方法引用

李玮玮

讲授思路

- Lambda表达式回顾
- 方法引用要求
- 方法引用的例子

Lambda表达式

- Lambda表达式是方法的实现
- Lambda表达式语法格式

  ([参数1],[参数2],...)->{ }
- 作用：
  - 匿名内部类的简化写法
- Lambda表达式的局限性
  - 接口
  - 仅包含一个抽象方法

方法引用

- 可看做是一个Lambda表达式
- 需要一个明确的目标类型充当功能性接口的实例
  - 被引用的方法与功能接口的 SAM(Single Abstract Method) 参数、返回类型相匹配
- 借用 C++ 的作用域解析操作符 “::”
- 作用
  - 提高复杂的Lambda表达式的可读性
  - 逻辑更清晰

方法引用条件

- 方法签名与功能性接口的 SAM 一致
  - 功能接口有唯一的抽象方法
  - 引用方法的参数与抽象方法的参数列表相同

- 例子：

  Runnable r = MyProgram::main;

  void main(String... args) 与 run() 方法能配上对

方法引用形式

- 静态方法 (ClassName::methName)
- 对象的实例方法 (instanceRef::methName)
- 对象的super 方法 (super::methName)
- 类型的实例方法 (ClassName::methName, 引用时和静态方法是一样的，但这里的 methName 是个实例方法)
- 类的构造方法 (ClassName::new)
- 数组的构造方法 (TypeName[]::new)

方法引用总结

- 提高Lambda表达式的可读性
- 将Lambda 表达式抽取到一个方法中，然后用方法引用指向这个方法
- 被引用的方法签名与功能性接口的 SAM保持一致
- 引用实例方法时，SAM 的第一个参数将作为引用方法的接收者
- 数组可理解为有一个接收整数的构造方法

Thank You