

千锋Java学院出品

Java8 新特性

Java Platform Enterprise Edition

课程目标

CONTENTS

ITEMS **1** Java8概述

ITEMS **2** Lambda表达式

ITEMS **3** 函数式接口

ITEMS **4** 方法引用

ITEMS **5** Stream API

ITEMS **6** 新时间API

- Java8 (又称 JKD1.8) 是 Java 语言开发的一个主要版本。
- Oracle公司于2014年3月18日发布Java8 。
 - 支持Lambda表达式
 - 函数式接口
 - 新的Stream API
 - 新的日期 API
 - 其他特性

Lambda表达式

- Lambda表达式：特殊的匿名内部类，语法更简洁。
- Lambda表达式允许把函数作为一个方法的参数（函数作为方法参数传递），将代码像数据一样传递。

- 基本语法：

```
<函数式接口> <变量名> = (参数1, 参数2...) -> {  
    //方法体  
};
```

- Lambda引入了新的操作符:->(箭头操作符),->将表达式分成两部分
 - 左侧: (参数1, 参数2...)表示参数列表
 - 右侧: {}内部是方法体
- 注意事项:
 - 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, 只需要参数的名称即可。
 - 如果执行语句只有一句, 且无返回值, {}可以省略, 若有返回值, 则若想省去{}, 则必须同时省略return, 且执行语句也保证只有一句。
 - Lambda不会生成一个单独的内部类文件。

- 如果一个接口只有一个抽象方法，则该接口称之为函数式接口，函数式接口可以使用Lambda表达式，Lambda表达式会被匹配到这个抽象方法上。
- `@FunctionalInterface` 注解检测接口是否符合函数式接口。

常见函数式接口

函数式接口	参数类型	返回类型	说明
Consumer<T> 消费型接口	T	void	void accept(T t);对类型为T的对象应用操作
Supplier<T> 供给型接口	无	T	T get(); 返回类型为T的对象
Function<T,R> 函数型接口	T	R	R apply(T t);对类型为T的对象应用操作，并返回类型为R类型的对象。
Predicate<T> 断言型接口	T	boolean	boolean test(T t);确定类型为T的对象是否满足条件，并返回boolean类型。

- 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经存在的方法，则可以使用方法引用。
- 常见形式
 - 对象::实例方法
 - 类::静态方法
 - 类::实例方法
 - 类::new

什么是Stream

- 流（Stream）与集合类似，但集合中保存的是数据，而Stream中保存对集合或数组数据的操作。



Stream特点

- Stream 自己不会存储元素。
- Stream 不会改变源对象。相反，他们会返回一个持有结果的新Stream。
- Stream 操作是延迟执行的，会等到需要结果的时候才执行。

Stream使用步骤

- **创建：**
 - 新建一个流。
- **中间操作：**
 - 在一个或多个步骤中，将初始Stream转化到另一个Stream的中间操作。
- **终止操作：**
 - 使用一个终止操作来产生一个结果。该操作会强制之前的延迟操作立即执行，在此之后，该Stream就不能使用了。

创建Stream

- 通过Collection对象的stream()或parallelStream()方法。
- 通过Arrays类的stream()方法。
- 通过Stream接口的of()、iterate()、generate()方法。
- 通过IntStream、LongStream、DoubleStream接口中的of、range、rangeClosed方法。

中间操作、终止操作

- 中间操作：
 - filter、limit、skip、distinct、sorted
 - map
 - parallel
- 终止操作：
 - forEach、min、max、count
 - reduce、collect

- 之前时间API存在问题：线程安全问题、设计混乱。
- 本地化日期时间 API：
 - LocalDate
 - LocalTime
 - LocalDateTime
- Instant：时间戳。
- ZoneId：时区。
- Date、Instant、LocalDateTime的转换。
- DateTimeFormatter：格式化类。

总结

- **Lambda表达式：**
 - 允许把函数作为一个方法的参数传递。
- **函数式接口：**
 - Consumer
 - Supplier
 - Function
 - Predicate
- **Stream API：**
 - 把对数据的操作封装成一个流。
 - 步骤：创建、中间操作、终止操作。
- **新时间API**

THANK YOU



做真实的自己，用良心做教育