JAVA8之函数式编程Supplier接口和Consumer接口

一. Supplier接口

顾名思义,这是一个供应商,提供者.就如一个工厂一样.该类的源码如下:

package java.util.function;

@FunctionalInterface

public interface Supplier<T> {

T get();

}

该接口就一个抽象方法get方法,该接口在JAVA8之函数式接口返回实例篇中第一个示例就是利用的该接口.不用传入任何参数,直接返回一个泛型T的实例.就如同无参构造一样.示例就不再重复写了.

二. Consumer接口

顾名思义,这是一个消费者,该类的源码如下:

package java.util.function;

import java.util.Objects;

@FunctionalInterface

public interface Consumer<T> {

void accept(T t);

default Consumer<T> andThen(Consumer<? super T> after) {

Objects.requireNonNull(after);

return (T t) -> { accept(t); after.accept(t); };

}

}

1. accept方法

该函数式接口的唯一的抽象方法,接收一个参数,没有返回值.

2. andThen方法

在执行完调用者方法后再执行传入参数的方法.

3. 实例

package com.yczuoxin.demo.consumer;

import java.util.function.Consumer;

public class ConsumerTest {

public static void main(String[] args) {

Consumer<Integer> consumer = (x) -> {

int num = x \* 2;

System.out.println(num);

};

Consumer<Integer> consumer1 = (x) -> {

int num = x \* 3;

System.out.println(num);

};

consumer.andThen(consumer1).accept(10);

}

}

输出结果为:

可以看出先将10传入consumer方法执行,然后再将10传入consumer2方法执行.

三. 总结

这两个接口方法很少,但是实用性很强,并且可以看成是一对逆运算.一个作为生产工厂生产对象(Supplier),另一个作为消费者去消费对象(Consumer).