java.lang.String  getSex ()
public void  setSex (java.lang.String arg0)
public void  sayChina ()
public void  sayHello (java.lang.String arg0,int arg1)
public final native void  wait (long arg0) throws
java.lang.InterruptedException
public final void  wait () throws java.lang.InterruptedException
public final void  wait (long arg0,int arg1) throws
java.lang.InterruptedException
public boolean  equals (java.lang.Object arg0)
public java.lang.String  toString ()
public native int  hashCode ()
public final native java.lang.Class  getClass ()
public final native void  notify ()
public final native void  notifyAll ()
```