

## JC2002 Java 编程 - 实践 2（第 2 天）

本实践分为两部分：Java 编程部分（任务 1-2）和 Git 部分（任务 3-5）。第一部分的目标是在实践中熟悉 Java 的基本语法和结构。

1. 编写一个简单的 Java 程序，将两个整数作为用户输入（**开始和结束**）。然后，编写一个从**开始到结束**的循环，计算从**开始到结束**的所有整数之和。例如，如果用户输入的**起点**值为 5，**终点**值为 10，结果将是  $5+6+7+8+9+10=45$ 。运行循环后，程序应将结果打印在控制台中。试着用不同的输入使用该程序，以确保其正常运行。
2. 在代码中添加一条条件语句，检查 **end** 是否大于或等于 **start**。如果 **start** 大于 **end**，程序应显示错误信息，而不是计算结果。尝试用不同的输入来使用程序，以确保其正常运行。

第二部分的目标是学习如何使用 Git，既可以自己使用，也可以与伙伴一起使用，从而了解如何使用 Git 共同完成一个项目。Git 部分分为三个任务。自己完成任务 3，然后找一个伙伴完成任务 4 和 5。

3. 自行在 <https://learngitbranching.js.org> 上进行实践。这些在线互动练习会指导你学习 Git 的基础知识，让你能在命令行上自如地使用 Git。最多不要超过 20-30 分钟，你可以稍后再回来自己完成练习。
4. 接下来，找一个合作伙伴来完成以下任务。在 GitHub 上创建一个仓库，其中包含一个以你的名字命名的 "**Readme.md**" 文件，并在另一个文件中创建一个 Java 应用程序。您可以使用在任务 1-2 中创建的 java 文件。提交到 GitHub 后，将你的合作伙伴作为协作者添加到仓库中，这样他们就能克隆并推送到本地仓库。

现在，您和您的伙伴应各自添加一个分支并修改 Java 代码（例如添加一些新功能），然后将这些修改合并到主分支中。在你们各自完成工作的同时，在 "**Readme.md**" 文件中添加注释，概述你们添加的功能。

5. 提交对文件的修改并将其推送到版本库。然后，请你的伙伴在他们的本地版本中修改该文件的同一行代码（不要拉取第一次修改），并尝试提交。您应该会看到存在冲突的警告。你需要在文本编辑器中打开有冲突的文件，并通过删除

冲突标记手动解决冲突。请点击[这里查看](https://imagej.net/develop/git/conflicts)：  
<https://imagej.net/develop/git/conflicts>。