

# 面向服务软件工程Assignment 1

## 1. Schema

- **Student.xsd**: 定义了基础的 `StudentType` 和 `CourseScoreType`，供其他模块复用。
- **Archive.xsd**: 引用 `Student.xsd`，扩展了学生个人档案信息（如出生日期、地址）。
- **ScoreList.xsd**: 定义了以学生为聚合的成绩单结构（用于验证 XML3）。

## 2. 随机生成学生成绩

### 2.1 实现逻辑

使用 Java DOM API 创建 XML 文档树。程序首先读取基础数据，然后循环生成9名虚拟学生的信息。对于每位学生，随机生成5门课程的平时成绩和期末成绩，计算总评，并包含强制逻辑以确保至少有一名学生的不及格成绩。

### 2.2 运行结果 (XML3)

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <StudentList xmlns="http://jw.nju.edu.cn/schema">
3      <Student>
4          <id>STU1000</id>
5          <name>Student_0</name>
6          <gender>Male</gender>
7          <Course courseId="C01">
8              <scoreDaily>67</scoreDaily>
9              <scoreFinal>91</scoreFinal>
10             <scoreTotal>83</scoreTotal>
11          </Course>
12          <Course courseId="C02">
13              <scoreDaily>89</scoreDaily>
14              <scoreFinal>43</scoreFinal>
15              <scoreTotal>56</scoreTotal>
16          </Course>
17          <Course courseId="C03">
18              <scoreDaily>99</scoreDaily>
19              <scoreFinal>95</scoreFinal>
20              <scoreTotal>96</scoreTotal>
21          </Course>
22          <Course courseId="C04">
23              <scoreDaily>63</scoreDaily>
24              <scoreFinal>91</scoreFinal>
25              <scoreTotal>82</scoreTotal>
26          </Course>
27          <Course courseId="C05">
28              <scoreDaily>97</scoreDaily>
29              <scoreFinal>76</scoreFinal>
30              <scoreTotal>82</scoreTotal>
31          </Course>
32      </Student>
```

## 3.按课程重组与排序

### 3.1 实现逻辑

- **分组 (Grouping):** 利用 Muenchian Method (或 `for-each-group`) 按 `courseId` 对成绩进行分组。
- **排序 (Sorting):** 在每门课程内部, 使用 `<xsl:sort>` 按 `scoreTotal` 进行降序排列。

### 3.2 运行结果 (XML4)

```
<?xml version="1.0" encoding="UTF-8"?><CourseGradeList>
  <Course id="C01">
    <StudentScore studentId="STU1006" score="90">
      <scoreDaily xmlns="http://jw.nju.edu.cn/schema">83</scoreDaily>
      <scoreFinal xmlns="http://jw.nju.edu.cn/schema">93</scoreFinal>
      <scoreTotal xmlns="http://jw.nju.edu.cn/schema">90</scoreTotal>
    </StudentScore>
    <StudentScore studentId="STU1003" score="89">
      <scoreDaily xmlns="http://jw.nju.edu.cn/schema">81</scoreDaily>
      <scoreFinal xmlns="http://jw.nju.edu.cn/schema">93</scoreFinal>
      <scoreTotal xmlns="http://jw.nju.edu.cn/schema">89</scoreTotal>
    </StudentScore>
    <StudentScore studentId="STU1004" score="88">
      <scoreDaily xmlns="http://jw.nju.edu.cn/schema">87</scoreDaily>
      <scoreFinal xmlns="http://jw.nju.edu.cn/schema">89</scoreFinal>
      <scoreTotal xmlns="http://jw.nju.edu.cn/schema">88</scoreTotal>
    </StudentScore>
    <StudentScore studentId="STU1009" score="84">
      <scoreDaily xmlns="http://jw.nju.edu.cn/schema">96</scoreDaily>
      <scoreFinal xmlns="http://jw.nju.edu.cn/schema">79</scoreFinal>
      <scoreTotal xmlns="http://jw.nju.edu.cn/schema">84</scoreTotal>
    </StudentScore>
    <StudentScore studentId="STU1000" score="83">
      <scoreDaily xmlns="http://jw.nju.edu.cn/schema">67</scoreDaily>
      <scoreFinal xmlns="http://jw.nju.edu.cn/schema">91</scoreFinal>
      <scoreTotal xmlns="http://jw.nju.edu.cn/schema">83</scoreTotal>
    </StudentScore>
    <StudentScore studentId="STU1008" score="83">
      <scoreDaily xmlns="http://jw.nju.edu.cn/schema">73</scoreDaily>
      <scoreFinal xmlns="http://jw.nju.edu.cn/schema">88</scoreFinal>
      <scoreTotal xmlns="http://jw.nju.edu.cn/schema">83</scoreTotal>
    </StudentScore>
    <StudentScore studentId="STU1001" score="80">
      <scoreDaily xmlns="http://jw.nju.edu.cn/schema">96</scoreDaily>
      <scoreFinal xmlns="http://jw.nju.edu.cn/schema">74</scoreFinal>
    </StudentScore>
  </Course>
</CourseGradeList>
```

## 4. 筛选不及格名单

### 4.1 实现逻辑

使用 SAX 解析器流式读取 XML4。通过 `startElement` 监听 `StudentScore` 和分数节点, 通过 `characters` 读取具体分数值。当检测到 `scoreTotal` 小于 60 分时, 将该条记录写入缓冲区, 并在 `endElement` 时决定是否输出到最终的 XML5 文件中。

### 4.2 运行结果(XML5)

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <FailedList>
3      <StudentScore studentId="STU1007" courseId="C01">
4
5          <scoreDaily>64</scoreDaily>
6
7          <scoreFinal>41</scoreFinal>
8
9          <scoreTotal>47</scoreTotal>
10
11      </StudentScore>
12      <StudentScore studentId="STU1002" courseId="C01">
13
14          <scoreDaily>93</scoreDaily>
15
16          <scoreFinal>9</scoreFinal>
17
18          <scoreTotal>34</scoreTotal>
19
20      </StudentScore>
21      <StudentScore studentId="STU1000" courseId="C02">
22
23          <scoreDaily>89</scoreDaily>
24
25          <scoreFinal>43</scoreFinal>
26
27          <scoreTotal>56</scoreTotal>
28
29      </StudentScore>
30      <StudentScore studentId="STU1007" courseId="C03">
31
32          <scoreDaily>76</scoreDaily>
33
34          <scoreFinal>41</scoreFinal>
35
36          <scoreTotal>51</scoreTotal>
37
38      </StudentScore>
39      <StudentScore studentId="STU1006" courseId="C03">
40
41          <scoreDaily>83</scoreDaily>
```