

Task1

1)

-整型

- byte存储中等范围整数，储存年龄等
- short存储中等范围整数
- int默认整型，日常整数计算首选
- long存储大范围整数（需加 L 后缀）如身份证号

-字符型

char存储单个字符，本质是 Unicode 编码值（十进制整数）

-浮点型

- float 单精度浮点数（需加 f 后缀），精度较低
- double双精度浮点数，默认浮点类型，精度更高

-布尔型

boolean储逻辑值，仅 true（真）和 false（假）两种取值

2)

整型	占用字节	表示范围	
byte	1字节	-128~127	
short	2字节	-32768~32767	
int	4字节	-2147483648~2147483647	
long	8 字节	-9223372036854775808 ~ 9223372036854775807	

3)是隐式类

因为char类型自动转换为 int 类型参与运算，没有我们手动去强制转换类型（显式转换），所以是隐式类型转换  
char 类型在参与运算时，会自动转换为对应的 ASCII 码值（int 类型）。'0' 对应的 ASCII 码值是 48。

然后，int a = 4, a + c 就相当于 4 + 48, 结果为 intb=a+c=52

4) 结果：false true false

```
Integer x = new Integer(18); Integer y = new Integer(18); System.out.println(x == y);
```

使用 new Integer() 分别创建对象 x 和 y，而这两个对象是不同的 Integer，对象 == 比较的是对象的引用（即内存地址），所以结果为 false。

```
Integer z = Integer.valueOf(18); Integer k = Integer.valueOf(18); System.out.println(z == k);
```

Integer.valueOf() 方法会用到 Integer 类型的缓存池，当传入的数值在此缓存池的范围（-128~127）内时，会直接从缓存池中获取已存在的 Integer 对象。18范围内，所以 z 和 k 引用的是缓存池中的同一个对象，== 比较结果为 true

```
Integer m = Integer.valueOf(300); Integer p = Integer.valueOf(300); System.out.println(m == p);
```

300 超出了 Integer 缓存池 -128 到 127 的范围, 此时 Integer.valueOf() 会像 new Integer() 一样, 每次都新建一个 Integer 对象。m 和 p 是不同的对象, == 比较引用, 结果为 false

## Task2

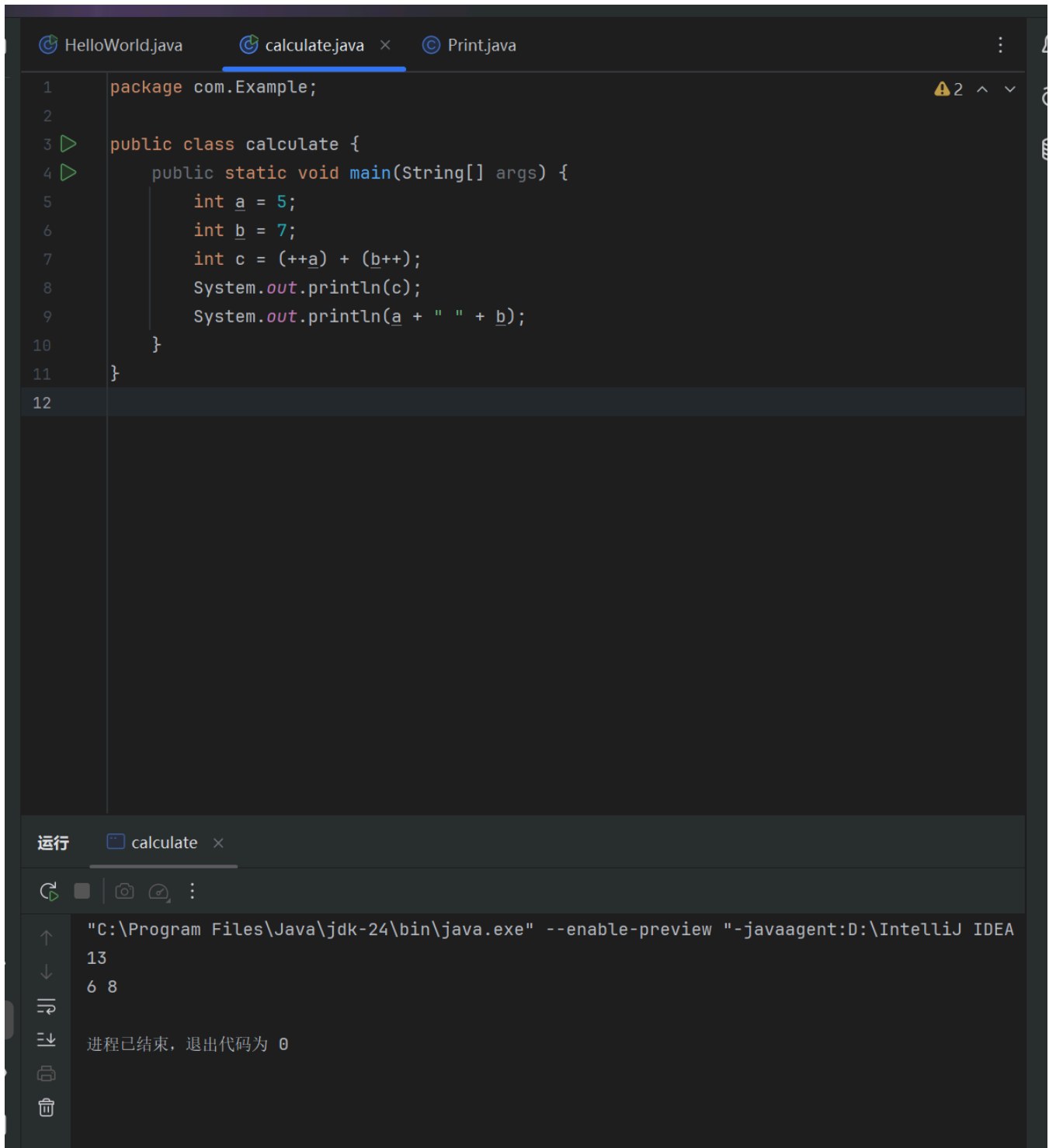
5) int a = 5 ;//a取5

int b = 7 ;//b取7

int c = (++a) + (b++)//a自增1为 6,b自增1的数为 8 ,c取a运算后的值和b运算前的和 (6+7)= 13

System.out.println( c );输出c的取值 13

System.out.println(a+" "+b);左到右执行, a 输出 6," "输出 ,b输出 8



The screenshot shows an IDE with three tabs: `HelloWorld.java`, `calculate.java` (active), and `Print.java`. The `calculate.java` file contains the following code:

```
1 package com.Example;
2
3 public class calculate {
4     public static void main(String[] args) {
5         int a = 5;
6         int b = 7;
7         int c = (++a) + (b++);
8         System.out.println(c);
9         System.out.println(a + " " + b);
10    }
11 }
12
```

Below the code editor, the **运行** (Run) tab is active, showing the command used to execute the program:

```
"C:\Program Files\Java\jdk-24\bin\java.exe" --enable-preview "-javaagent:D:\IntelliJ IDEA
```

The output of the program is displayed below the command:

```
13
6 8
```

At the bottom of the run window, it states: **进程已结束，退出代码为 0** (Process ended, exit code 0).

#### 6) int类型通常以二进制补码的形式存储

原码：最高位为符号位（0 表示正数，1 表示负数），其余位表示数值大小

补码：正数的补码和原码相同，负数的补码是其原码除符号位外，其余各位取反，然后末位加 1

float类型遵循IEEE 754 标准进行存储

符号位（1 位）：用于表示数的正负，0 表示正数，1 表示负数。

指数位（8 位）：以偏移值 127 为基准，存储实际指数加上偏移值后的结果。

尾数位（23 位）：存储小数部分，默认最高位是 1

两个正数相加结果为负数是因为发生了溢出。在int的二进制补码运算中，两个超出了 int 类型所能表示的范围的数相加，会导致最高位（符号位）发生进位，原本表示正数的符号位变为 1，结果就成了负数。而对于float，当运算结果超出了float 所能表示的最大范围时，会得到 Infinity（正无穷），如果是负数超出范围则得到 -Infinity（负无穷）。

#### Task3

1)

```

import java.util.HashMap;
import java.util.Map;

public class Main {
    public static void main(String[] args) {
        String[] rawData = {"math:5", "English:10", "Chinese:10", "math:20",
                             "English:10", "chemistry:30", "math:10", "math:20"};
        // 键为科目名称，值为错题数量，修正泛型类型
        Map<String, Integer> map = new HashMap<>();
        for (String entry : rawData) {
            String[] parts = entry.split(":");
            String subject = parts[0]; // 获取科目名称
            int score = Integer.parseInt(parts[1]); // 获取该记录的错题数

            // 判断map中是否包含该科目
            if (map.containsKey(subject)) {
                // 累加错题数
                map.put(subject, map.get(subject) + score);
            } else {
                // 首次出现的科目，直接存入
                map.put(subject, score);
            }
        }

        System.out.println("错题统计结果: ");

        for (Map.Entry<String, Integer> entry : map.entrySet()) {
            System.out.println(entry.getKey() + ": " + entry.getValue() + "题");
        }
    }
}

```

```

"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:D:\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=56983" -Dfile.encoding=UTF-8
错题统计结果:
chemistry: 30题
English: 20题
Chinese: 10题
math: 55题

```

2)

```

import java.util.HashMap;
import java.util.Map;

public class Main {
    public static void main(String[] args) {
        String[] rawData = {"math:5", "English:10", "Chinese:10", "math:20",
                             "English:10", "chemistry:30", "math:10", "math:20"};

        // 键为科目名称，值为错题数量
    }
}

```

```

Map<String, Integer> map = new HashMap<>();
for (String entry : rawData) {
    String[] parts = entry.split(":");
    String subject = parts[0];
    int score = Integer.parseInt(parts[1]);

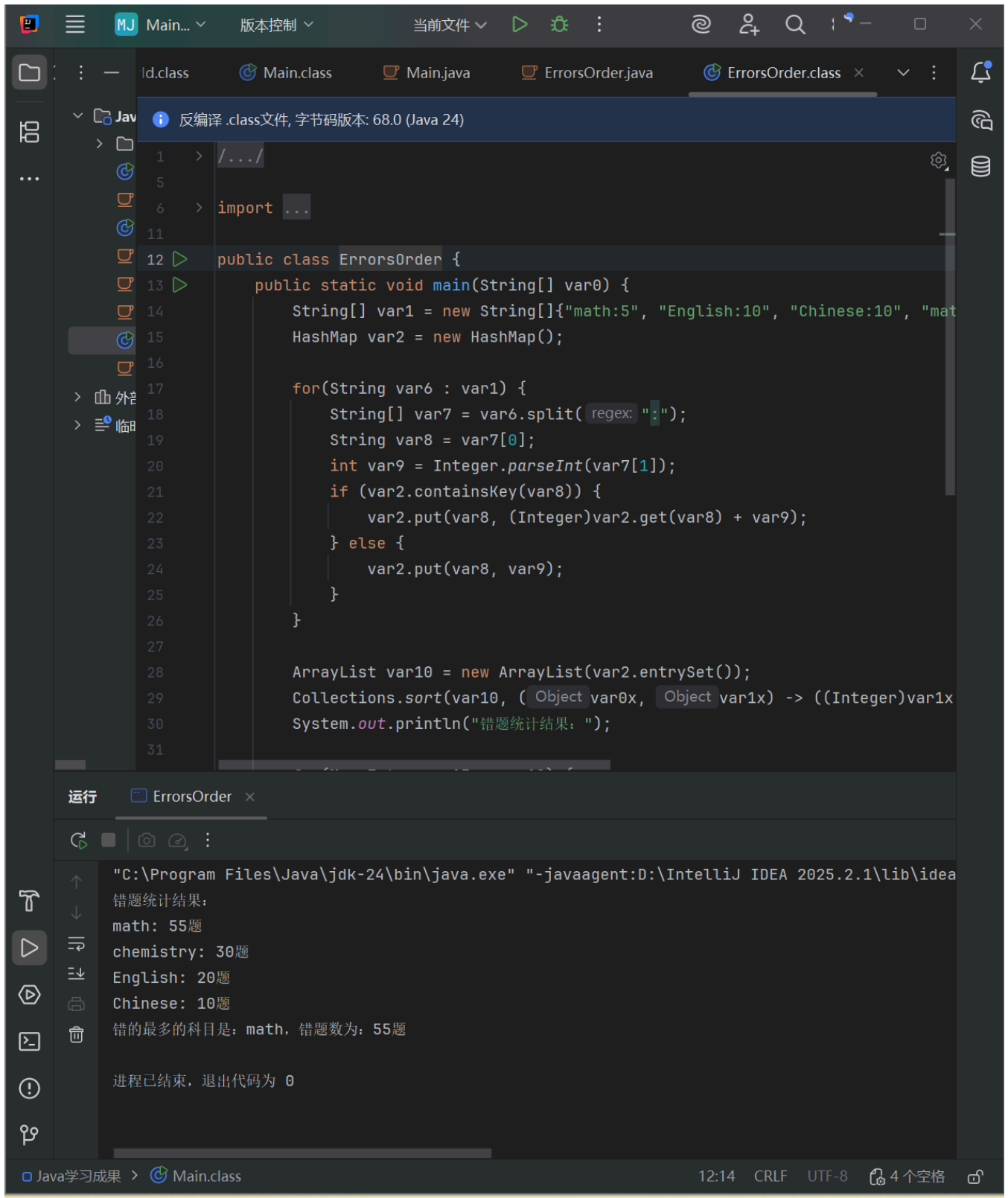
    if (map.containsKey(subject)) {
        map.put(subject, map.get(subject) + score);
    } else {
        map.put(subject, score);
    }
}

System.out.println("错题统计结果: ");
for (Map.Entry<String, Integer> entry : map.entrySet()) {
    System.out.println(entry.getKey() + ": " + entry.getValue() + "题");
}

// 找出错题最多的科目
String maxSubject = null;
int maxErrors = 0;
for (Map.Entry<String, Integer> entry : map.entrySet()) {
    if (entry.getValue() > maxErrors) {
        maxErrors = entry.getValue();
        maxSubject = entry.getKey();
    }
}

System.out.println("\n错题最多的科目是: " + maxSubject + ",错题数为: " + maxErrors +
"题");
}
}

```



3)

```
import java.util.*;  
  
public class ErrorQuestionStatistics {  
    public static void main(String[] args) {  
        // 原始数据  
        String rawData =  
            "math:5,English:10,Chinese:10,math:20,English:10,chemistry:30,math:10,math:20";  
        // 用于存储科目和对应错题数的Map
```

```

Map<String, Integer> subjectErrorMap = new HashMap<>();

// 解析原始数据并统计
String[] dataItems = rawData.split(",");
for (String item : dataItems) {
    String[] parts = item.split(":");
    if (parts.length == 2) {
        String subject = parts[0];
        int errorCount = Integer.parseInt(parts[1]);
        subjectErrorMap.put(subject, subjectErrorMap.getOrDefault(subject, 0) +
errorCount);
    }
}

// 用于后续输入的Scanner
Scanner scanner = new Scanner(System.in);
System.out.println("输入科目：错题数（完成后输入exit）");
while (true) {
    String input = scanner.nextLine();
    if ("exit".equals(input)) {
        break;
    }
    String[] inputParts = input.split(" ");
    if (inputParts.length == 2) {
        String subject = inputParts[0];
        try {
            int errorCount = Integer.parseInt(inputParts[1]);
            subjectErrorMap.put(subject, subjectErrorMap.getOrDefault(subject, 0) +
errorCount);
        } catch (NumberFormatException e) {
            System.out.println("错题数格式错误，请输入整数！");
        }
    } else {
        System.out.println("输入格式错误，应为“科目 错题数”！");
    }
}

// 转换为List以便排序
List<Map.Entry<String, Integer>> entryList = new ArrayList<>
(subjectErrorMap.entrySet());
// 按错题数降序排序
entryList.sort((entry1, entry2) -> entry2.getValue().compareTo(entry1.getValue()));

System.out.println("错题统计结果：");
int maxErrorCount = -1;
String maxErrorSubject = "";
for (Map.Entry<String, Integer> entry : entryList) {
    String subject = entry.getKey();
    int errorCount = entry.getValue();
    System.out.println(subject + ": " + errorCount + "题");
    if (errorCount > maxErrorCount) {
        maxErrorCount = errorCount;
        maxErrorSubject = subject;
    }
}

```

```

    }
}

System.out.println("错的最多的科目是: " + maxErrorSubject + ", 错题数为: " +
maxErrorCount + "题");

scanner.close();
}
}

```

The screenshot shows an IDE with the following code in `ErrorsOrder.java`:

```

import java.util.*;

public class ErrorsOrder {
    public static void main(String[] var0) {
        String[] var1 = new String[]{"math:5", "English:10", "Chinese:10", "math:20", "English:10", "chemistry:30", "math:10", "math:20"};
        HashMap var2 = new HashMap();
        Scanner var3 = new Scanner(System.in);

        for(String var8 : var1) {
            String[] var9 = var8.split(":");
            String var10 = var9[0];
            int var11 = Integer.parseInt(var9[1]);
            if (var2.containsKey(var10)) {
                var2.put(var10, (Integer)var2.get(var10) + var11);
            } else {
                var2.put(var10, var11);
            }
        }

        System.out.println("输入科目, 错题数 (完成后输入exit)");
    }
}

```

The output window shows the following execution results:

```

C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:D:\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=S4559" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
输入科目, 错题数 (完成后输入exit)
math:23
exit
错题统计结果:
math: 78题
chemistry: 30题
English: 20题
Chinese: 10题
错的最多的科目是: math, 错题数为: 78题

```

4)

```

import java.util.HashMap;
import java.util.Map;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class ErrorsOrder {
    public static void main(String[] args) {
        // 原始数据
        String[] rawDat = {"math:5", "English:10", "Chinese:10", "math:20",
            "English:10", "chemistry:30", "math:10", "math:20"};
        String[] xiaohongErrors = {"math:0", "Chinese:0", "English:0", "physics:0",
            "math:0"};

        // 创建Map存储错题统计
        Map<String, Integer> xiaomingMap = new HashMap<>();
        Map<String, Integer> xiaohongMap = new HashMap<>();

        // 加载原始数据
        processData(rawDat, xiaomingMap);
        processData(xiaohongErrors, xiaohongMap);
    }
}

```



```

Scanner sc = new Scanner(System.in);

// 输入小明的新错题数据
System.out.println("请输入小明的新错题数据（格式：科目:错题数，输入exit结束）：");
inputNewErrors(sc, xiaomingMap);

// 输入小红的新错题数据
System.out.println("\n请输入小红的新错题数据（格式：科目:错题数，输入exit结束）：");
inputNewErrors(sc, xiaohongMap);

// 显示统计结果
System.out.println("\n===== 小明错题统计结果 =====");
displayStatistics(xiaomingMap);

System.out.println("\n===== 小红错题统计结果 =====");
displayStatistics(xiaohongMap);

sc.close();
}

/**
 * 处理原始数据
 */
private static void processData(String[] data, Map<String, Integer> map) {
    for (String entry : data) {
        String[] parts = entry.split(":");
        String subject = parts[0];
        int error = Integer.parseInt(parts[1]);
        updateMap(subject, error, map);
    }
}

/**
 * 输入新的错题数据
 */
private static void inputNewErrors(Scanner sc, Map<String, Integer> map) {
    String input;
    while (true) {
        input = sc.nextLine();
        if (input.equalsIgnoreCase("exit")) {
            break;
        }
        String[] parts = input.split(":");
        if (parts.length == 2) {
            String subject = parts[0];
            try {
                int error = Integer.parseInt(parts[1]);
                updateMap(subject, error, map);
            } catch (NumberFormatException e) {
                System.out.println("输入无效，请输入数字作为错题数！");
            }
        } else {
            System.out.println("输入格式错误，请使用：科目:错题数");
        }
    }
}

```

```

    }
}

/**
 * 更新Map中的错题数
 */
private static void updateMap(String subject, int error, Map<String, Integer> map) {
    map.put(subject, map.getOrDefault(subject, 0) + error);
}

/**
 * 显示统计结果并找错题最多的科目
 */
private static void displayStatistics(Map<String, Integer> map) {
    if (map.isEmpty()) {
        System.out.println("没有错题数据!");
        return;
    }

    // 将Map转为List并按错题数降序排序
    ArrayList<Map.Entry<String, Integer>> list = new ArrayList<>(map.entrySet());
    Collections.sort(list, (o1, o2) -> o2.getValue().compareTo(o1.getValue()));

    for (Map.Entry<String, Integer> entry : list) {
        System.out.println(entry.getKey() + ": " + entry.getValue() + "题");
    }

    Map.Entry<String, Integer> maxEntry = list.get(0);
    System.out.println("错的最多的科目是: " + maxEntry.getKey() + ", 错题数为: " +
maxEntry.getValue() + "题");
}
}

```

```
"C:\Program Files\Java\jdk-24\bin\java.exe" --enable-preview "-javaagent:D:\IntelliJ IDE
```

请输入小明的新错题数据（格式：科目:错题数，输入exit结束）：

*math 5*

输入格式错误，请使用：科目:错题数

*math:5*

*exit*

请输入小红的新错题数据（格式：科目:错题数，输入exit结束）：

*english:4*

*exit*

===== 小明错题统计结果 =====

math: 60题

chemistry: 30题

English: 20题

Chinese: 10题

错的最多的科目是: math, 错题数为: 60题

===== 小红错题统计结果 =====

english: 4题

English: 0题

physics: 0题

Chinese: 0题

math: 0题

错的最多的科目是: english, 错题数为: 4题

进程已结束，退出代码为 0