**基础题**

### 练习一：Pedicate接口使用

1. 请在测试类main方法中完成以下需求

已知有Integer[] arr = {-12345, 9999, 520, 0,-38,-7758520,941213}

1. 使用lambda表达式创建Predicate对象p1,p1能判断整数是否是自然数(大于等于0)
2. 使用lambda表达式创建Predicate对象p2,p2能判断整数的绝对值是否大于100
3. 使用lambda表达式创建Predicate对象p3,p3能判断整数是否是偶数

遍历arr，仅利用已创建的Predicate对象(不使用任何逻辑运算符)，完成以下需求

1. 打印自然数的个数
2. 打印负整数的个数
3. 打印绝对值大于100的偶数的个数
4. 打印是负整数或偶数的数的个数

答案

import java.util.function.Predicate;  
public class Test03 {  
 public static void main(String[] args) {  
 Integer[] arr = {-12345, 9999, 520, 0,-38,-7758520,941213};  
  
 //a) 使用lambda表达式创建Predicate对象p1,p1能判断整数是否是自然数  
 Predicate<Integer> p1 = (s) -> s>=0;  
 //b) 使用lambda表达式创建Predicate对象p2,p2能判断整数的绝对值是否大于100  
 Predicate<Integer> p2 = (s) -> Math.abs(s)>100;  
 //c) 使用lambda表达式创建Predicate对象p3,p3能判断整数是否是偶数  
 Predicate<Integer> p3 = (s) -> s%2==0;  
  
 //e) 遍历arr，仅利用已创建的Predicate对象(不使用任何逻辑运算符)，完成以下需求  
 int count1 = 0;  
 int count2 = 0;  
 int count3 = 0;  
 int count4 = 0;  
 for (Integer i : arr) {  
 //统计自然数个数  
 if (p1.test(i)){  
 count1++;  
 }  
 //统计负整数个数  
 if (p1.negate().test(i)){  
 count2++;  
 }  
 //统计绝对值大于100的偶数个数  
 if (p2.and(p3).test(i)){  
 count3++;  
 }  
 //统计是负整数或偶数的数的个数  
 if (p1.negate().or(p3).test(i)){  
 count4++;  
 }  
 }  
 //分别打印结果  
 System.out.println("自然数的个数为："+count1);  
 System.out.println("负整数的个数为："+count2);  
 System.out.println("绝对值大于100的偶数的个数为："+count3);  
 System.out.println("是负整数或偶数的数的个数为："+count4);  
 }  
}

### 练习二：Function接口使用

1. 使用lambda表达式分别将以下功能封装到Function对象中
2. 求Integer类型ArrayList中所有元素的平均数
3. 将Map<String,Integer>中value存到ArrayList<Integer>中
4. 已知学生成绩如下

|  |  |
| --- | --- |
| 姓名 | 成绩 |
| 岑小村 | 59 |
| 谷天洛 | 82 |
| 渣渣辉 | 98 |
| 蓝小月 | 65 |
| 皮几万 | 70 |

1. 以学生姓名为key成绩为value创建集合并存储数据，使用刚刚创建的Function对象求学生的平均成绩

答案

import java.util.\*;  
import java.util.function.Function;  
public class Test04 {  
 public static void main(String[] args) {  
 //1. 使用lambda表达式分别将以下功能封装到Function对象中  
 //a) 求Integer类型ArrayList中所有元素的平均数  
 Function<ArrayList<Integer>,Integer> f1 = (list)->{  
 Integer sum = 0;  
 for (Integer i : list) {  
 sum+=i;  
 }  
 return sum/list.size();  
 };  
  
 //b) 将Map<String,Integer>中value存到ArrayList<Integer>中  
 Function<Map<String,Integer>,ArrayList<Integer>> f2 = (map)->{  
 /\*ArrayList<Integer> list = new ArrayList<>();  
 for (String s : map.keySet()) {  
 Integer i = map.get(s);  
 list.add(i);  
 }\*/  
 Collection<Integer> values = map.values();  
 ArrayList<Integer> list = new ArrayList<>();  
 list.addAll(values);  
 return list;  
 };  
   
 //2 将学生姓名和成绩封装到map中  
 Map<String,Integer> map = new HashMap<String, Integer>();  
 map.put("岑小村", 59);  
 map.put("谷天洛", 82);  
 map.put("渣渣辉", 98);  
 map.put("蓝小月", 65);  
 map.put("皮几万", 70);  
  
 //利用Function求平均成绩  
 Integer avg = f2.andThen(f1).apply(map);  
 System.out.println("学生平均成绩为："+avg);  
 }  
}

### 练习三：如何获取流

问题：

简述单列集合、双列集合、数组分别如何获取Stream流对象，并进行演示

答：

1、java.util.Collection接口中加入了default方法stream()获取流对象，因此其所有实现类均可通过此方式获取流。

2、java.util.Map接口想要获取流，先通过keySet()、values()或entrySet()方法获取键、值或键值对的单列集合，再通过stream()获取流对象。

3、数组获取流，使用Stream接口中的的静态方法of(T...values)获取流。

public static void main(String[] args) {  
 List<String> list = new ArrayList<>();  
 Stream<String> stream1 = list.stream();  
  
 Set<String> set = new HashSet<>();  
 Stream<String> stream2 = set.stream();  
  
 Map<String, String> map = new HashMap<>();  
 Stream<String> keyStream = map.keySet().stream();  
 Stream<String> valueStream = map.values().stream();  
 Stream<Map.Entry<String,String>>entryStream = map.entrySet().stream();  
  
 String[] array = {"东邪", "西毒", "南帝", "北丐", "中神通"};  
 Stream<String> stream = Stream.of(array);  
 }

### 练习四：过滤：filter、结果收集(数组)

问题：

有如下7个元素黄药师，冯蘅，郭靖，黄蓉，郭芙，郭襄，郭破虏，使用Stream将以郭字开头的元素存入新数组

答：

import java.util.stream.Stream;  
public class Test {  
 public static void main(String[] args) {  
 Stream<String> stream = Stream.of("黄药师", "冯蘅", "郭靖", "黄蓉", "郭芙", "郭襄", "郭破虏");  
 String[] guos = stream.filter(s -> s.startsWith("郭")).toArray(String[]::new);  
 }  
}

### 练习五：取用前几个：limit、跳过前几个：skip

问题：

已知ArrayList集合中有如下元素{陈玄风、梅超风、陆乘风、曲灵风、武眠风、冯默风、罗玉风}，使用Stream

1、取出前2个元素并在控制台打印输出。

2、取出后2个元素并在控制台打印输出。

答：

import java.util.ArrayList;  
public class Test04 {  
 public static void main(String[] args) {  
 ArrayList<String> list = new ArrayList<>();  
 list.add("陈玄风");  
 list.add("梅超风");  
 list.add("陆乘风");  
 list.add("曲灵风");  
 list.add("武眠风");  
 list.add("冯默风");  
 list.add("罗玉风");

list.stream().limit(2).forEach(System.out::println);  
 list.stream().skip(list.size() - 2).forEach(System.out::println);  
 }  
}

### 练习六：映射：map、逐一消费：forEach

问题：

有如下整数1，-2，-3，4，-5

使用Stream取元素绝对值并打印

答：

import java.util.stream.Stream;  
public class Test {  
 public static void main(String[] args) {  
 Stream<Integer> stream = Stream.of(1, -2, -3, 4,-5);  
 stream.map(Math::abs).forEach(System.out::println);  
 }  
}

### 练习七：组合：concat、结果收集(list)

问题：

已知数组arr1中有如下元素{郭靖，杨康}，arr2中有如下元素{黄蓉，穆念慈}，使用Stream将二者合并到List集合

答：

import java.util.stream.Stream;  
public class Test {  
 public static void main(String[] args) {  
 Stream<String> streamA = Stream.of("郭靖", "杨康");  
 Stream<String> streamB = Stream.of("黄蓉", "穆念慈");  
 List<String> strList = Stream.concat(streamA, streamB).collect(Collectors.toList());  
 }  
}

### 练习八：获取并发流

问题：

请分别写出获取并发流的两种方式。

答：

public class Test {  
 public static void main(String[] args) {  
 Collection<String> coll = new ArrayList<>();  
 Stream<String> parallelStream1 = coll.parallelStream();

Stream<Integer> parallelStream2 = Stream.of(100, 200, 300, 400).parallel();  
 }  
}