Java09-Task1pro

首先在 Song 类中补充了构造方法，和 getGenre（）和 toString（）方法：

```
public class Song {  25 个用法  新 *
    Song(String title, String artist, String genre, int year, int timesPla
        this.title = title;
        this.artist = artist;
        this.genre = genre;
        this.year = year;
        this.timesPlayed = timesPlayed;

    }
    public String getGenre() {  1 个用法  新 *
        return genre;
    }
    public String toString() {  新 *
        return (title + " " +"from:"+ artist + " " +"genre is:" +genre + '
    }
    }
```

在 Songs 类中通过 getSongs（）方法返回一个 Song 类型的列表（得到 Song 对象）。因此我的想法是首先 new 一个 Songs 对象出来，然后引用 getSongs 方法得到一个元素类型为 Song 的列表，用 List<Song> songlist 来接。从而得到那个有很多 new Song 对象的列表。

下一步，先用 songlist.stream() 得到一个流，再用 filter（genre ->"Rock".equals(genre.getGenre())过滤得到 genre 为“Rock”的 Song 对象。最后用 collect（Collectors.toList()）归到一个 list 中，我定义一个 Song 类型的 list collect 来接，最后可以打印 collect 看具体的情况。

注意！当我没有写 toString（）方法时，最后会得到地址信息而不是具体的字符串。

代码如下所示：

```java
public class demo4 {  新 *
    static class Songs {  2 个用法  新 *
        public List<Song> getSongs() {  1 个用法  新 *
                new Song( title: "Immigrant song",  artist: "Led Zeppelin",  ge
    }
}
public  static void main(String[] args) {  新 *
    Songs songs = new Songs();
    List<Song> songList = songs.getSongs();
    List<Song> collect = songList.stream().filter(genre -> "Rock".equals(
    System.out.println(collect);


}
```

最后执行结果：

```
 .encoding=UTF-8 -classpath /Users/duqiu/Desktop/java/javapro/out/production/java1 com.java.Stream.demo4
[Cassidy from:Grateful Dead genre is:Rock 1972 123, With a Little Help from My Friends from:The Beatles genre
 is:Rock 1967 168, Come Together from:Ike & Tina Turner genre is:Rock 1970 165, With a Little Help from My
 Friends from:Joe Cocker genre is:Rock 1968 46, Immigrant song from:Led Zeppelin genre is:Rock 1970 484]

进程已结束，退出代码为 0

java1 > src > com > java > Stream > Song > toString
```

然后是打印出所有的歌曲流派：

利用 map 方法：在 new Fuction<T,T>时，可以进行流中数据类型转换（但是不能有基本数据类型）同理，可以接受一种对象，然后返回不同的对象。

以下为 map 方法代码：

```java
35  public  static void main(String[] args) {  新 *
36      Songs songs = new Songs();
37      List<Song> songList = songs.getSongs();
38      List<Song> collect = songList.stream().filter(genre -> "Rock".equals(genre.getGenre())).collect(Collecto
39      System.out.println(collect);
40      songList.stream().map(Song::getGenre).forEach(System.out::println);
41
```

以上为利用 Lambda 表达式简化的结果，也可以写为 genre->genre.getGenre(),完成一次映射，再打印输出结果。实现由参数到方法的处理，例如打印方法如果参数类型相同，System.out.println() -> System.out:： println);Lambda 表达式可实现该操作。
Function<T,R>是对类型为 T 的对象实现操作，结果返回 R 类型的对象，包含方法 R apply

(T, t);

代码执行结果为:（只截取了部分结果，（结果太长了））。