千锋教育™
WWW.QFEDU.COM

JDK 8

Java Platform Standard Edition

Java教学部

课程目标

CONTENTS

什么是Lambda

- 概念：允许把函数作为一个方法的参数。（函数作为参数传递到方法中）

```
<函数式接口>  <变量名> = (参数1,参数2...) -> {

        //方法体

}
```

注意事项

- 新的操作符 ->（箭头操作符）
  - (参数1,参数2)->表示参数列表
  - ->{ }方法体

- 形参列表的数据类型会自动推断
- 如果形参列表为空，只需保留()
- 如果形参只有1个，()可以省略，只要参数名字即可
- 如果执行语句只有1句，且无返回值，{}可以省略。
- 若有返回值，仍想省略{}，return也省略。保证执行语句只有1句
- Lambda表达式不会生成单独的内部类文件
- lambda访问局部变量时，变量要修饰final，如果没加，会自动添加

函数式接口

- 如果一个接口只有一个抽象方法，则该接口称为函数式接口。
- 为了确保接口达到要求，可以添加@FunctionalInterface注解。
- 内置四个核心函数式接口：
- Consumer<T> 消费型接口　　void accept(T t);
- Supplier<T> 供给型接口　　T get();
- Function<T,R> 函数型接口　　R apply(T t);
- Predicate<T> 断言型接口　　boolean test(T t);

方法引用

- 方法引用是Lambda表达式的一种简写形式，如果Lambda表达式方法体中只是调用一个特定的已存在的方法，则可以使用方法引用

- 使用 :: 操作符将对象或类和方法名的名字分隔开来。
  - 对象::实例方法
  - 类::静态方法
  - 类::实例方法
  - 类::new
- 注意：调用的方法参数列表与返回值类型，要与函数型接口中的方法参数列表与返回值类型一致

Stream API

- Stream是Java8中处理数组、集合的抽象概念。
- 可以执行非常复杂的查找、过滤、映射等操作。

常用方法（中间操作）

- Stream<T> filter(Predicate<? super T> predicate); 过滤
- Stream<T> limit(long maxSize); 截断，不超过给定数量
- Stream<T> distinct(); 筛选，利用hashCode和equals
- <R> Stream<R> map(Function<? super T, ? extends R> mapper);
- Stream<T> sorted(); 自然排序
- Stream<T> sorted(Comparator<? super T> comparator); 定制排序

常用方法（终止操作）

- long count(); 返回流中元素的总个数
- void forEach(Consumer<? super T> action); 遍历
- boolean anyMatch(Predicate<? super T> predicate); 是否至少匹配一个
- boolean allMatch(Predicate<? super T> predicate); 是否匹配所有元素
- boolean noneMatch(Predicate<? super T> predicate); 是否没有匹配
- Optional<T> findFirst(); 返回第一个
- Optional<T> findAny(); 返回任意
- Optional<T> min(Comparator<? super T> comparator); 返回最小
- Optional<T> max(Comparator<? super T> comparator); 返回最大

总结

- lambda表达式
- 函数式接口
- 方法引用
- Stream API
- 获得Stream 流
- 中间操作
- 终止操作