

Java - 09

流与文件I/O

在学习这个板块之前，建议先看一下：[Java 8 Stream](#)

task 1 流API

这个任务中我们学习流API，不过在这之前，我们会先向你介绍一些很基本的方法。要使用流API,我们需要一个**Stream**对象。如果我们有一个类似List的集合，它没有实现**Stream**。不过**Collection**接口有一个**Stream**方法，会返回对应的那个Collection的一个Stream对象。现在我们尝试以下的操作：

```
List<String> strings = List.of("I", "am", "a", "list", "of", "Strings");
Stream<String> stream = strings.stream();
//调用流API的方法，例如我们希望最多有4个元素
Stream<String> limit = stream.limit(4);
//最后我们打印结果
System.out.println("limit = " + limit);
```

我们会得到什么结果，结合资料思考一下为什么会得到这样的结果，而不是我们所期望的结果？

任务：再学习一下流API,修改上面的代码使之生成我们想要的结果。你可以尝试创建自己的流，并总结一下使用流的规则。

tips: 你可以看到在处理一些流的过程中我们将操作堆叠在一起，放在一个流管道里，就像这样：

```
List<Integer> squaresList = numbers.stream()
    .map(i -> i * i) //你可能会对这种表达式有兴趣
    .sorted((x, y) -> y - x) //你可能会对这种表达式有
    兴趣
    .collect(Collectors.toList());
```

学习一下这种操作堆叠的原理，并结合上面使用流API的规则总结规律。

或许你会对上面注释的部分感兴趣，这叫做**Lambda**表达式，强烈建议你深入掌握这种技巧，这会对你的代码质量有很大的提升：[详解Java中的Lambda表达式](#)，[Java Lambda 表达式](#)

要求：将实现的代码、运行截图和自己的学习总结推送到GitHub仓库上，此处提交链接：

进阶挑战——应用流API

现在我们又成为了音乐厅的员工，这次我们需要再次改进我们的自动点歌机，以满足老板永无止境的
要求。我们对点歌机进行了升级，现在的点歌机的**Song**类如下：

```
public class Song{
    private String title;
    private String artist;
    private int genre;
    private int year;
    private int timesPlayed;
    // 利用注解或者自己创建构造器和get方法
}
```

同时我们也给出了模拟代码：

```
class Songs {
    public List<Song> getSongs() {
        return List.of(
            new Song("$10", "Hitchhiker", "Electronic", 2016, 183),
            new Song("Havana", "Camila Cabello", "R&B", 2017, 324),
            new Song("Cassidy", "Grateful Dead", "Rock", 1972, 123),
            new Song("50 ways", "Paul Simon", "Soft Rock", 1975, 199),
            new Song("Hurt", "Nine Inch Nails", "Industrial Rock", 1995, 257),
            new Song("Silence", "Deleerium", "Electronic", 1999, 134),
            new Song("Hurt", "Johnny Cash", "Soft Rock", 2002, 392),
            new Song("Watercolour", "Pendulum", "Electronic", 2010, 155),
            new Song("The Outsider", "A Perfect Circle", "Alternative Rock", 2004, 312),
            new Song("With a Little Help from My Friends", "The Beatles", "Rock", 1967,
168),
            new Song("Come Together", "The Beatles", "Blues rock", 1968, 173),
            new Song("Come Together", "Ike & Tina Turner", "Rock", 1970, 165),
            new Song("With a Little Help from My Friends", "Joe Cocker", "Rock", 1968,
46),
            new Song("Immigrant Song", "Karen O", "Industrial Rock", 2011, 12),
            new Song("Breathe", "The Prodigy", "Electronic", 1996, 337),
            new Song("What's Going On", "Gaye", "R&B", 1971, 420),
            new Song("Hallucinate", "Dua Lipa", "Pop", 2020, 75),
            new Song("Walk Me Home", "P!nk", "Pop", 2019, 459),
            new Song("I am not a woman, I'm a god", "Halsey", "Alternative Rock", 2021,
384),
            new Song("Pasos de cero", "Pablo Alborán", "Latin", 2014, 117),
            new Song("Smooth", "Santana", "Latin", 1999, 244),
            new Song("Immigrant song", "Led Zeppelin", "Rock", 1970, 484));
    }
}
```

你的任务：

1. 注意到更新后的歌曲包含歌曲的流派（**genre**），你需要找出所有包含“**rock**”流派的歌曲。

tip: 你需要过滤（**filter**）数据，只保留某一个流派的**Song**，你可以把目标歌曲收集（**collect**）到一个新的**List**中。

2. 列出所有的歌曲流派。

tip: **map**方法接受一个**Function**，可以接受一个类型的对象并返回不同的对象，这意味着**Lambda**表达式可能会派上用场。

要求：将实现的代码和运行截图推送到GitHub仓库上，此处提交链接：

task 2 串行化

对象有行为和状态。行为在类中，但状态在各个对象中，那么保存一个对象的状态时会发生什么呢？

事实上，如何保存Java程序的状态，一般有以下两种选择：

1. 如果你的数据只由生成它的Java程序使用：你可以选择串行化并保存到某个文件当中，这样当你再需要用这个状态的时候，又可以读取串行化对象，把它变成一个可以使用的对象。
2. 如果你的数据由其他程序使用：你必须遵循其他程序的约定，写一个纯文本文件以方便读取。

任务：结合之前学习的流的知识，了解一下`InputStream/OutputStream`类，这里有教程可供参考：[Java 流\(Stream\)、文件\(File\)和IO](#)

你需要做的是：尝试将上文的歌曲作为串行化对象写入文件，再通过逆串行化将对象的状态读取出来。

tip: 了解一下`Serializable`接口，掌握它的功能。

要求：将实现的代码和运行截图推送到GitHub仓库上，此处提交链接：

进阶挑战——文件I/O

写文本数据（实际上就是String）和写对象类似，只不过你要写一个String而不是一个对象，而且要使用`FileWriter`而不是`FileOutputStream`：

```
//就像这样
class WriteAFile {
    public static void main(String[] args) {
        // 所有I/O代码都有可能抛出一个IOException
        try {
            FileWriter writer = new FileWriter("Foo.txt");// 如果文件“Foo.txt”不存在，则会
            // 自动创建这个文件
            writer.write("hello foo!");
            writer.close();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

任务：

请自行学习关于文件写入和读入的操作，实现自动点歌机中将Song类写入和读出文件中的功能，参考上文链接。

要求：将实现的代码和运行截图推送到GitHub仓库上，此处提交链接：

出题人QQ: 1727448271@qq.com