



链滴

Stream 流式计算 (Juc-08)

作者: [yscx](#)

原文链接: <https://ld246.com/article/1606120456767>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



概述

大数据：存储+计算

集合、mysql，实质上都是去存储用的

例题：

题目要求：一分钟内完成此，只能使用一行代码实现

现在有5个用户，筛选：

1. id必须是偶数
2. 年龄必须大于23岁
3. 用户名转成大写字母
4. 用户名字母倒着排序
5. 只输出一个用户

```
package net.yscx.stream;
```

```
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;
```

```
import java.util.Arrays;  
import java.util.List;
```

```
/**  
 * @Author WangFuKun
```

```

* @create 2020/11/23 15:44
*/
/*
*题目描述：一分钟内完成此题，只能用一行代码实现！现在有5个用户进行以下条件进行筛选
* id必须是偶数
* 年龄必须大于23岁
* 用户名转成大写字母
* 用户名字母倒着排序
* 只输出一个用户
* */
public class Test {
    public static void main(String[] args) {
        User u1 = new User(1, "a", 21);
        User u2 = new User(2, "b", 24);
        User u3 = new User(3, "c", 38);
        User u4 = new User(4, "d", 32);
        User u5 = new User(5, "e", 31);
        //集合负责储存
        List<User> list = Arrays.asList(u1, u2, u3, u4, u5);
        //计算交给Stream流
        list.stream()
            .filter(u->{return u.getId()%2==0;})
            .filter(u->{return u.getAge()>23;})
            .sorted((uu1,uu2)->{return uu2.getName().compareTo(uu1.getName());})
            .limit(2)
            .forEach(System.out::println);
    }
}
@Data
@AllArgsConstructor
@NoArgsConstructor
class User {
    private int id;
    private String name;
    private int age;
}

```

具体的其他操作可以看文档，也可以看上一篇[函数式接口](#)