Atitit java**[方法引用(Method References)](http://blog.csdn.net/mn960mn/article/details/49155993)** 与c#委托与脚本语言js的函数指针

**[1.1.](#_Toc24387)** [java](#_Toc24387)**[方法引用(Method References) 与c#委托与脚本语言js的函数指针这几个概念是类似的，](#_Toc24387)** [1](#_Toc24387)

**[1.2.](#_Toc14049)** [java](#_Toc14049)**[方法引用(Method References)](#_Toc14049)** [方法引用分为4类 1](#_Toc14049)

[1.3. 范例 3](#_Toc25927)

## java**[方法引用(Method References)](http://blog.csdn.net/mn960mn/article/details/49155993) 与c#委托与脚本语言js的函数指针这几个概念是类似的，**

**C#委托本质是函书接口，js是duck type原理。**

## java**[方法引用(Method References)](http://blog.csdn.net/mn960mn/article/details/49155993)** 方法引用分为4类

方法引用的唯一用途是支持Lambda表达式的简写，不用Lambda表达式，就用不着方法引用

方法引用分为4类，常用的是前三种。方法引用也受到访问控制权限的限制，可以通过在引用位置是否能够调用被引用方法来判断。具体分类信息如下：

引用静态方法   
ContainingClass::staticMethodName   
例子: String::valueOf，对应的Lambda：(s) -> String.valueOf(s)   
比较容易理解，和静态方法调用相比，只是把.换为::

引用特定对象的实例方法   
containingObject::instanceMethodName   
例子: x::toString，对应的Lambda：() -> this.toString()   
与引用静态方法相比，都换为实例的而已

引用构造函数   
ClassName::new   
例子: String::new，对应的Lambda：() -> new String()   
构造函数本质上是静态方法，只是方法名字比较特殊。

引用特定类型的任意对象的实例方法   
ContainingType::methodName   
例子: String::toString，对应的Lambda：(s) -> s.toString()   
太难以理解了。难以理解的东西，也难以维护。建议还是不要用该种方法引用。   
实例方法要通过对象来调用，方法引用对应Lambda，Lambda的第一个参数会成为调用实例方法的对象。

## 范例

/\*\*

\* 静态方法当作另外一个静态方法

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

*test*( class1::*method2*);

}

**private** **static** **void** test( Runnable object) {

object.run();

}

**private** **static** **void** method2() {

System.***out***.println("aa");

}

Ref参考资料

JDK1.8新特性之方法引用(Method References) - 走马行酒醴，驱车布鱼肉 - 博客频道 - CSDN.NET.html

作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ）

汉字名：艾提拉（艾龙），   EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

Atiend