



姓名：李全欣 班级：计算机2018级4-7 成绩：100分

一.简答题 (共7题,100.0分)

1 以下Student类可直接使用。

```
public class Student {  
    private int number;  
    private String name;  
    private String clazz;  
    private int score;  
  
    public Student(int number, String name, String clazz, int score) {  
        this.number = number;  
        this.name = name;  
        this.clazz = clazz;  
        this.score = score;  
    }  
    // 省略getter/setter方法  
}
```

在以下类中，按需求编写方法，完成对STUDENTS集合的操作

```
public class StreamTest {  
    private static final List<Student> STUDENTS = create();  
    private static final String CLAZZ1 = "软件1班";  
    private static final String CLAZZ2 = "软件2班";  
    private static List<Student> create() {  
        Student s1 = new Student(2018008, "张扬", CLAZZ2, 66);  
        Student s2 = new Student(2018005, "刘飞", CLAZZ1, 92);  
        Student s3 = new Student(2018007, "李明", CLAZZ2, 42);  
        Student s4 = new Student(2018006, "赵勇", CLAZZ2, 56);  
        Student s5 = new Student(2018002, "王磊", CLAZZ1, 81);  
        Student s6 = new Student(2018010, "牛娜", CLAZZ1, 78);  
        List<Student> students = new ArrayList<>();  
        students.add(s1);students.add(s2);students.add(s3);  
        students.add(s4);students.add(s5);students.add(s6);  
        return students;  
    }  
  
    public static void main(String[] args) {  
        // 调用实现方法测试  
    }  
}
```

```
}
```

```
// 实现方法
```

```
}
```

说明：

需求描述中的指定X，均指方法的参数

所有方法均有返回值，尝试直接编程return语句，基于stream操作流直接返回所需结果

如果返回集合，使用List集合类型

尝试使用简写

注意过滤代码格式

方法1，获取成绩小于等于指定分数，的全部学生

正确答案：

```
/**
 * 获取成绩小于等于指定分数，的全部学生
 * @param score
 * @return
 */
private static List<Student> listStudents(int score) {
    return STUDENTS.stream()
        .filter(s -> s.getScore() <= score)
        .collect(Collectors.toList());
}
```

我的答案：

```
public static void main(String[] args) {
    // 调用实现方法测试
    //forEach();
    filter(66);
}
private static void filter(int score ){
    STUDENTS.stream()
        .filter(s -> s.getScore() <= score)
        .collect(Collectors.toList())
        .forEach(s -> System.out.println(s.getNumber()+" "+s.getName()+" "+s.getClazz()+" "+s.getScore()));
}
}
```

```
-----
2018008 张扬 软件2班 66
2018007 李明 软件2班 42
2018006 赵勇 软件2班 56
```

2

方法2，获取指定班级，成绩小于等于指定分数，的全部学生

正确答案：

```
/**
 * 获取指定班级，成绩小于等于指定分数，的全部学生
 * @param score
 * @param clazz
 * @return
 */
private static List<Student> listStudents(int score, String clazz) {
    return STUDENTS.stream()
        .filter(s -> s.getScore() <= score && s.getClazz().equals(clazz))
        .collect(Collectors.toList());
}
```

```

        .filter(s -> clazz.equals(s.getClass()))
        .filter(s -> s.getScore() <= score)
        .collect(Collectors.toList());
    }

```

我的答案：

```

public static void main(String[] args) {
    // 调用实现方法测试
    filter(CLAZZ1,99);
}

// 实现方法
private static void filter(String clazz ,int score ){
    STUDENTS.stream()
        .filter(s -> s.getClass().equals(clazz) )
        .filter(s -> s.getScore()<=score)
        .collect(Collectors.toList())
        .forEach(s -> System.out.println(s.getNumber()+" "+s.getName()+" "+s.getClass()+" "+s.getScore()));
    }
}

```

2018005 刘飞 软件1班 92
 2018002 王磊 软件1班 81
 2018010 牛娜 软件1班 78

3

方法3，获取指定班级，成绩小于等于指定分数，的全部学生的姓名。注意返回类型

正确答案：

```

/**
 * 获取指定班级，成绩小于等于指定分数，的全部学生的姓名。注意返回类型
 * @param calzz
 * @return
 */
private static List<String> listStudents(String calzz, int score) {
    return STUDENTS.stream()
        .filter(s -> calzz.equals(s.getClass()))
        .filter(s -> s.getScore() <= score)
        .map(Student::getName)
        .collect(Collectors.toList());
}

```

我的答案：

```

public static void main(String[] args) {
    filter(CLAZZ1,99);
}

// 实现方法
private static void filter(String clazz ,int score ){
    STUDENTS.stream()
        .filter(s -> s.getClass().equals(clazz) )
        .filter(s -> s.getScore()<=score)
        .collect(Collectors.toList())
        .forEach(s -> System.out.println(s.getName()));
    }
}

```

Problems @ Javadoc Declaration Console

<terminated> StreamTest [Java Application] C:\Program Files

刘飞
 王磊
 牛娜

4

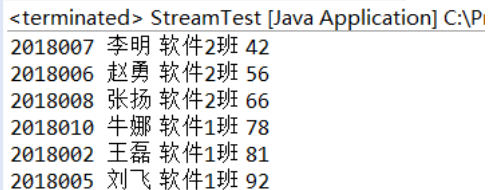
方法4，按成绩由低到高排序，返回全部学生

正确答案：

```
/**
 * 按成绩由低到高排序，返回全部学生
 * @return
 */
private static List<Student> listStudentsSorted() {
    return STUDENTS.stream()
        .sorted(Comparator.comparing(Student::getScore))
        .collect(Collectors.toList());
}
```

我的答案：

```
public static void main(String[] args) {
    sort();
}
private static void sort(){
    STUDENTS.stream()
        .sorted(Comparator.comparing(Student::getScore))
        .collect(Collectors.toList())
        .forEach(s -> System.out.println(s.getNumber()+" "+s.getName()+" "+s.getClazz()+" "+s.getScore()));
}
```



```
<terminated> StreamTest [Java Application] C:\P
2018007 李明 软件2班 42
2018006 赵勇 软件2班 56
2018008 张杨 软件2班 66
2018010 牛娜 软件1班 78
2018002 王磊 软件1班 81
2018005 刘飞 软件1班 92
```

5

方法5，获取指定班级，成绩小于等于指定分数，成绩由高到低排序，的全部学生

正确答案：

```
/**
 * 获取指定班级，成绩小于等于指定分数，成绩由高到低排序，的全部学生
 * @param score
 * @param clazz
 * @return
 */
private static List<Student> listStudentsSorted(int score, String clazz) {
    return STUDENTS.stream()
        .filter(s -> clazz.equals(s.getClazz()))
        .filter(s -> s.getScore() <= score)
        .sorted(Comparator.comparing(Student::getScore).reversed())
        .collect(Collectors.toList());
}
```

我的答案：

```
public static void main(String[] args) {
    // 调用实现方法测试
    filter(CLASS1,99);
}

// 实现方法
private static void filter(String clazz ,int score ){
    STUDENTS.stream()
        .filter(s -> s.getClazz().equals(clazz) )
        .filter(s -> s.getScore()<=score)
        .sorted(Comparator.comparing(Student::getScore))
        .collect(Collectors.toList())
}
```

```

        .forEach(s -> System.out.println(s.getNumber()+" "+s.getName()+" "+s.getClazz()+" "+s.getScore()));
    }
}
<terminated> Stream<Integer>
2018010 牛娜 软件1班 78
2018002 王磊 软件1班 81
2018005 刘飞 软件1班 92

```

6 方法6，获取指定班级，成绩小于等于指定分数，成绩由高到低排序，的全部学生的学号。注意返回类型

正确答案：

```

/**
 * 获取指定班级，成绩小于等于指定分数，成绩由高到低排序，的全部学生的学号。注意返回类型
 * @param score
 * @param clazz
 * @return
 */
private static List<Integer> listStudentsSorted2(int score, String clazz) {
    return STUDENTS.stream()
        .filter(s -> clazz.equals(s.getClazz()))
        .filter(s -> s.getScore() <= score)
        .sorted(Comparator.comparing(Student::getScore).reversed())
        .map(Student::getNumber)
        .collect(Collectors.toList());
}

```

我的答案：

```

public static void main(String[] args) {
    // 调用实现方法测试
    filter(CLAZZ1,99);
}

// 实现方法
private static void filter(String clazz ,int score ){
    STUDENTS.stream()
        .filter(s -> s.getClazz().equals(clazz) )
        .filter(s -> s.getScore()<=score)
        .sorted(Comparator.comparing(Student::getScore))
        .collect(Collectors.toList())
        .forEach(s -> System.out.println(s.getNumber()));
}
}
<terminated> Stre
2018010
2018002
2018005

```

7 方法7，获取指定班级，成绩小于等于指定分数，的全部学生

以学生学号为键，学生分数为值，Map分组，返回

正确答案：

```

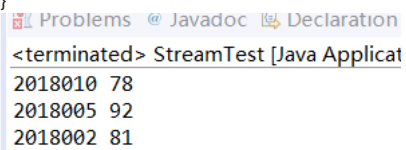
/**
 * 获取指定班级，成绩小于等于指定分数，的全部学生
 * 以学生学号为键，学生分数为值，Map分组，返回
 * @param score
 * @param clazz
 * @return
 */
private static Map<Integer, Integer> listStudentsSorted3(int score, String clazz) {
    return STUDENTS.stream()
        .filter(s -> clazz.equals(s.getClazz()))
        .filter(s -> s.getScore() <= score)

```

```
        .collect(Collectors.toMap(Student::getNumber, Student::getScore));  
    }  
}
```

我的答案：

```
public static void main(String[] args) {  
    // 调用实现方法测试  
    map(CLAZZ1,99);  
}  
private static void map(String clazz,int score) {  
    STUDENTS.stream()  
        .filter(s -> s.getClass().equals(clazz) )  
        .filter(s -> s.getScore()<=score)  
        .collect(Collectors.toMap(Student::getNumber, Student::getScore))  
        .forEach((k,v) -> {  
            System.out.println(k+" "+v);  
        });  
}
```



<terminated> StreamTest [Java Applicat
2018010 78
2018005 92
2018002 81
