# 一、学习目标

1. 能够使用@FunctionalInterface注解

2. 能够自定义无参无返回函数式接口

3. 能够自定义有参有返回函数式接口

4. 能够理解**Lambda延迟执行**的特点

5. 能够使用Lambda作为方法的参数

6. 能够使用Lambda作为方法的返回值

7. 能够使用**Supplier**函数式接口

8. 能够使用**Consumer**函数式接口

9. 能够使用**Function**函数式接口

10. 能够使用**Predicate**函数式接口

# 第一章 函数式接口

## 1.1 Lambda

一般都叫Lambda表达式（λ expression）是一个[匿名函数](https://baike.baidu.com/item/%E5%8C%BF%E5%90%8D%E5%87%BD%E6%95%B0/4337265)，Lambda表达式基于数学中的[λ演算](https://baike.baidu.com/item/%CE%BB%E6%BC%94%E7%AE%97)得名，直接对应于其中的lambda抽象（lambda abstraction），是一个匿名函数，即**没有函数名的函数**。Lambda表达式可以表示[闭包](https://baike.baidu.com/item/%E9%97%AD%E5%8C%85/10908873)（注意和数学传统意义上的不同）。

## 1.2 编程范式

常见的编程范式有**命令式编程（Imperative programming）**,**函数式编程**，**逻辑式编程**。常见的面向对象是一种命令式编程。

**命令式编程：**是面向计算机的抽象，有**变量**（对应着存储单元），**赋值语句**（获取，存储），**表达式**（内存引用和算术运算），**控制语句**（跳转、循环等），一句话：命令式编程是一个**冯诺依曼机**的**指令序列**。

**函数式编程：**是面向数学的抽象，将计算描述为一种**表达式求值**，一句话：函数式程序就是一个**表达式**。

## 1.3 函数式接口

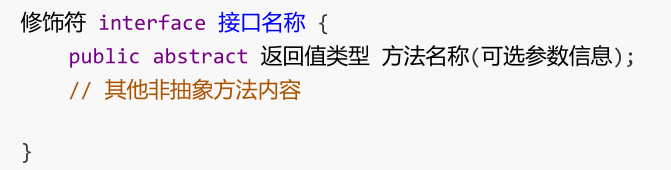
有且仅有一个抽象方法的接口。这个接口就是函数，就是一个表达式，所以可以使用Lambda表达式。

即适用于函数式编程场景的接口。而Java中的函数式编程体现就是Lambda，所以函数式接口就是可以适用于Lambda使用的接口。只有确保接口中有且仅有一个抽象方法，Java中的Lambda才能顺利地进行推导。

**“语法糖”**是指使用更加方便，但是原理不变的代码语法。例如在遍历集合时使用的for-each语法，其实底层的实现原理仍然是迭代器，这便是“语法糖”。从应用层面来讲，Java中的Lambda可以被当做是匿名内部类的“语法糖”，但是二者在原理上是不同的。

## 1.4 格式

只要确保接口中有且仅有一个抽象方法即可：



由于接口当中抽象方法的 public abstract 是可以省略的，所以定义一个函数式接口很简单：

public interface MyfactionaInterface {  
 void myMethod();  
}

#### 今日内容介绍

##### 1.1.1知识概述

* 今日内容介绍

##### 1.1.2视频详情



##### 1.1.3总结与补充

无

##### 1.1.4课堂提问与练习

无

##### 1.1.5习题答案

无

#### 函数式接口的概念&函数式接口的定义

##### 1.2.1知识概述

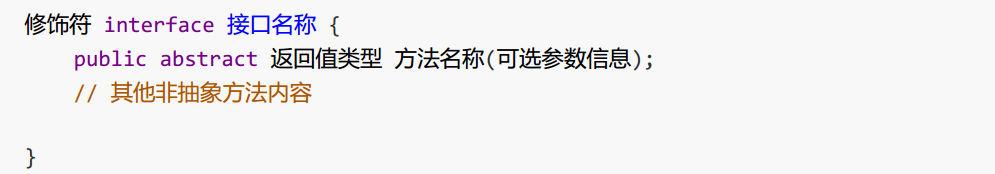
* 有且仅有一个抽象方法的接口

##### 1.2.2视频详情



##### 1.2.3总结与补充

函数式接口在Java中是指：有且仅有一个抽象方法的接口。



##### 1.2.4课堂提问与练习

无

##### 1.2.5习题答案

无

#### 函数式接口的使用

##### 1.3.1知识概述

* 使用函数式接口

##### 1.3.2视频详情



##### 1.3.3总结与补充

三种使用方式:

1. 单独编写接口实现类

2. 匿名内部类

3. Lambda



##### 1.3.4课堂提问与练习

无

##### 1.3.5习题答案

无

#### 性能浪费的日志案例

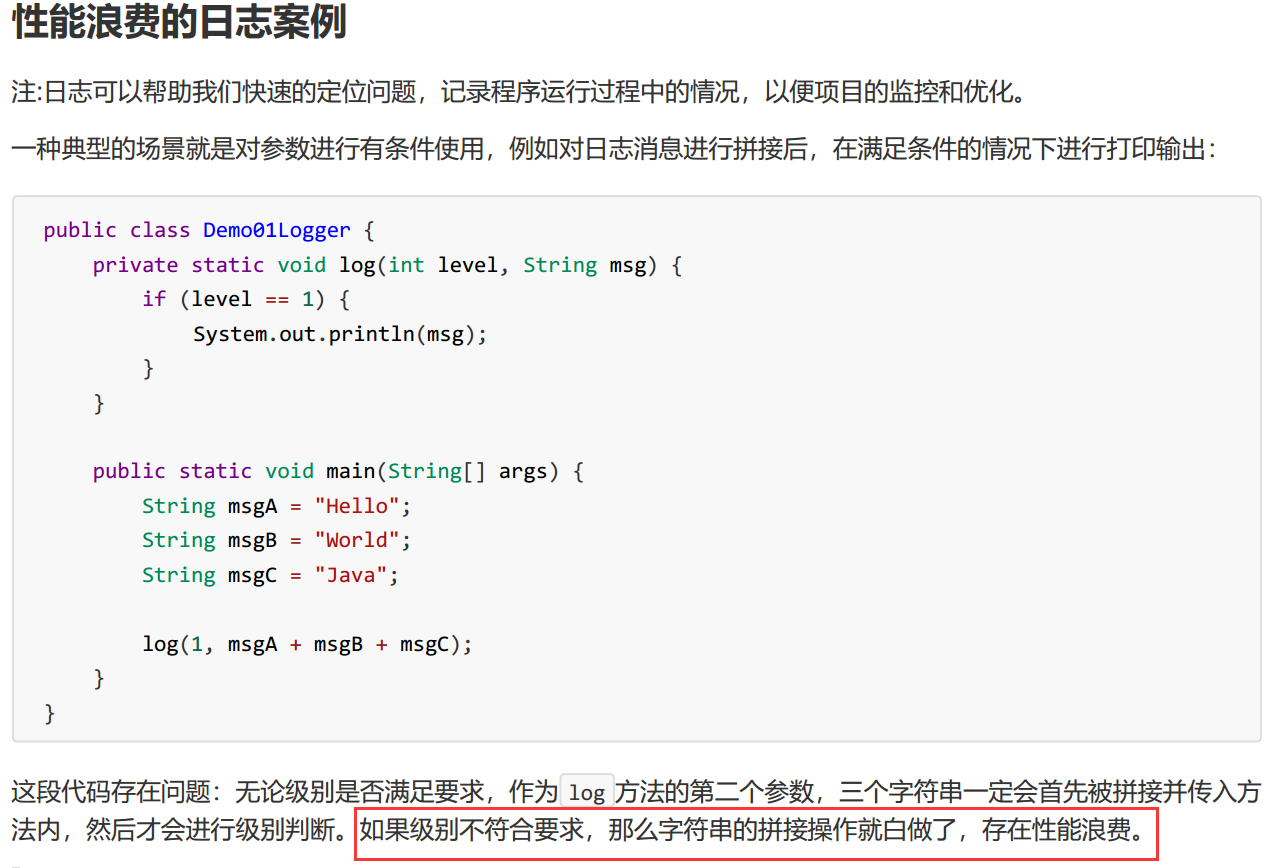
##### 1.4.1知识概述

* 案例演示, 引入Lambda延迟执行

##### 1.4.2视频详情



##### 1.4.3总结与补充



##### 1.4.4课堂提问与练习

无

##### 1.4.5习题答案

无

#### Thread类的常用方法\_设置线程名称的方法

##### 1.5.1知识概述

* setName();
* new Thread(“线程名字”);

##### 1.5.2视频详情



##### 1.5.3总结与补充

public Thread(String name) :分配一个指定名字的新的线程对象

public Thread(Runnable target,String name) :分配一个带有指定目标新的线程对象并指定名字

public void setName(String name) : 为线程对象设置名字.

##### 1.5.4课堂提问与练习

无

##### 1.5.5习题答案

无

### 第二堂课

本节知识点：

05\_使用Lambda优化日志案例

06\_函数式接口作为方法的参数案例

07\_函数式接口作为方法的返回值类型案例

视频时间：

24分55秒

本节目标 ：

1. 理解Lambda的延迟执行
2. 独立编写函数式接口作为方法的参数案例
3. 独立编写函数式接口作为方法的返回值类型案例

#### 使用Lambda优化日志案例

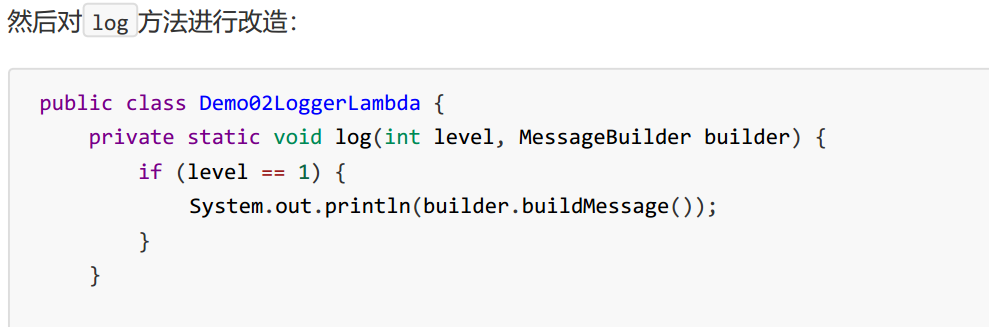
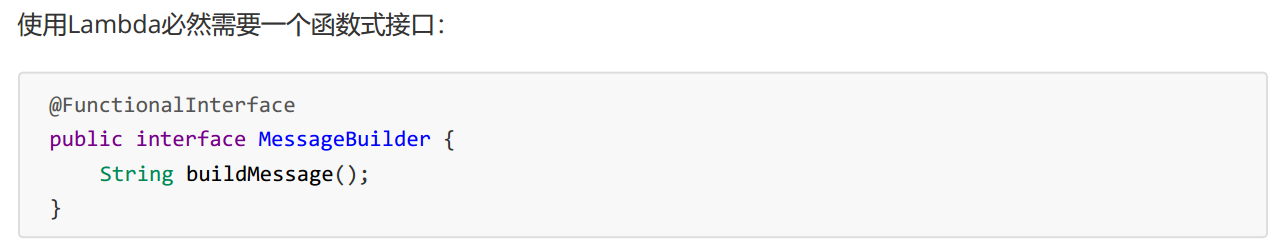
##### 2.1.1知识概述

* 使用Lambda优化日志案例

##### 2.1.2视频详情



##### 2.1.3总结与补充



##### 2.1.5习题答案

无

#### 函数式接口作为方法的参数案例

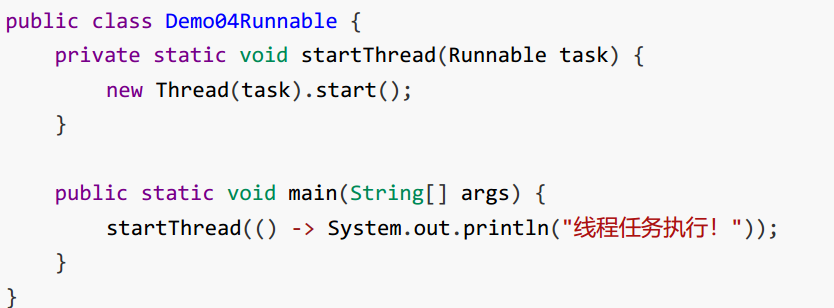
##### 2.2.1知识概述

* 使用Lambda开启线程-> Runnable

##### 2.2.2视频详情



##### 2.2.3总结与补充



##### 2.2.4课堂提问与练习

无

##### 2.2.5习题答案

无

#### 函数式接口作为方法的返回值类型案例

##### 2.3.1知识概述

* 函数式接口Comparator的使用

##### 2.3.2视频详情

##### 2.3.3总结与补充



##### 2.3.4课堂提问与练习

无

##### 2.3.5习题答案

无

### 第三堂课

本节知识点：

08\_常用的函数式接口\_Supplier接口

09\_常用的函数式接口\_Supplier接口练习\_求数组元素最大值

10\_常用的函数式接口\_Consumer接口

11\_常用的函数式接口\_Consumer接口的默认方法andThen

视频时间:

31分52秒

本节目标 ：

1. 理解生产型Supplier接口的作用
2. 独立编写使用Supplier求数组最大值案例
3. 理解消费型接口Consumer的作用
4. 掌握Consumer接口中andThen方法的使用

#### 常用的函数式接口\_Supplier接口

##### 3.1.1知识概述

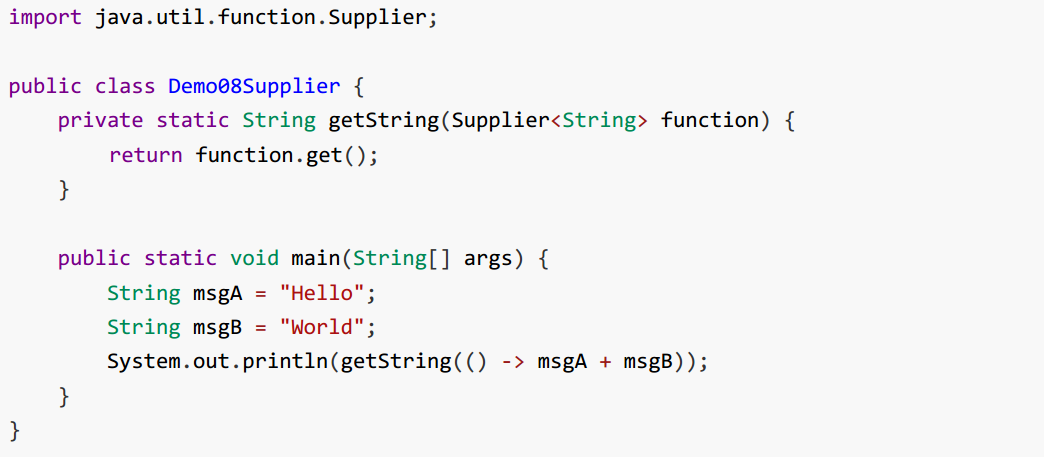
* 生产型接口Supplier

##### 3.1.2视频详情



##### 3.1.3总结与补充

java.util.function.Supplier<T> 接口仅包含一个无参的方法： T get() 。用来获取一个泛型参数指定类型的对象数据。由于这是一个函数式接口，这也就意味着对应的Lambda表达式需要“对外提供”一个符合泛型类型的对象数据。



##### 3.1.4课堂提问与练习

无

##### 3.1.5习题答案

无

#### 常用的函数式接口\_Supplier接口练习\_求数组元素最大值

##### 3.2.1知识概述

* 使用 Supplier 接口作为方法参数类型，通过Lambda表达式求出int数组中的最大值

##### 3.2.2视频详情



##### 3.2.3总结与补充



##### 3.2.4课堂提问与练习

无

##### 3.2.5习题答案

无

#### 常用的函数式接口\_Consumer接口

##### 3.3.1知识概述

* 消费型接口Consumer

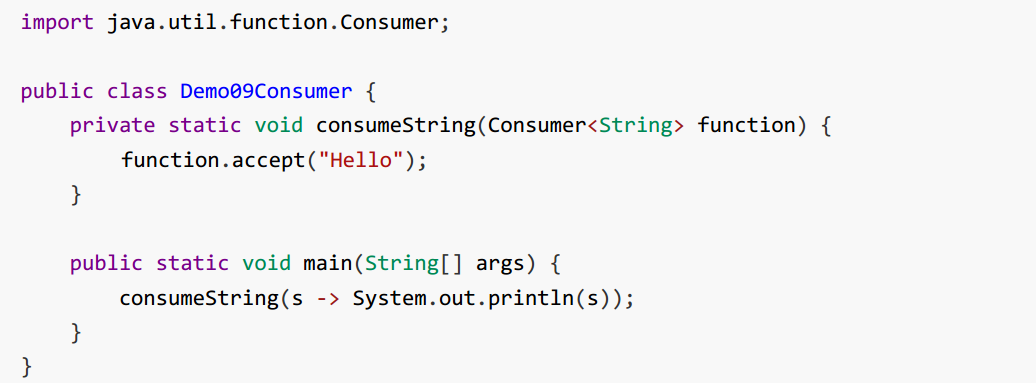
##### 3.3.2视频详情



##### 3.3.3总结与补充

java.util.function.Consumer<T> 接口则正好与Supplier接口相反，它不是生产一个数据，而是消费一个数据，其数据类型由泛型决定

Consumer 接口中包含抽象方法 void accept(T t) ，意为消费一个指定泛型的数据



##### 3.3.4课堂提问与练习

无

##### 3.3.5习题答案

无

#### 常用的函数式接口\_Consumer接口的默认方法andThen

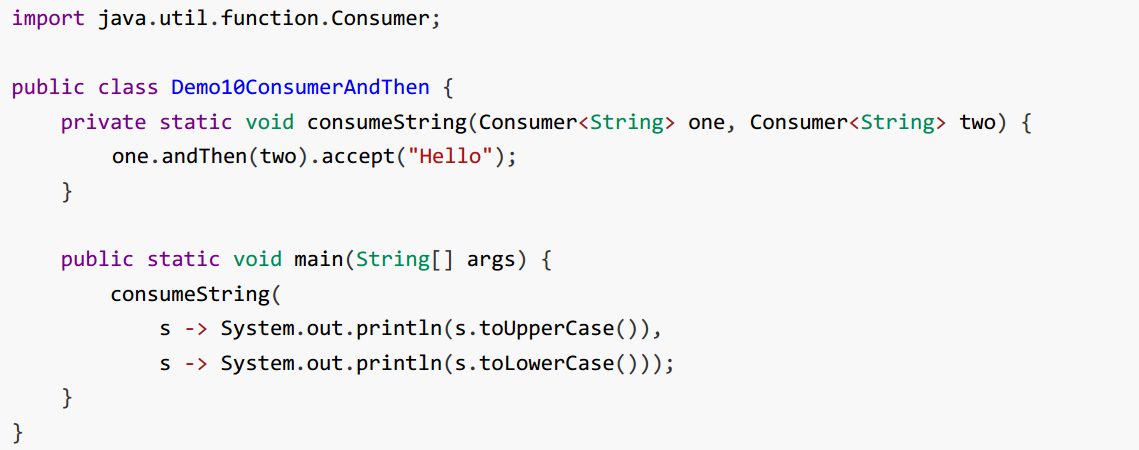
##### 3.3.1知识概述

* 如果一个方法的参数和返回值全都是 Consumer 类型，那么就可以实现效果：消费数据的时候，首先做一个操作，然后再做一个操作，实现组合

##### 3.3.2视频详情



##### 3.3.3总结与补充



##### 3.3.4课堂提问与练习

无

##### 3.3.5习题答案

无

### 第四堂课

本节知识点：

12\_常用的函数式接口\_Consumer接口练习\_字符串拼接输出

13\_常用的函数式接口\_Predicate接口

14\_常用的函数式接口\_Predicate接口\_默认方法and

15\_常用的函数式接口\_Predicate接口\_默认方法or&negate

视频时间：

36分40秒

本节目标 ：

独立编写Consumer接口完成字符串拼接案例

理解条件Predicate接口的作用

掌握Predicate中and方法的使用

掌握Predicate中or、negate方法的使用

#### 4.1 常用的函数式接口\_Consumer接口练习\_字符串拼接输出

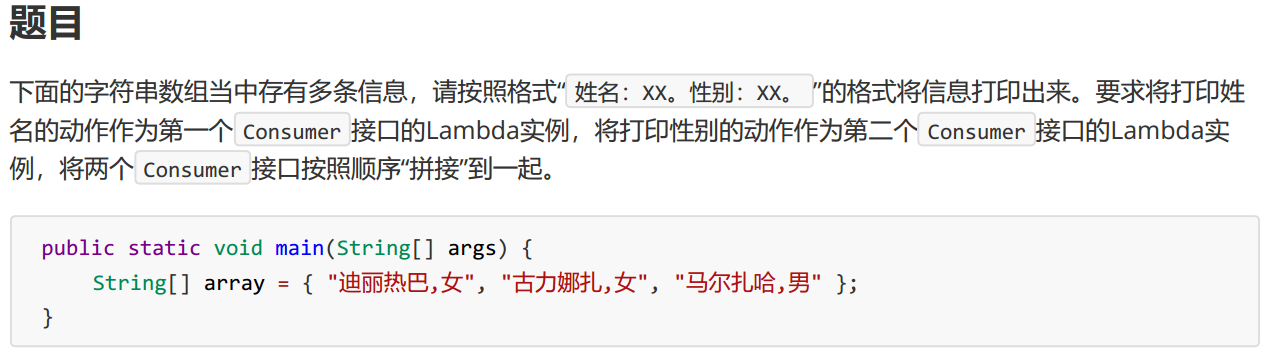
##### 4.1.1知识概述

* 常用的函数式接口\_Consumer接口练习\_字符串拼接输出

##### 4.1.2视频详情



##### 4.1.3总结与补充



##### 4.1.4课堂提问与练习

无

##### 4.1.5习题答案

无

#### 4.2 常用的函数式接口\_Predicate接口

##### 4.2.1知识概述

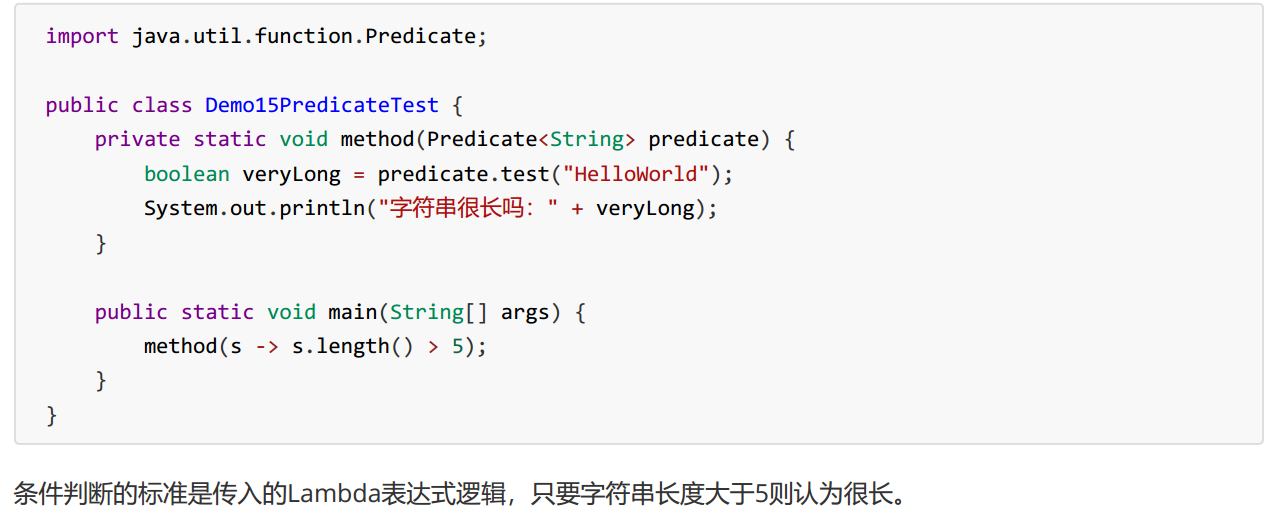
* 常用的函数式接口\_Predicate接口

##### 4.2.2视频详情



##### 4.2.3总结与补充

有时候我们需要对某种类型的数据进行判断，从而得到一个boolean值结果。这时可以使用java.util.function.Predicate<T> 接口



##### 4.2.4课堂提问与练习

无

##### 4.2.5习题答案

无

#### 4.3 常用的函数式接口\_Predicate接口\_默认方法and

##### 4.3.1知识概述

* 常用的函数式接口\_Predicate接口\_默认方法and

##### 4.3.2视频详情



##### 4.3.3总结与补充



##### 4.3.4课堂提问与练习

无

##### 4.3.5习题答案

无

#### 4.4 常用的函数式接口\_Predicate接口\_默认方法or&negate

##### 4.4.1知识概述

* Or : 或者
* negate : 非

##### 4.4.2视频详情



##### 4.4.3总结与补充

与 and 的“与”类似，默认方法 or 实现逻辑关系中的“或”

negate - > 非

##### 4.4.4课堂提问与练习

无

##### 4.4.5习题答案

无

### 第五堂课

本节知识点：

16\_常用的函数式接口\_Predicate接口练习\_集合信息筛选

17\_常用的函数式接口\_Function接口

18\_常用的函数式接口\_Function接口\_默认方法andThen

19\_常用的函数式接口\_Function接口练习\_自定义函数模型拼接

视频时间：

38分19秒

本节目标 ：

掌握Predicate集合信息筛选案例

理解转换型接口Function接口的作用

掌握Function接口andThen方法的使用

独立完成案例编写

#### 5.1 常用的函数式接口\_Predicate接口练习\_集合信息筛选

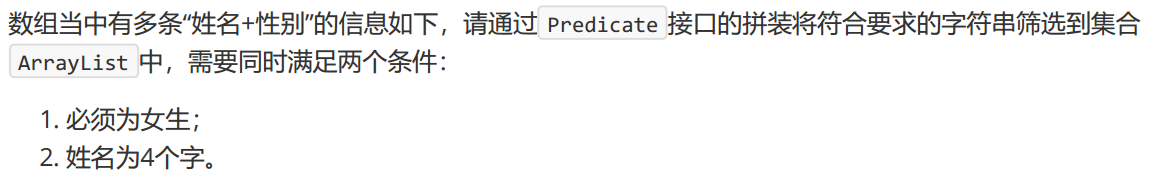
##### 5.1.1知识概述

* Predicate案例

##### 5.1.2视频详情



##### 5.1.3总结与补充



##### 5.1.4课堂提问与练习

无

##### 5.1.5习题答案

无

#### 5.2 常用的函数式接口\_Function接口

##### 5.2.1知识概述

* 转换型接口Function

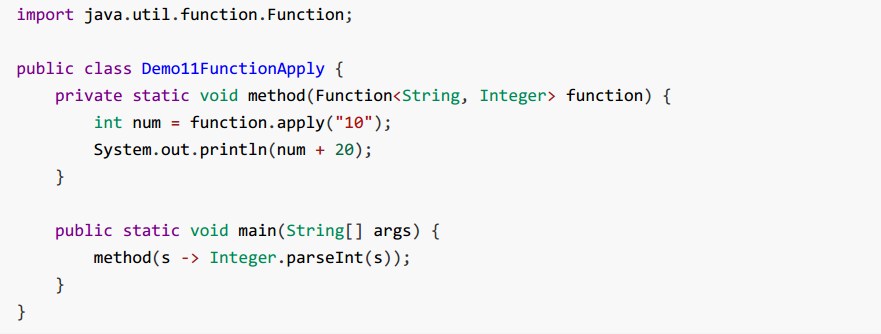
##### 5.2.2视频详情



##### 5.2.3总结与补充

java.util.function.Function<T,R> 接口用来根据一个类型的数据得到另一个类型的数据，前者称为前置条件，后者称为后置条件

Function 接口中最主要的抽象方法为： R apply(T t) ，根据类型T的参数获取类型R的结果。使用的场景例如：将 String 类型转换为 Integer 类型。



##### 5.2.4课堂提问与练习

无

##### 5.2.5习题答案

无

#### 5.3 常用的函数式接口\_Function接口\_默认方法andThen

##### 5.3.1知识概述

* Function 接口中有一个默认的 andThen 方法，用来进行组合操作。

##### 5.3.2视频详情



##### 5.3.3总结与补充



##### 5.3.4课堂提问与练习

无

##### 5.3.5习题答案

无

#### 5.4 常用的函数式接口\_Function接口练习\_自定义函数模型拼接

##### 5.3.1知识概述

* Function接口练习

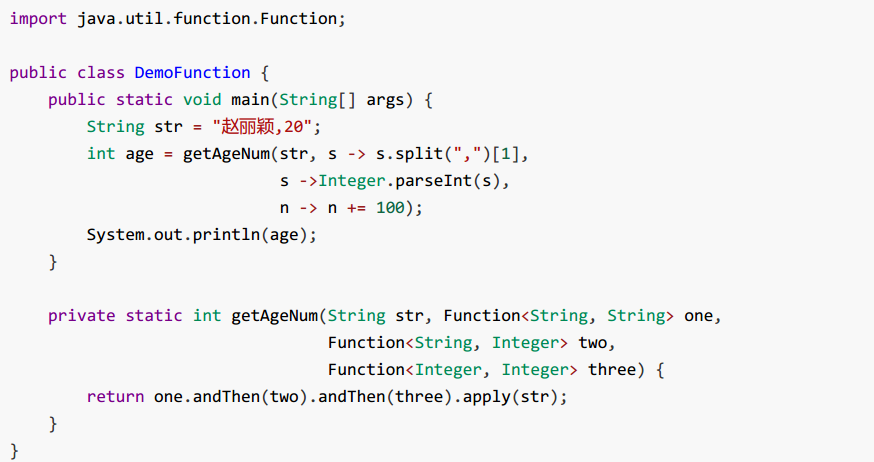
##### 5.3.2视频详情



##### 5.3.3总结与补充

请使用 Function 进行函数模型的拼接，按照顺序需要执行的多个函数操作为：  
 String str = "赵丽颖,20";

1. 将字符串截取数字年龄部分，得到字符串；  
 2. 将上一步的字符串转换成为int类型的数字；  
 3. 将上一步的int数字累加100，得到结果int数字



##### 5.3.4课堂提问与练习

无

##### 5.3.5习题答案

无